# An Approach based on the ifcOWL Ontology to Support Indoor Navigation

**ABSTRACT** This paper presents an indoor navigation support system based on the Building Information Models (BIM) paradigm. Although BIM is initially defined for the Architecture, Engineering and Construction/Facility Management (AEC/FM) industry, the authors believe that it can provide added value in this context. To this end, the authors will focus on the Industry Foundation Classes (IFC) standard for the formal representation of BIM. The approach followed in this paper will be based on the ifcOWL ontology, which translates the IFC schemas into Ontology Web Language (OWL). This way of expressing the elements of a building can be used to code information that is very useful for navigation, such as the location of elements related to the actions desired by the user. It is important to note that this design is intended to be used as a complement to other well-known tools and techniques for indoor navigation. The proposed modifications have been successfully tested in a variety of simulated and real scenarios. The main limitation of the proposal is the immense amount of information contained in the ifcOWL ontology, which causes difficulties involving its processing and the time necessary to perform operations on it.

## 1. INTRODUCTION

The widespread adoption of the Building Information Models (BIM) paradigm in Architecture, Engineering and Construction/Facility Management (AEC/FM) industry has brought undoubted benefits over previous methods: cost reduction during the design and construction, better interoperability among stakeholders, better management of the life cycle of buildings (planning, construction, maintenance, dismantling), among others [1]. Furthermore, it has made the actors involved more likely to share the "semantic" information of buildings, which implies a collective improvement of knowledge. In this context, it is thus necessary to define a formal representation that acts as a standard, the Industry Foundation Classes (IFC), an open standard for over 15 years.

The motivation of BIM with regard to data reuse and compatibility between heterogeneous applications could suggest interacting with the concept of Semantic Web (SW). Faithfully translating the IFC schemas into OWL (*Ontology Web Language*) is equivalent to generating an ontology from IFC schemas, and would yield substantial advantages, among which the authors highlight the following [3]:

- Reuse of existing ontologies both in the field of AEC and other areas.
- Facilitate the use of SW tools, especially those related to searches, such as SPARQL.
- Provide the logical formalism that the EXPRESS language (in which IFC schemes are usually expressed) lacks.
- Communication with external data sources.
- New use cases and applications.
- Encourage open publication of models
- Use reasoning techniques developed for the SW [4].
- Discover and use resources present in the SW.
- Soft integration with the Internet of Things (IoT) or with Geographic Information Systems (GIS).
- Support for stakeholders to make faster decisions with more detailed global information.
- More intuitive manipulation of building information [5].
- More widespread use due to the wider community of users, and more possibilities to develop technology in this sense.
- It allows predefined relations, restricted constructions and expressiveness, avoids fragmentation and favors the appearance of de facto standards [6].

These possible advantages have encouraged several researchers to implement a set of more or less functional IFC-OWL conversions, despite the fact that IFC has a rather complicated and compressed structure. One of these implementations is ifcOWL [3, 7-12]. This ontology maintains the strong structure of the original EXPRESS version [13], although some expressiveness is lost with the conversion [14] due to certain problems, such as the conversion of list data types. In addition, ifcOWL is a huge ontology which, due to its size, poses serious difficulties for use in real applications requiring consultation or reasoning [7]. However, the advantages obtained with the conversion compensate for these disadvantages, and it is reasonable to affirm that ifcOWL is a reliable version of the IFC schema. This ontology also has two interesting

features in favor of its adoption: its growing use and the support of an organization of experts, such as the buildingSMART Linked Data Working Group.

Although BIM is not generally considered suitable for navigation [15], it provides some promising features in this regard, such as semantic information about building elements and their use, the properties of each building element, the spatial relationships between those elements and their geometric representation [16]. Furthermore, the very nature of ifcOWL (focused on the description of building elements) allows it to be integrated with complementary approaches for indoor navigation. The aim of this paper is to leverage these possible advantages by proposing some uses of ifcOWL for an indoor navigation system, and in particular to determine the global position of the elements in the navigation map and to fix the destination position based on the orders of the users. It is important to note that this design is not intended to be used as a standalone system, but as a complement to other well-known tools and techniques for indoor navigation.

The rest of this document is structured as follows. In subsection 1.1, the concept of ontology and the Semantic Web is briefly discussed. The authors consider it important to introduce the reader to these concepts, which are the basis of the ensuing work. In Subsection 1.2, BIM and the structure of IFC are presented in more detail. Subsection 1.3 summarizes work related to ontology applications in the field of the AEC, especially those based on ifcOWL. This recapitulation will prepare the reader for the description of the system implementation contained in Section 2. The results of the application of the approach are shown in Section Finally, the conclusions are presented.

## 1.1. BRIEF INTRODUCTION TO THE SEMANTIC WEB: THE CONCEPT OF ONTOLOGY

The World Wide Web contains a large and growing amount of information that is mainly encoded using HTML. This language is based on tags intended mainly to instruct a web browser how the information should be displayed, and not on the rational organization of the information. This implies that the power of its use is conditioned by the ability of users to navigate between heterogeneous sources of information and avoid saturation and data overload. The Semantic Web concept [22, 23] implies the goal of encoding information in a way that is understandable to both humans and computer systems. This goal has involved the development of new and increasingly complex forms of knowledge expression for true semantic interoperability.

As can be deduced from the above, the paradigm for the operation of the semantic Web implies the need for formal definitions of the domain models. These formal models (called ontologies) should include the definition of the terms used and their relationships explicitly and more effectively than thesauri or metadata. In the field of Artificial Intelligence, an ontology is "an artefact constituted by a specific vocabulary used to describe a certain reality, along with a set of explicit assumptions related to the desired meaning of the vocabulary" [24]. Although Studer's [25] is the best-known definition of ontology ("a formal, explicit specification of a shared conceptualization"), other authors have proposed other definitions that adjust the meaning of an ontology in a less ambiguous manner [26-29]. These definitions integrate the ideas of explicit representation of knowledge [30] and the method of communication of the actors involved in a domain (human-human, human-computer or computer-computer [31]). Ontologies are also defined as "*a well-founded mechanism for the representation and exchange of structured information*" [32], so they have been successfully applied in several fields.

Ontologies consist mainly of vocabulary, but this is not enough to express the intended meaning when planning a new design. A functional ontology is composed of concepts (abstract terms generally organized into taxonomies), relationships (links between non-hierarchical concepts), instances (concrete existences of abstract concepts) and axioms (valid rules in the domain model).

The growing interest in ontologies has given rise to new forms of expression of knowledge which add the capacity to deduce new knowledge that is not explicitly expressed from an analysis of existing data. In this way, systems are provided with a greater capacity for reasoning and a common way of expressing knowledge between the different systems that collaborate toward achieving the same goal. The World Wide Web Consortium (W3C) defines the Resource Description Framework (RDF) as the "standard model for data interchange on the Web" [33]. RDF represents an increase in semantic interoperability, but it is insufficient for the needs of the SW. In practice, other languages with more possibilities are used, such as OWL, for building ontologies [34-36].

## 1.2. BIM AND IFC SCHEMA STRUCTURE

There are several formal definitions of what a BIM is. Some relevant definitions are given below.

- *"A BIM is a digital representation of physical and functional characteristics of a facility. As such it serves as a shared knowledge resource for information about a facility forming a reliable basis for decisions during its lifecycle from inception onward"*. [37]
- *"Discrete set of electronic object-oriented information used for design, construction and operation of a built asset"*. [38]

- *"Shared digital representation of physical and functional characteristics of any built object (including buildings, bridges, roads, etc.) which forms a reliable basis for decisions".* [39]

As is to be expected from these definitions, BIM is used in several areas of interest such as: implementation and adoption, energy efficiency and simulation or training in academia and industry, among others [40]. However, the use of BIMs produces a large amount of data, since any type of information in the AEC field should be described in the models. In addition, this immense amount of data must be adequately handled by very heterogeneous sectors [41].

IFC is a neutral platform and open-data model for representing most of the components of a physical building [42]. It was proposed by buildingSMART, which defines it as a *"conceptual data schema and an exchange file format for BIM data"* [43]. IFC specifications include entities that define building elements (walls, beams, doors...), geometry features (extruded solid area, swept area solid...) and basic constructs (cartesian points...). IFC is mainly encoded in EXPRESS schemas (although an XML schema specification is also available), which contain hundreds of entities organized in an object-based hierarchy. However, the promising impact that the advantages offered by IFC should have is mitigated by a set of severe constraints [44]:

- IFC is suitable for the production and exchange of information but it does not have an optimal structure for searching and analyzing results.
- The use of IFC schemas has contributed to the adoption of the BIM approach in numerous applications; however, the rigidity of its schema places serious limitations on the integration of heterogeneous sources of information from different domains [45].
- It is highly redundant to cover the needs and point of view of the various AEC sectors, which complicates its use [46].
- Lack of formal rigidness.
- Limited reuse and interoperability.
- Lack of built-in distribution [10].

IFC schema are structured into four layers [47]: resource layer, core layer, interoperability layer and domain/application layer. IFC allows new classes to be defined as subclasses of an existing class, from which they inherit their properties. Figure 1 shows an extract of the IFC hierarchy. All IFC classes (except resource classes) are derived from the IfcRoot superclass. This class allows establishing basic attributes of the model such as identity, description and ownership. IfcRoot yields three fundamental classes for the categories defined: objects (IfcObjectDefinition), relationships (IfcRelationship) and properties (IfcPropertyDefinition). Each of these three classes has its respective inheritance tree of subclasses. An in-depth analysis of the IFC structure can be found in [48] and in the corresponding technical documentation [49].
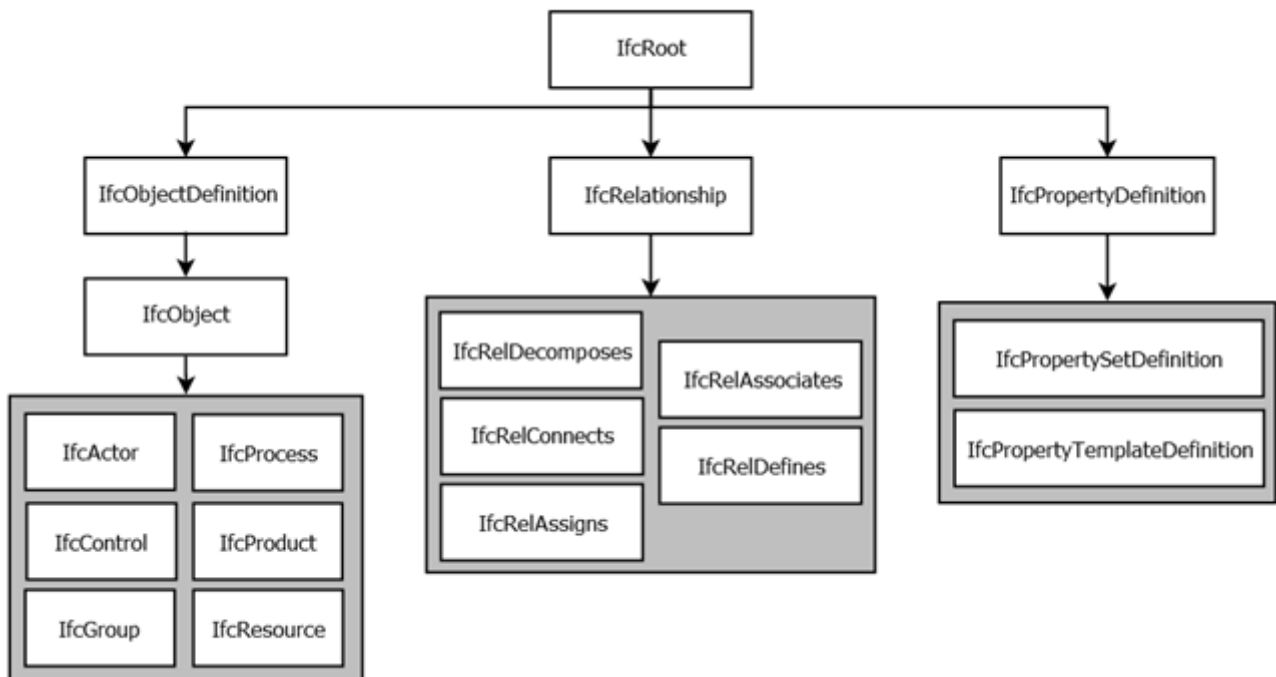
FIGURE 1. IFC basic hierarchy.

## 1.3. RELATED WORK ON BIM AND ONTOLOGIES: IFCOWL

There are quite a few references to the use of ontologies in fields related to the construction industry, but it is also true that, in general, the use of SW technologies in AEC is not as broad as initially expected [50]. As indicated above, this paper will focus on the use of ontologies based on BIM standards, and in particular on the use of ifcOWL.

Liu et al. [51] establish a model for estimating the cost of building construction based on ontologies. These ontologies mainly refer to the model concept, work item and construction conditions. For the reasoning they use, among other tools, Jess and SWRLJessBridge. In another work, Barbau et al. [52] describe the OntoSTEP project, which involves the development of a plugin for the Protégé tool. This plugin is capable of translating STEP's EXPRESS schemas into OWL. In [42], Farias et al. present another semi-automatic method to generate an OWL ontology for COBie, another BIM standard. Cursi, Simeone and Coraglia [53] present a prototype for integrating a building information model with a series of ontologies that act as a knowledge base. Lee, Eastman and Solihin [54] propose a new way of formalizing the knowledge domain in the context of building information modeling. This proposal, based on ontological principles, aims to reduce the redundant requirements and rules for the exchange of information when model views are defined. Zhang, Beetz and de Vries [44] propose a general framework based on expanded SPARQL functions with a dual purpose: to simplify the search syntax and to retrieve useful information related to 3D geometry based on the requirement to check the use cases. Krijnen and Beetz [55] develop a SPARQL implementation that works directly on a binary serialization format for IFC instance models. This format is based on ISO 10303-26 and relies on the HDF5 standard, resulting in files that are 2-3 times smaller than RDF databases. Terkaj and Pauwels [7] analyze attempts in the literature to obtain a usable ontology from IFC EXPRESS schemas. From this analysis, these authors conclude that the ontology should be used initially to allow the IFC instances conversion into RDF graphs, and they propose a new conversion procedure.

In the literature, there are several works in this regard that document some problems with the IFC-OWL conversion. These problems focus especially on those structures allowed in EXPRESS that are shared differently in OWL. One example is the ordered lists in EXPRESS, where each item represents a concept according to its position in the list. In OWL, it would be translated by a succession of rdf:list-rdf:first-rdf:rest items, which is considered in depth in [3]. This makes it difficult to enjoy the full semantic power of the resulting ontologies. Despite these problems, several conversions are available, since the transition from IFC to OWL is not unique and each author has adapted the conversion to the specific needs of the user [55]. Some of these proposals are analyzed and compared in [9], where the need for a standard for the effective use of the ontology is also confirmed.

Focusing on the ifcOWL, Schevers and Drogemuller [57] develop a primitive (and incomplete) version of a unidirectional conversion of an IFC schema to an OWL ontology for research purposes. Later, Pauwels and Terkaj, based on several previous studies such as [6], propose a conversion procedure from EXPRESS schema to OWL ontology [9]. That work can be regarded as the basis of the ifcOWL ontology. In [5], a semi-automatic method is proposed for generating an ontology called IfcWoD to yield a better ontology for linking IFC data in the field of the Semantic Web. This ontology uses terms included in the ifcOWL. The authors show that the two ontologies together (ifcOWL and ifcWoD) simplify the syntax and response time when searching. Terkaj and Sojic [11] propose improving the ifcOWL ontology by including new classes that explain the relationships between existing classes, especially those relationships between IFC object occurrences, object type and pre-defined property sets. In [13], the authors analyze three approaches to reference checking for AEC to obtain a benchmark performance.

In [16], a BIM-based model (called BO-IDM) is defined that provides highly detailed semantic information. This model represents a building using 18 classes. In [58], the authors detail a Semantic Web-based approach to defining building views from the ifcOWL ontology. This ontology, expressed in the OWL/RDF serialization, is combined with SWRL logical rules. The use of these logical rules makes it possible to outline the definition of new concepts (ExternalDoor, BuildingEnvelope, etc.) and thus increase the expressiveness of the models, while at the same time extracting subgraphs from the ontology for processing as building views. These building views contain the necessary and sufficient information needed to properly execute a building process. Pauwels et al. [8] propose an improvement of the ifcOWL ontology through an alternative representation for geometric data. This representation relies on the inclusion of a WKT (Well-Known Text) serialization. Another improvement proposed by Pauwels and Roxin [12] is SimpleBIM, a simplification of building models defined in ifcOWL graphs that removes the geometry part and reuses new properties defined by ifcWoD.

Other authors have proposed extensions of the ifcOWL ontology by including other well-known ontologies. This is the case of Chbeir et al. [59], who have developed a knowledge-based model for storing building information. This new model, called OntoH2G, extends, among other ontologies, the Semantic Sensor Network Ontology (SSN) and the W3C Geospatial Ontology (W3CGeo). Bus et al [60] use ifcOWL to automate the model checking for compliance against normative regulations (particularly the French one). To this end, they define a Regulations ontology that uses the terms of a simplified version of

ifcOWL and expands it with certain complex concepts related to regulation. Finally, this paper will cite the work of Gómez-Romero et al. [45], which presents a fuzzy extension of the ifcOWL ontology.

As a summary, Table 1 shows a reduced set of extensions proposed in the literature for the ifcOWL (those cited in this section), with examples of the terms implemented.

TABLE 1
Proposed Ifcowl Extensions in the Literature (Examples)

| Reference | Implemented ontology/vocabulary | Other ontologies involved in the work | Examples of classes/properties implemented |
|---|---|---|---|
| Isikdag, Zlatanova and Underwood (2013) | BO-IDM | | WallPart, SlabPart, BeamPart, DoorPart, Storey_ID, OpeningDirection. |
| Pauwels et al. (2015) | | OWLList, OLO | coordinates_IfcCartesianPoint, IfcPolyLoop.Polygon, IfcPolyLine.Points, IfcCartesianPointList3D.CoordList. |
| Goméz-Romero et al. (2015) | | | Similar_to_IfcMaterial, High_IfcWindow, Wide_IfcWindow. |
| de Farias, Roxin and Nicolle (2015) | ifcWoD | | isDefinedBy_IfcObject, isPredecessorTo_IfcProcess, hasSimpleProperty, hasComplexProperty, hasReferenceValue, has SimpleValue, hasListValue, hasEnumeratedValue, hasTableValue, hasBoundedValue. |
| Pauwels and Roxin (2016) | SimpleBIM | | Area, Base_Constraint, Base_Extension_Distance, Base_Offset, Base_is_Attached, Category, Enable_Analytical_Model. ExtendToStructure, Family, Family_and_Type, isExternal, Length, LoadBearing, Location_line, Phase_Created, Reference, Related_To_Mass. Room_Bounding, Structural, Structural_Usage. |
| Belsky, Sacks and Brilakis (2016) | | | is_made_of, is_a, is_related_to, is_part_of, Face, orientation, proportion, is_adjacent_to, create_relationship. |
| Zhang, Beetz and de Vries (2018) | BimSPARQL | | hasBodyGeometry, hasAABB, hasMVBB, hasOverallHeight, hasSurface, hasUpperSurface, hasVolume, hasSpaceArea, hasWindowArea, hasGrossWallArea, touches, disjoints, intersects, within, equals,contains, distance, distanceZ, distanceXY, hasUpperStorey, isLocatedInStorey. |
| Chbeir et al. (2019) | OntoH2G | SSN, QUDT, W3C Time Ontology, W3CGeo | Occupant, Action, Need, Driver, BuildingInfrastructure, BuildingAppliance, BuildingState, EconomicProperty, PhysicsProperty, OccupantStateProperty. |

## 2. MATERIAL AND METHODS

In this section, the authors will describe the system implementation process, which includes the use of ifcOWL to support well-known techniques in indoor navigation. This description also includes a brief analysis of the software tools used in the process.

Before starting with this description, it is important to emphasize that, in the context of this project, the IFC files can be used as a source model that will be complemented with the information provided by the different sensors included in the navigation system. In this sense, some recent works already propose extracting the information contained in an IFC file to determine evacuation routes for people with mobility problems [61] or support for robot navigation [62, 63]. An interesting work in this regard is that of Taneja et al. [64], who describe a series of algorithms for automatically generating indoor navigation models from BIM. However, the scarcity of references in the literature shows that the benefits of this application have not been sufficiently exploited, especially as more and more BIM models become available (especially in public buildings). In the context of this work, the information contained in the IFC file can be used to support the construction of a (ROS) navigation map [65]. A way to do this is to extract a map in SVG format from the IFC file and export it to a graphic format compatible with ROS, like PNG files. Conversion tools such as ifcConverter [66] can be used to generate the SVG file. An interesting feature of this tool is that it allows using command lines to select the classes of elements to be displayed in the output. As an example, Figure 2 shows the conversion of the IFC file for the duplex apartment model available in [67]. However, despite this interesting contribution from IFC to the project, the aim of this paper is to focus on the use of techniques that take advantage of the semantic approach.
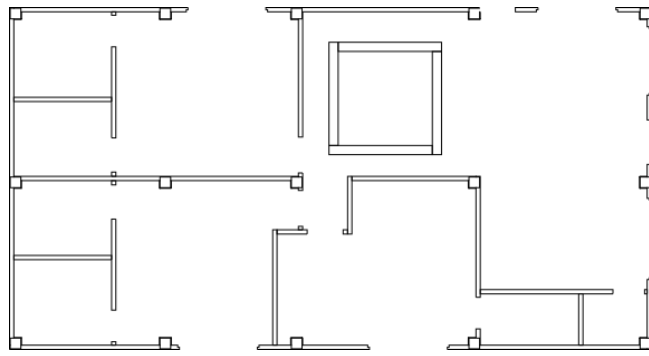
FIGURE 2. Example of IFC to SVG conversion

Figure 3 shows the proposed flow for the system implemented with the software applications used. RDF statements are obtained from the supplied IFC file. These statements (and those that could be formed from the observations made by the system sensors) will be treated together with the ifcOWL ontology and a set of items (classes, properties and individuals) proposed as an extension of the ontology. The system implemented performs knowledge inference operations on the resulting items. The result of these operations is included in the ontology. If the user communicates their intention to carry out one of the anticipated actions through the corresponding adapted interface, a query will be made to the ontology module, which will respond with a list with the locations of possible destinations. The authors will describe the system in greater detail in what remains of the section. As we can see, this method seeks an improved representation of the navigation environment. In this way, strictly geometrical data is included, as well as high-level qualitative information and relationships with other objects in the environment, as indicated in [68]. Thanks to this approach, semantic maps can be used as an element to improve the navigation and as a basis for decision-making. The ontology will be nourished both by the information contained in the IFC file and by the information collected by the different sensors available after the recognition algorithms are applied to the data provided. Once this information is stored, the ontology can interact with the semantic map to make navigation decisions. This proposal differs from the references mentioned above, which are more oriented toward improving searches within IFC files and toward static ontologies in terms of their content.
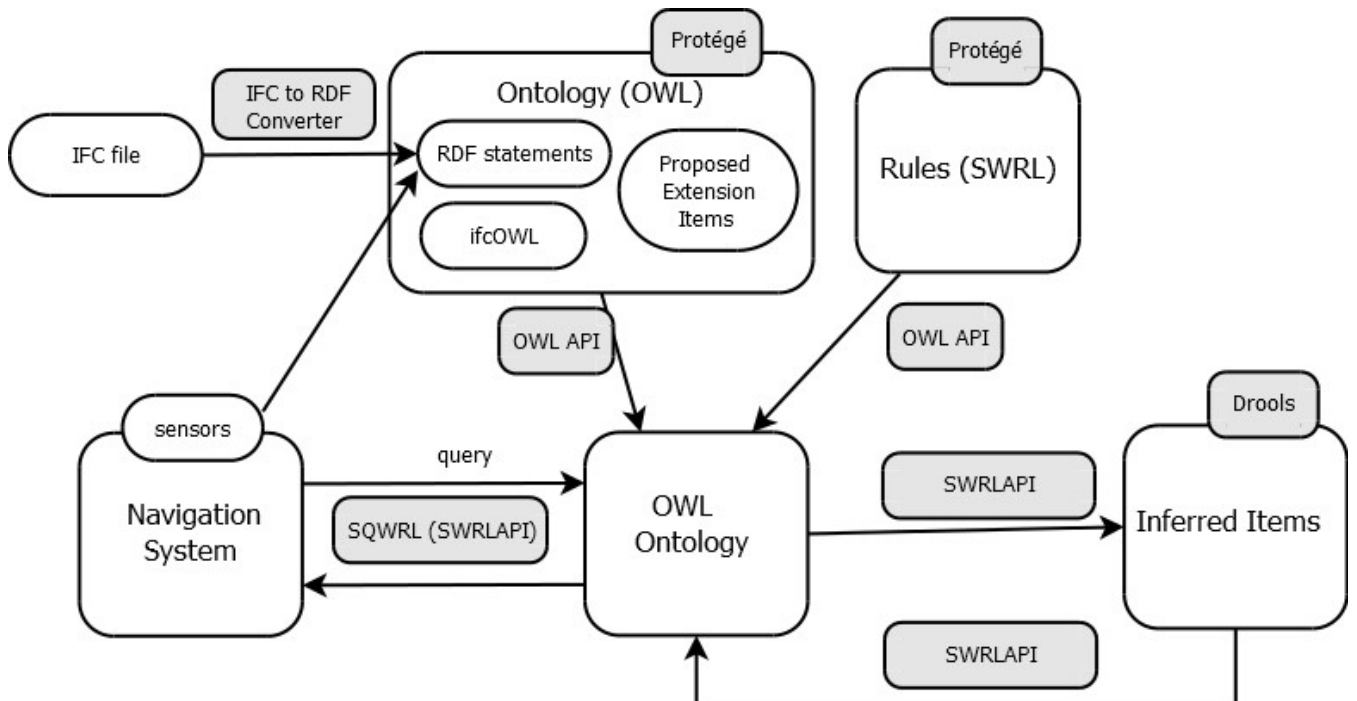


FIGURE 3. Proposed flow in the system implemented.

The functionality is achieved by expanding the ifcOWL ontology with simple but powerful modifications, as will be seen later in this section. The method applied also allows for the inclusion of new elements in a simple way. Accordingly the authors wish to show the potential of reusing ontologies (initially designed for other industries) in the field of mobile Robotics. In this sense, Table 2 shows the list of namespaces used by the ontology (from IFC version 2.3).

TABLE 2
Prefixes and Namespaces for the Ontology Implemented (from IFC2.3)

| Prefix | Namespace | Comment |
|---|---|---|
| xml | http://www.w3.org/XML/1998/namespace | XML namespace |
| xsd | http://www.w3.org/2001/XMLSchema# | XML Schema namespace |
| rdf | http://www.w3.org/1999/02/22-rdf-syntax-ns# | RDF namespace |
| rdfs | http://www.w3.org/2000/01/rdf-schema# | RDF Schema namespace |
| owl | http://www.w3.org/2002/07/owl# | OWL namespace |
| swrl | http://www.w3.org/2003/11/swrl# | SWRL namespace |
| swrla | http://swrl.stanford.edu/ontologies/3.3/swrla.owl# | SWRL namespace (isRuleEnabled) |
| swrlb | http://www.w3.org/2003/11/swrlb# | SWRL namespace (built-ins) |
| geo | http://www.w3.org/2003/11/geo# | Namespace for ad hoc built-in |
| ifcowl | https://standards.buildingsmart.org/IFC/DEV/IFC2x3/TC1/OWL# | IFC namespace (v 2.3) |
| list | https://w3id.org/list# | LIST ontology namespace |
| expr | https://w3id.org/express# | EXPRESS namespace |
| inst | https://w3id.org/instances# | Namespace for RDF instances and extended items |
| words | http://www.semanticweb.org/e/ontologies/2019/3/wordsontology87# | Namespace for command processing |

One of the first tools to be used in the implementation must be an ontology editor/viewer. Although there are other tools for developing and implementing ontologies, the authors have used Protégé [69, 70]. Next, in order to take advantage of ifcOWL, an IFC2RDF translator [71] is used to obtain RDF graphs, which can be used to manage the stored information in a more practical way by using reasoning techniques developed for the SW. Its operation required some minor modifications to the code provided by the developer. Some of these modifications involve the use of namespaces in ontologies, since namespaces for ontologies have changed, under the impulse of the Technical Room of buildingSMART International. Another necessary modification is the redefinition of the annotation axioms for the inference rules to work. RDF Turtle syntax is used for the output files.

To simplify the management of the ontology, we followed a strategy similar to that posed in [8], in particular in terms of the Cartesian coordinates and rotation angles in the relative direction of the axes. As mentioned above, the ifcOWL ontology makes use of an intricate system based on RDF lists. In this work, these lists are converted to a WKT in the IFC2RDF translator runtime. That WKT is accessed for each instance through a data property called inst:hasValueAsWKT, defined ad hoc for this work. Although there are specific programs and other ways of calculating the global position of the elements, the authors have opted for the way described so as to achieve a more direct integration with the rest of the elements and at the same time show the potential of the ontology. However, by their nature, ontologies can be rapidly integrated with other elements with which they share the knowledge domain.

The positions of the elements in IFC are defined relative to another element, except in an IfcProduct instance (more specifically, an instance of one of its subclasses) that does not have the corresponding property defined. However, an alternative to this chain system of relative positions would be desirable for navigation. The option proposed in this work is to take advantage of the strength of semantics in ontology to convert relative positions into global ones, thus facilitating the determination of the position of the navigation system and the elements of its environment. In this respect, the following data properties are defined as an extension of the ifcOWL ontology.

$$\text{inst:hasGlobalPosition}^{I} \subseteq \text{ifcowl:IfcProduct}^{I} \times \text{xsd:string}^{I}. \tag{1}$$

$$\text{inst: hasNoPlacementRelTo}^{I} \subseteq (\text{ifcowl:IfcProduct}^{I} \cup \text{ifcowl:IfcGridPlacement}^{I}) \times \text{xsd:boolean}^{I}. \tag{2}$$

The data property inst:hasGlobalPosition indicates the global position of each instance of a subclass of IfcProduct with respect to the instance taken as origin in the navigation map. This position is stored as a string, similar to a WKT. The second property, inst:hasNoPlacementRelTo, has a Boolean value as range that indicates if it is the element without relative reference.

A reason for using the ifcOWL is to take advantage of its structured information by including ontological deduction rules. To this end, the authors will use the Semantic Web Rule Language (SWRL) [72], proposed by W3C, which allows editing Horn-type rules that include statements expressed in OWL. edition is done through the SWRLTab of the Protégé tool. As we will see later, the nature of the problem of indoor navigation is adequately adapted to this approach. Using SWRL involves selecting an inference engine that allows executing those rules, deducing new statements about the knowledge domain and including them as facts in the ontology. Taking into account the characteristics indicated in [73], the authors have opted for SWRLAPI [74], a set of tools based on OWLAPI for efficiently managing SWRL rules. These definitions make it possible to calculate the global positions of the elements in the navigation environment. To do this, the SWRL rules shown in Table 3 are defined. These rules rely on a built-in property, implemented by the authors for this work, that takes advantage of the modularity of SWRLAPI. This built-in property, called geo:globalPosition, allows calculatig the global position of the IfcProduct when the global position of the instance with respect to which its relative position is defined is known. In this way, the calculation of the global position of each IfcProduct instance is propagated. This design of the rules takes into account two important considerations of the SWRL language: the Open World Assumption and the monotonic inference. According to the former, if an instance has not explicitly defined a value for a property, it can be for two reasons: because it really has no value for that property or because that information is missing in the ontology. In other words, every individual in the ontology is supposed to have a certain value for a property, unless explicitly stated otherwise. As for monotonic inference, it implies that the retraction or modification of statements is not supported [72].

TABLE 3
SWRL rules used for element global position

| Rule ID | SWRL rule |
|---|---|
| S1 | ifcowl:IfcProduct(?x) ^ ifcowl:objectPlacement_IfcProduct(?x, ?y) ^ inst:hasNoPlacementRelTo(?y, true) ^ ifcowl:relativePlacement_IfcLocalPlacement(?y, ?z) ^ ifcowl:location_IfcPlacement(?z, ?c) ^ inst:hasValueAsWKT(?c, ?d) -> inst:hasGlobalPosition(?x, ?d) |
| S2 | ifcowl:IfcProduct(?x) ^ ifcowl:objectPlacement_IfcProduct(?x, ?y) ^ ifcowl:relativePlacement_IfcLocalPlacement(?y, ?z) ^ ifcowl:location_IfcPlacement(?z, ?c) ^ inst:hasValueAsWKT(?c, ?d) ^ ifcowl:refDirection_IfcAxis2Placement3D(?z, ?r) ^ ifcowl:axis_IfcAxis2Placement3D(?z, ?ax) ^ inst:hasValueAsWKT(?ax, ?v1) ^ inst:hasValueAsWKT(?r, ?v2) ^ ifcowl:placementRelTo_IfcLocalPlacement(?y, ?rel) ^ ifcowl:IfcProduct(?x1) ^ ifcowl:objectPlacement_IfcProduct(?x1, ?rel) ^ inst:hasGlobalPosition(?x1, ?pos1) ^ geo:globalPosition(?result, ?pos1, ?d, ?v1, ?v2) ^ differentFrom(?x, ?x1) -> inst:hasGlobalPosition(?x, ?result) |

To improve indoor navigation, it would be useful to know the location of the different types of rooms in the environment: kitchen, bedroom, toilet, and so on. This information is not explicitly defined in the IFC files, and thus not in the ifcOWL ontology either. However, the scheme allows for the inclusion of characteristic elements of some rooms: microwave, refrigerator, dishwasher, toilet tank, etc. By determining the global position of each element, it would be possible to locate the probable global position of each type of room in the navigation environment. An important factor to work with is determining the location of each element in the hierarchy tree of the ifcOWL ontology. This is not an easy operation, however, since here the implementer has to deal with the intricacies of language, which often make it cumbersome to determine the different types and classes. In addition, the authors recommend being very careful with the changes between the different versions of IFC. For example, a microwave is an instance of the class ifcowl:IfcFlowTerminal whose value for the property ifcowl:IfcElectricApplianceType is ifcowl:MICROWAVE. This value corresponds to one of the possible items predefined for the property, which are defined in the ontology as instances of the class ifcowl:IfcElectricApplianceTypeEnum. As an example, Table 4 shows the instances defined for ifcowl:IfcElectricApplianceTypeEnum.

TABLE 4
Instances defined for ifcowl:IfcElectricApplianceTypeEnum in ifcOWL ontology

| | | |
|---|---|---|
| ifcowl:COMPUTER | ifcowl:HANDDRYER | ifcowl:TELEPHONE |
| ifcowl:DIRECTWATERHEATER | ifcowl:INDIRECTWATERHEATER | ifcowl:TUMBLEDRYER |
| ifcowl:DISHWASHER | ifcowl:MICROWAVE | ifcowl:TV |
| ifcowl:ELECTRICCOOKER | ifcowl:NOTDEFINED | ifcowl:USERDEFINED |
| ifcowl:ELECTRICHEATER | ifcowl:PHOTOCOPIER | ifcowl:VENDINGMACHINE |
| ifcowl:FACSIMILE | ifcowl:PRINTER | ifcowl:WASHINGMACHINE |
| ifcowl:FREESTANDINGFAN | ifcowl:RADIANTHEATER | ifcowl:WATERCOOLER |
| ifcowl:FREEZER | ifcowl:REFRIGERATOR | ifcowl:WATERHEATER |
| ifcowl:FRIDGE_FREEZER | ifcowl:SCANNER | |

For the implementation, the authors have followed the methodology of defining express:ENUMERATION subclasses. The ontology has been extended by defining a subclass, inst:RoomTypeEnum.

$$inst:RoomTypeEnum^{I} \subseteq express:ENUMERATION^{I} \tag{3}$$

This enumeration will contain instances referring to each type of room to be implemented. In the case of the kitchen, it would be formally defined as

$$inst:RoomTypeEnum(inst:KITCHEN)^{I} \tag{4}$$

There are currently four room types implemented (inst:BATHROOM, inst:BEDROOM, inst:KITCHEN and inst:OTHER_ROOM), although it is easily expandable to other types. For proper performance, these individuals must be explicitly declared as different from each other by means of the built-in differentFrom. This type of room can be linked to the instances of its characteristic elements by means of a new object property defined in the ontology: inst:hasUsualLocation.

$$inst:hasUsualLocation^{I} \subseteq ifcowl:IfcProduct^{I} \text{ x } inst:RoomTypeEnum^{I}. \tag{5}$$

This definition can be used to implement a set of SWRL inference rules for associating elements with their usual location. Thus, for example, the following rule indicates that every microwave instance has the kitchen as its usual place.

$$\begin{aligned} &ifcowl:IfcProduct(?x) \text{ } \wedge \text{ } ifcowl:predefinedType\_IfcElectricApplianceType \text{ } (?x, \text{ } ifcowl:MICROWAVE) \text{ } -> \\ &inst:hasUsualLocation(?x, inst:KITCHEN). \end{aligned} \tag{6}$$

Accordingly, Table 5 shows a set of common positions of elements that can be assumed in a first version of the prototype implemented, for kitchen and bathroom instances.

TABLE 5
Common positions of elements assumed in the prototype implemented
(ifcowl prefix has been removed for the sake of simplicity)

| Ifcowl:IfcProduct Individual | inst:RoomTypeEnum Individual |
|---|---|
| Dishwasher, Electric Cooker, Fridge Freezer, Hand Dryer, Kitchen Exhaust, Microwave, Refrigerator, Sink | inst:KITCHEN |
| Bath, Toilet tank, Sanitary Fixture, Shower, Toilet Pan, Urinal, Wash Hand Basin, WC Seat | Inst:BATHROOM |

An important advantage of using ifcOWL is being able to search the ontology. In practice, two search languages are used: SPARQL and SQWRL. The SPARQL [75] language is widely used, but it has the problem [74] of not fully understanding OWL semantics. On the other hand, SQWRL (Semantic Query-enhanced Web Rule Language) [76] is an OWL-specific search language with a more compact and easy-to-use syntax, but the authors believe ifcOWL has not been used enough.

The extension proposed in this work for the ifcOWL ontology allows implementing SQWRL searches like the one shown in Table 6. In this case, the prototype wants to retrieve the list of coordinates of all elements (IfcProduct instances) whose usual location is the kitchen.

TABLE 6
Example of SQWRL search for the system implemented

| Search ID | SQWRL expression |
|---|---|
| Q1 | ifcowl:IfcProduct(?x) ^ inst:hasUsualLocation(?x, inst:KITCHEN) ^ inst:hasGlobalPosition(?x, ?position) -> |

As indicated above, the intent of this implementation of the approach is to support indoor navigation. In particular, the authors have applied these statements to a semi-autonomous wheelchair. In the current prototype (described in Section 3), the goal is for the ontology to allow suggesting a destination position when an action is required by the user. In this case, a user command reception system similar to the one described in [77] is implemented, which includes a mechanism to check if the reception time of each word of the order is below a previously defined maximum. Each type of input device (joystick, eye tracker, voice recognizer, etc.) will have a different maximum time. Any order received that exceeds this minimum time will be rejected. This time control will be carried out by a module other than the ontology module.

The result of that processing is also treated by a simple but effective ontological approach in a separate module, in order to improve system performance. To do this, three main classes are defined: words:Sentence, words:Word and words:Destination. Each sentence (command from the processing module) will be composed of a set of words and a result that will be the probable destination for the action that will be relayed to the rest of the modules of the wheelchair. Thus, each sentence received creates an individual with the corresponding object properties, in which each word is included in the ontology by means of an individual of the class words:Word. From the superclass words:Word, a series of subclasses is defined relative to the different processable actions and to the locations: DesireAction, FridgeObject, KitchenAction, etc. As for the properties, the object properties listed in Table 7 are defined, with their corresponding ranges and domains.

TABLE 7
Defined object properties for sentence processing

| Object Property | Domain | Range |
|---|---|---|
| words:hasDestination | words:Sentence | words:Destination |
| words:isAfter | words:Word | words:Word |
| words:startsWith | words:Sentence | words:Word |

The fact that the words:isAfter property is defined as transitive facilitates the normalization process, since in both the phrase "I want to drink water" and "I want to drink a glass of water", the word "water" is after the word "to drink". The data property words:hasString is also defined as follows:

$$\text{words:hasString}^I \subseteq \text{words:Word}^I \times \text{xsd:string}^I. \tag{7}$$

The value of this property contains the corresponding string of each word in the sentence.

The ontology is complemented with a set of predefined individuals, which will help to match each word from the command with its category (formally with the subclass of words: Word). Thus, for example, the EatAction individual is defined as follows:

$$\text{words:EatAction(words:KitchenAction)}^I \tag{8}$$

The definition of these entities allows SWRL rules to be defined for determining the probable destination where the action requested by the User will be carried out. As an example, Table 8 shows the SWRL rule for planned actions with a fridge as the destination. Thus, a command received with the text *"I want to drink some beer, please"*, would activate this rule.

Note that the proposed approach is compatible with situations where the IFC file is not available, since elements of the navigation environment can be included either manually or as the wheelchair sensors discover these elements (one way to do this is by recognizing scenes through the cameras that determine what type of room the system is in). This inclusion of new information could be done on the fly, without the need to recompile any code. In addition, the way the information is stored means that it can be exchanged with other devices in the environment by taking advantage of Internet of Things (IoT) technology. Finally, the ontological approach is also compatible with actions such as ignoring events not contemplated in the design and complementary situations based, for example, on the user's health status.

TABLE 8
Example of SWRL rule for planned actions with a fridge as the destination

| Search ID | SWRL expression |
|---|---|

| | |
|---|---|
| R1 | words:Sentence(?x) ^ words:startsWith(?x, ?w1) ^ words:DesireAction(?dw2) ^ words:KitchenAction(?dw3) ^ words:FridgeObject(?dw4) ^ words:isAfter(?w2, ?w1) ^ words:isAfter(?w3, ?w2) ^ words:isAfter(?w4, ?w3) ^ words:hasString(?w2, ?s2) ^ words:hasString(?w3, ?s3) ^ words:hasString(?w4, ?s4) ^ words:hasString(?dw2, ?ds2) ^ words:hasString(?dw3, ?ds3) ^ words:hasString(?dw4, ?ds4) ^ swrlb:equal(?ds2, ?s2) ^ swrlb:equal(?ds3, ?s3) ^ swrlb:equal(?ds4, ?s4) -> words:hasDestination(?x, words:FridgeDestination) |

## 3. RESULTS

The authors have carried out several simulation tests in which the indoor navigation environment is defined by the set of demo files supplied by various tools and IFC repositories. An example file of these sites is shown in Figure 4,. The tests were conducted using an Intel® Core ™ i7-2600 CPU @ 3.40 GHz processor, 8GB RAM.
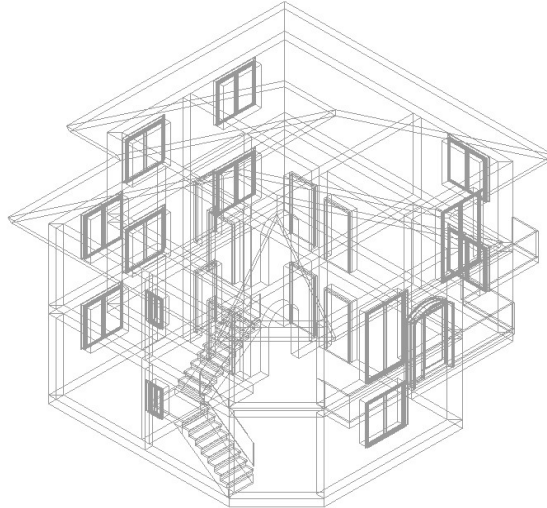


FIGURE 4.  Example of IFC file (supplied by usBIM).

It is important to remark that, although the approach proposed in this paper is general, its development has taken into account the particular case of a smart wheelchair, as noted above. In particular, the prototype described in depth in [78] was used (Figure 5).

FIGURE 5. Prototype of smart wheelchair implemented

However, one of the problems to solve is the large amount of information generated from the ifcOWL, which even exceeds the information deduced by the SWRLAPI engine. As an example of this immense amount of information, the example file "Hello Wall" available in [79] has been used. This simple file (with 163 lines of text) consists of only three elements, IfcWallStandardCase, IfcOpeningElement, and IfcWindowand. This example has been used in this work as a scheme to add elements and obtain results. Table 9 shows an extract of the ontology metrics before and after the use of SWRLAPI in this case.

TABLE 9
Effect of applying SWRLAPI to the "Hello Wall" (+ifcOWL) ontology (v 2.3)

| Item | Before SWRLAPI | After SWRLAPI |
|---|---|---|
| Axiom | 18889 | 75206 |
| Logical axiom count | 12605 | 68922 |

An alternative is to decrease the size of the ontology by removing those entities that have no impact on the application domain. This yields a more workable subset of the ontology. In the case of indoor navigation, for example, it can be assumed that information such as the material of which a wall is constructed or the persons involved in the construction of the building could be ignored in the processing of the ontology. In this work, the original ifcOWL ontology has been simplified, resulting in an ontology whose main elements are:
  - Elements referred to the location.
  - Subclasses of the IfcProduct class.
  - Elements proposed as an extension of the ifcOWL ontology in the framework of this work.
  - Subclasses of express:ENUMERATION.
  - Object properties of predefined types.

Table 10 shows the number of axioms resulting from this simplification. A comparison with Table 9 reveals a significant reduction in the number of axioms included in the ontology. This simplification also implies a considerable reduction in the time needed to process the operations in the ontology. The variation in the corresponding items is indicated in parentheses.

TABLE 10
Effect of applying SWRLAPI on the "Hello Wall" (+ifcOWL) ontology (Simplified)

| Item | Before SWRLAPI | After SWRLAPI |
|---|---|---|
| Axiom | 6005 (↓68.2%) | 10037 (↓86.7%) |
| Logical axiom count | 3514 (↓73,1%) | 7145 (↓89,6%) |

The performance of this simplified ontology (which could be reduced even more in terms of the number of axioms) was analyzed with respect to the processing times, in a similar way to [73]. The first analysis will focus on the time taken by the rules engine with respect to the number of instances of subclasses of the IfcProduct class. It should be noted that each inclusion of a new element implies entering a minimum of 21 axioms in the ontology: six OWLDeclarationAxiom, six OWLClassAssertionAxiom, six OWLObjectPropertyAssertionAxiom and three OWLDataPropertyAssertionAxiom. From the "Hello_Wall" example, an individual set has been included and the average values obtained in the activation of the rules are shown in Figure 6. For a number of observations inferior to 1000 individuals, the time to activate all the SWRL rules (and therefore to build the navigation map) was verified to be acceptable (less than 80 seconds). In this sense, it is necessary to clarify that this calculation operation is only carried out once, when the navigation system is initialized.
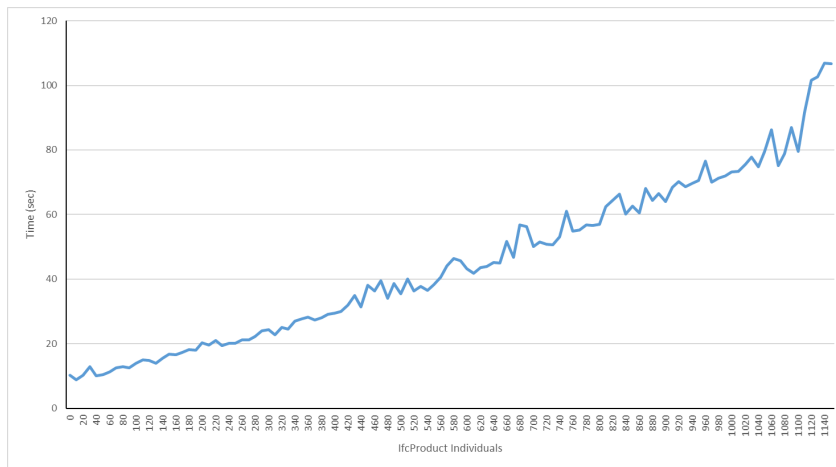


FIGURE 6. Trend in processing time for a set of new IfcProduct individuals.

In the second experiment, the trend in the processing time for the SQWRL searches is analyzed with respect to the number of individuals of the IfcProduct subclass present in the ontology. For this experiment, the search shown in Table 8 (for processing an order by the user) was selected. Figure 7 shows the result for the processing time when the number of individuals (instances of subclasses of the IfcProduct class) in the ontology is increased. As in the previous case, for a number of observations inferior to 1000 individuals, acceptable data are obtained for the problem at hand and in the interaction with the potential users. Once the query is made, the result for each room type can be stored in the System's semantic map, meaning it is not necessary to make the query again. Another alternative would be to perform an initial query for each room type defined.
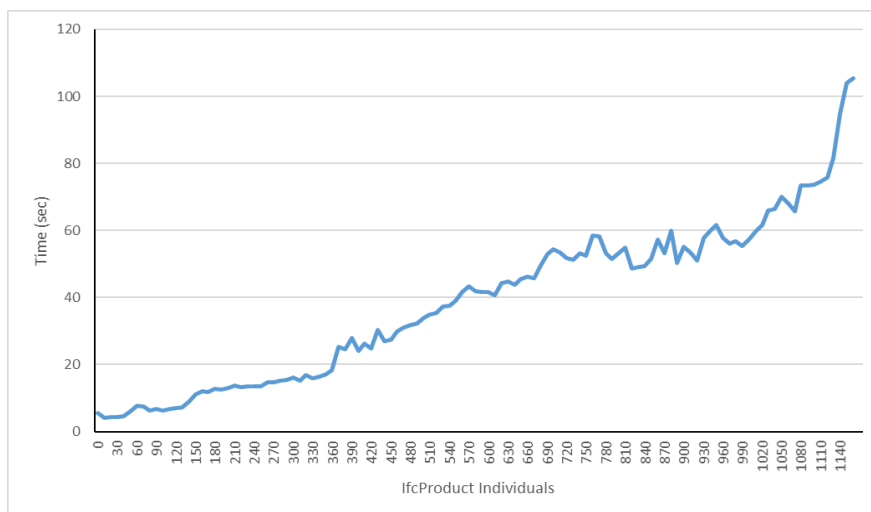
FIGURE 7. Trend in processing time for SQWRL queries.

## 4. OPEN LINES FOR FUTURE WORK

The approach presented in this paper also offers several open areas of work besides those intended to optimize the size of the resulting ontology. Most of these areas are related to the advancement of technology and the development of new approaches in the fields of AEC and indoor navigation.

For the operational extension of the ifcOWL, it is possible to resort to existing ontologies or resources (precisely one of the strengths in the field of ontologies is that they can be reused and integrated into systems). In this sense, a complementary element is the topological vision of the indoorGML, a standard proposed by the Open Geospatial Consortium (OGC) aimed at facilitating indoor navigation. Teo and Yu [80] provided an important relationship between IFC and indoorGML, adapting the previous work described in [81]. The IFC entities involved in this case are IfcSpace, IfcOpeningElement, IfcDoor, IfcOpeningElement, IfcWindow, IfcRelSpaceBoundary, IfcWall and IfcWallStandardCase. This list could serve as a starting point for better optimizing the subset of elements of the ontology if it is used in this work. The joint use of ifcOWL and indoorGML is another open line to explore new works.

Apart from this, a further area to explore is other recent advances that take advantage of the benefits of the Semantic Web in the field of AEC to reduce the immense amount of information stored in IFC files. In this sense, an interesting approach is the Building Topology Ontology (BOT), proposed by the W3C Linked Building Data Community Group. This group defines it as "*a minimal ontology for describing the core topological concepts of a building*" [82]. This ontology is complemented by other proposals from the same development group, related to products, properties and project management. As in the case of RDF, an IFC to LBD (Linked Building Data) converter is available [83].

Another line to explore in this work is to delve a little more into the semantic relationships contained in the ontology, such that a more complex use can be made by means of, for example, the properties of the relationship between the elements of the IFC scheme shown in Section 2. The IfcRelConnects property seems particularly promising in this regard. The operation of the most promising properties will be validated and, if the results are suitable, they will be coded as ROS modules, and the ways of communicating with the module corresponding to the semantic map will be defined. Configuration options will make it possible to choose to use one or more of these algorithms at any given moment.

## 5. CONCLUSIONS

This paper presents an approach based on the ifcOWL ontology as a support for indoor navigation. To this end, a series of modifications of this ontology have been proposed that include new items, SWRL ontological deduction rules and SQWRL searches. This approach was validated using several tests, with satisfactory results. However, the authors consider it important to integrate their findings into real, more advanced prototypes in different environments. The authors also note that the proposed approach is easily expandable to other types of rooms, configurations, etc., and it should be mainly considered as a complement to other well-known indoor navigation techniques. By using this ontology, it is possible to interact quickly and easily with other elements within the same knowledge domain.

References
[1] P. Pauwels, S. Zhang and Y. -. Lee, "Semantic web technologies in AEC industry: A literature overview," Autom. Constr., vol. 73, pp. 145-165, 2017.
[2] B. Zhong, H. Wu, H. Li, S. Sepasgozar, H. Luo and L. He, "A scientometric analysis and critical review of construction related ontology research," Autom. Constr., vol. 101, pp. 17-31, 2019.
[3] P. Pauwels, W. Terkaj, T. Krijnen and J. Beetz, "Coping with lists in the ifcOWL ontology," in EG-ICE 2015 - 22nd Workshop of the European Group of Intelligent Computing in Engineering, Eindhoven, The Netherlands, pp. 111-120, 2015.
[4] P. Pauwels, S. Törmä, J. Beetz, M. Weise and T. Liebich, "Linked data in architecture and construction," Automation in Construction, vol. 57, pp. 175-177, 2015.
[5] T. M. d. Farias, A. Roxin and C. Nicolle, "IfcWoD, semantically adapting IFC model relations into OWL properties," in 32nd CIB W78 Conference, Eindhoven, The Netherlands, pp. 175-185, 2015.
[6] J. Beetz, v. J. P. Leeuwen and d. B. Vries, "An ontology web language notation of the industry foundation classes," in Proceedings of the 22nd CIB W78 Conference on Information Technology in Construction, pp. 193-198, 2005.
[7] W. Terkaj and P. Pauwels, "A method to generate a modular ifcOWL ontology," in CEUR Workshop Proceedings, , 2017.

[8] P. Pauwels, T. Krijnen, W. Terkaj and J. Beetz, "Enhancing the ifcOWL ontology with an alternative representation for geometric data," Autom. Constr., vol. 80, pp. 77-94, 2017.

[9] P. Pauwels and W. Terkaj, "EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology," Autom. Constr., vol. 63, pp. 100-133, 2016.

[10] J. Beetz, J. Van Leeuwen and B. De Vries, "IfcOWL: A case of transforming EXPRESS schemas into ontologies," Artif Intell Eng Des Anal Manuf, vol. 23, *(1),* pp. 89-101, 2009.

[11] W. Terkaj and A. Šojic, "Ontology-based representation of IFC EXPRESS rules: An enhancement of the ifcOWL ontology," Autom. Constr., vol. 57, pp. 188-201, 2015.

[12] P. Pauwels and A. Roxin, "SimpleBIM: From full ifcOWL graphs to simplified building graphs," in eWork and eBusiness in Architecture, Engineering and Construction - Proceedings of the 11th European Conference on Product and Process Modelling, ECPPM 2016, pp. 11-18, 2016.

[13] P. Pauwels, T. M. de Farias, C. Zhang, A. Roxin, J. Beetz, J. De Roo and C. Nicolle, "A performance benchmark over semantic rule checking approaches in construction industry," Adv. Eng. Inf., vol. 33, pp. 68-88, 2017.

[14] P. Bonsma, I. Bonsma, A. E. Ziri, S. Parenti, P. M. Lerones, J. L. Hernández, F. Maietti, M. Medici, B. Turillazzi and E. Iadanza, "INCEPTION standard for heritage BIM models," *Lect. Notes Comput. Sci.,* vol. 10058 LNCS, pp. 590-599, 2016.

[15] M. Boysen, C. De Haas, H. Lu and X. Xie, "A journey from IFC files to indoor navigation," *Lect. Notes Comput. Sci.,* vol. 8470 LNCS, pp. 148-165, 2014.

[16] U. Isikdag, S. Zlatanova and J. Underwood, "A BIM-Oriented Model for supporting indoor navigation requirements," Comput. Environ. Urban Syst., vol. 41, pp. 112-123, 2013.

[17] D. P. Kun, E. Baksáné Varga and Z. Tóth, "Ontology based indoor navigation service for the ILONA system," Infocommunications J., vol. 10, *(3),* pp. 20-29, 2018.

[18] P. M. Dudas, M. Ghafourian and H. A. Karimi, "ONALIN: Ontology and algorithm for indoor routing," in Proceedings - IEEE International Conference on Mobile Data Management, pp. 720-725, 2009.

[19] G. Sriharee, "A symbolic-based indoor navigation system with direction-based navigation instruction," in Procedia Computer Science, pp. 647-653, 2015.

[20] C. Anagnostopoulos, V. Tsetsos, P. Kikiras and S. P. Hadjiefthymiades, "OntoNav: A semantic indoor navigation system," in CEUR Workshop Proceedings, , 2005.

[21] J. Scholz and S. Schabus, "An indoor navigation ontology for production assets in a production environment," Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 8728, pp. 204-220, 2014.

[22] T. Berners-Lee and J. Hendler, "Publishing on the semantic web," Nature, vol. 410, *(6832),* pp. 1023-1024, 2001.

[23] T. Berners-Lee, J. Hendler and O. Lassila, "The semantic web," Sci. Am., vol. 284, *(5),* pp. 34-43, 2001.

[24] N. Guarino, "Formal ontology, conceptual analysis and knowledge representation," International Journal of Human - Computer Studies, vol. 43, *(5-6),* pp. 625-640, 1995.

[25] R. Studer, V. R. Benjamins and D. Fensel, "Knowledge Engineering: Principles and methods," Data Knowl. Eng., vol. 25, *(1-2),* pp. 161-197, 1998.

[26] T. R. Gruber, "A translation approach to portable ontology specifications," Knowledge Acquisition, vol. 5, *(2),* pp. 199-220, 1993.

[27] B. Swartout, R. Patil, K. Knight and T. Russ, "Toward distributed use of large-scale ontologies," in Ontological Engineering, AAAI-97 Spring Symposium Series, pp. 138-148, 1997.

[28] E. Blomqvist and A. Öhgren, "Constructing an enterprise ontology for an automotive supplier," Eng Appl Artif Intell, vol. 21, *(3),* pp. 386-397, 2008.

[29] M. Uschold and M. Gruninger, "Ontologies: Principles, methods and applications," Knowl. Eng. Rev., vol. 11, *(2),* pp. 93-136, 1996.

[30] I. García Rodríguez, "Modelos de conocimiento basados en ontologías para la construcción de software en el dominio de la Ingeniería de control," Universidad de León, 2012.

[31] R. B. Altman, M. Bada, X. J. Chai, M. W. Carillo, R. O. Chen and N. F. Abernethy, "RiboWeb: An Ontology-Based System for Collaborative Molecular Biology," IEEE Intell Syst their Appl, vol. 14, *(5),* pp. 68-76, 1999.

[32] J. Ye, L. Coyle, S. Dobson and P. Nixon, "Ontology-based models in pervasive computing systems," Knowl. Eng. Rev., vol. 22, *(4),* pp. 315-347, 2007.

[33] *World Wide Web Consortium (W3C): Resource Description Framework (RDF)*. Available: https://www.w3.org/RDF/ (accessed on 2019-03-04).

[34] A. Rector, N. Drummond, M. Horridge, J. Rogers, H. Knublauch, R. Stevens, H. Wang and C. Wroe, "OWL pizzas: Practical experience of teaching OWL-DL: Common errors and common patterns," in Lecture Notes in Artificial Intelligence (Subseries of Lecture Notes in Computer Science), pp. 63-81, 2004.

[35] *World Wide Web Consortium (W3C): OWL Web Ontology Language Guide*. Available: http://www.w3.org/TR/owl-guide/ (accessed on 2019-03-04).

[36] L. Ma, Y. Yang, Z. Qiu, G. Xie, Y. Pan and S. Liu, "Towards a complete OWL ontology benchmark," Eswc 2006.Lncs, pp. 125-139, 2006.

[37] NBIMS-US™, "National BIM standard-united states® (NBIMS-US™) fact sheet," 2015.

[38] BSI Group, "BSI PAS 1192-5:2015: Specification for security-minded building information modelling, digital built environments and smart asset management." 2015.

[39] ISO 29481-1:2010, "Building information modelling – information delivery manual – part 1: Methodology and format," 2010.

[40] M. Yalcinkaya and V. Singh, "Building information modeling (BIM) for facilities management-literature review and future needs," *IFIP Advances in Information and Communication Technology,* vol. 442, pp. 1-10, 2014.

[41] M. Niknam and S. Karshenas, "Sustainable design of buildings using semantic BIM and semantic web services," in Procedia Engineering, pp. 909-917, 2015.

[42] T. M. Farias, A. Roxin and C. Nicolle, "COBieOWL, an OWL Ontology based on COBie standard," *Lect. Notes Comput. Sci.,* vol. 9415, pp. 361-377, 2015.

[43] *buildingSMART web page*. Available: https://www.buildingsmart.org/ (accessed on 2019-04-03).

[44] C. Zhang, J. Beetz and B. De Vries, "BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data," Semantic Web, vol. 9, *(6),* pp. 829-855, 2018.

[45] J. Gómez-Romero, F. Bobillo, M. Ros, M. Molina-Solana, M. D. Ruiz and M. J. Martín-Bautista, "A fuzzy extension of the semantic Building Information Model," Autom. Constr., vol. 57, pp. 202-212, 2015.

[46] M. Belsky, R. Sacks and I. Brilakis, "Semantic Enrichment for Building Information Modeling," Comput. -Aided Civ. Infrastruct. Eng., vol. 31, *(4),* pp. 261-274, 2016.

[47] W. Terkaj and G. P. Viganò, "Semantic GIOVE-VF: An ontology-based virtual factory tool," in CEUR Workshop Proceedings, , 2017.

[48] J. Wix and T. Liebich, "IFC: Current reality and future potential," Proc. ISPE Int. Conf. Concurrent Eng., pp. 679-686, 2003.

[49] *buildingSMART International Limited: IFC4 Official Release*. Available: http://www.buildingsmart-tech.org/ifc/IFC4/final/html/ (accessed on 2019-03-04).

[50] K. Häfele, J. Benner, N. Nisbet, M. Weise and A. Geiger, "Recommendation to buildingSMART, OGC, WC3 on open standards improvement," Streamer consortium, Tech. Rep. D8.8, 2017.

[51] X. Liu, Z. Li and S. Jiang, "Ontology-based representation and reasoning in building construction cost estimation in China," Future Internet, vol. 8, *(3),* 2016.

[52] R. Barbau, S. Krima, S. Rachuri, A. Narayanan, X. Fiorentini, S. Foufou and R. D. Sriram, "OntoSTEP: Enriching product model data using ontologies," CAD Comput Aided Des, vol. 44, *(6),* pp. 575-590, 2012.

[53] S. Cursi, D. Simeone and U. M. Coraglia, "An ontology-based platform for BIM semantic enrichment," in ShoCK – Sharing of Computable Knowledge - Proceedings of the 35th International Conference on Education and Research in Computer Aided Architectural Design in Europe, Rome, Italy, pp. 649-656, 2017.

[54] Y. -. Lee, C. M. Eastman and W. Solihin, "An ontology-based approach for developing data exchange requirements and model views of building information modeling," Adv. Eng. Inf., vol. 30, *(3),* pp. 354-367, 2016.

[55] T. Krijnen and J. Beetz, "A SPARQL query engine for binary-formatted IFC building models," Autom. Constr., vol. 95, pp. 46-63, 2018.

[56] Z. Ma and Z. Liu, "Ontology- and freeware-based platform for rapid development of BIM applications with reasoning support," Autom. Constr., vol. 90, pp. 1-8, 2018.

[57] H. Schevers and R. Drogemuller, "Converting the industry foundation classes to the web ontology language," in Proceedings - First International Conference on Semantics, Knowledge and Grid, SKG 2005, Beijing, China, , 2005.

[58] T. M. D. Farias, A. Roxin and C. Nicolle, "A rule-based methodology to extract building model views," Autom. Constr., vol. 92, pp. 214-229, 2018.

[59] R. Chbeir, Y. Cardinale, A. Corchero, P. Bourreau, K. Salameh, N. Charbel, L. Kallab, C. Angsuchotmetee and G. Calis, "OntoH2G: A semantic model to represent building infrastructure and occupant interactions," *Smart Innov. Syst. Technol.,* vol. 131, pp. 148-158, 2019.

[60] N. Bus, A. Roxin, G. Picinbono and M. Fahad, "Towards french smart building code: Compliance checking based on semantic rules," in CEUR Workshop Proceedings, pp. 6-15, 2018.

[61] B. Strug and G. Slusarczyk, "Reasoning about accessibility for disabled using building graph models based on BIM/IFC," Vis. Eng., vol. 5, *(1),* 2017.

[62] W. Y. Lin, P. H. Lin and H. P. Tserng, "Automating the generation of indoor space topology for 3D route planning using BIM and 3D-GIS technique," in ISARC 2017 - Proceedings of the 34th International Symposium on Automation and Robotics in Construction, National Taiwan University of Science and Technology (NTUST), Taipei, Taiwan, pp. 437-444, 2017.

[63] A. Hamieh, D. Deneux and C. Tahon, "BiMov: BIM-based indoor path planning," in *Advances on Mechanics, Design Engineering and Manufacturing : Proceedings of the International Joint Conference on Mechanics, Design Engineering {\&} Advanced Manufacturing (JCM 2016), 14-16 September, 2016, Catania, Italy*, B. Eynard, V. Nigrelli, S. M. Oliveri, G. Peris-Fajarnes and S. Rizzuti, Eds. Cham: Springer International Publishing, 2017, pp. 889-899.

[64] S. Taneja, B. Akinci, J. H. Garrett Jr. and L. Soibelman, "Algorithms for automated generation of navigation models from building information models to support indoor map-matching," Autom. Constr., vol. 61, pp. 24-41, 2016.

[65] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler and A. Y. Ng, "ROS: An open-source robot operating system," in ICRA Workshop on Open Source Software, , 2009.

[66] *IfcOpenShell: IfcConverter web site*. Available: http://www.ifcopenshell.org/ifcconvert.html (accessed on 2019-03-04).

[67] *National Institute of Building Sciences web page. Common Building Information Model Files and Tools.* Available: https://www.nibs.org/page/bsa_commonbimfiles (accessed on 2019-03-04).

[68] I. Kostavelis and A. Gasteratos, "Semantic mapping for mobile robotics tasks: A survey," Rob Autom Syst, vol. 66, pp. 86-103, 2015.

[69] H. Knublauch, M. Horridge, M. Musen, A. Rector, R. Stevens, N. Drummond, P. Lord, N. F. Noy, J. Seidenberg and H. Wang, "The Protégé OWL experience," Proc.OWL: Experiences and Directions Workshop, 2005.

[70] M. Horridge, "A practical guide to building OWL ontologies using protégé 4 and CO-ODE tools edition 1.3," The University of Manchester, 2011.

[71] *IfctoRDF converter*. Available: https://github.com/pipauwel/IFCtoRDF (accessed on 2019-03-04).

[72] I. Horrocks, P. F. Patel-Schneider, H. Boley, S. Tabet, B. Grosof and M. Dean, "SWRL: A semantic web rule language combining OWL and RuleML," W3C Member Submission, 2004.

[73] E. Gonzalez, R. Marichal and A. Hamilton, "Software Experience for an Ontology-Based Approach for the Definition of Alarms in Geographical Sensor Systems," IEEE Access, vol. 6, pp. 55556-55572, 2018.

[74] M. J. O'Connor, R. D. Shankar, C. Nyulas, A. K. Das and M. A. Musen, "The SWRLAPI: A development environment for working with SWRL rules," in OWL: Experiences and Directions (OWLED), 4th International Workshop, Washington, D.C., U.S.A., , 2008.

[75] J. Lee, M. -. Pham, J. Lee, W. -. Han, H. Cho, H. Yu and J. -. Lee, "Processing SPARQL queries with regular expressions in RDF databases," BMC Bioinformatics, vol. 12, *(SUPPL. 2),* 2011.

[76] M. O'Connor, S. Tu, C. Nyulas, A. Das and M. Musen, "Querying the semantic Web with SWRL," *Lect. Notes Comput. Sci.,* vol. 4824 LNCS, pp. 155-159, 2007.

[77] N. T. Djaid, N. Saadia and A. Ramdane-Cherif, "Multimodal fusion engine for an intelligent assistance robot using ontology," in Procedia Computer Science, pp. 129-136, 2015.

[78] R. Arnay, J. Hernandez-Aceituno, J. Toledo and L. Acosta, "Laser and Optical Flow Fusion for a Non-Intrusive Obstacle Detection System on an Intelligent Wheelchair," IEEE Sensors J., vol. 18, *(9),* pp. 3799-3805, 2018.

[79] *buildingSMART International Limited: Hello Wall Example*. Available: http://www.buildingsmart-tech.org/implementation/get-started/hello-world (accessed on 2019-03-28).

[80] T. A. Teo and S. C. Yu, "The extraction of indoor building information from bim to OGC indoorgml," in International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives, pp. 167-170, 2017.

[81] J. -. Kim, S. -. Yoo and K. -. Li, "Integrating IndoorGML and CityGML for indoor space," *Lect. Notes Comput. Sci.,* vol. 8470 LNCS, pp. 184-196, 2014.

[82] *Linked Building Data Community Group. Building Topology Ontology. Draft Community Group Report 07 January 2019*. Available: https://w3c-lbd-cg.github.io/bot/ (accessed on 2019-03-04).

[83] M. Bonduel, J. Oraskari, P. Pauwels, M. Vergauwen and R. Klein, "The IFC to linked building data converter - current status," in CEUR Workshop Proceedings, pp. 34-43, 2018.