Implementation of IFC-based process models for collaborative management of temporal scheduling through Linked Data

Davide Avogaro, davide.avogaro.2@phd.unipd.it
Department of Management and Engineering, University of Padua, Italy

Michele Berlato, michele.berlato@phd.unipd.it
Department of Management and Engineering, University of Padua, Italy

Carlo Zanchetta, carlo.zanchetta@unipd.it
Department of Civil, Environmental and Architectural Engineering, University of Padua, Italy

Abstract

Nowadays construction scheduling relies on enrichment of BIM objects with additional attributes commonly expressed as time scheduling and cost estimation parameters. Typical approaches based on the assignment of static attributes present some limitations because this kind of information cannot manage the assignment of a BIM object to multiple activities based on its involvement in multiple construction processes. Instead, an approach based on interrelated information classes allows the application of robust procedures and standardized data models. The study examines the possibility of using IFC classes both as product and process models which can be correlated according to standardizable procedures. At the same time, grouping and assigning works to digital process models makes it possible to use knowledge graphs and RDF. Additionally, IfcOWL ontology maintains IFC standardization. This aspect is essential to avoid the creation of new ontologies. Besides in the AEC Industry, real-time information coming from various sources is quite typical. As a result, cloud collaboration based on linked data models (LDM) is becoming a reference to support collaboration among different stakeholders. Moreover, it is possible to use libraries of predefined process templates related to openBIM databases. This article demonstrates how IFC protocols solve typical inconsistencies of modeling with not standardized time scheduling and cost estimation parameters, keeping the potential of LDM but offering an already operational and implementable standard rather than relying on the definition of new cost or process ontologies.

Keywords: IFC, BPMN, time scheduling, Process Models, Linked Data, IfcOWL, RDF

1 Introduction

The AEC world is highly fragmented and involves various knowledge domains. Projects are highly complex, difficult to replicate and involve many actors, complicating information exchange. One of the most valuable technologies for information management is BIM (Building Information Modeling). However, BIM software often does not cover all knowledge domains and all needs of professionals. This is why different applications are used, not always interoperable and integrable with each other. From a technical point of view, there is a problem with the data exchange format.

To overcome this problem, it is necessary to use an open standard format such as IFC. This standard is defined as "*An open international standard for Building Information Model (BIM) data that are exchanged and shared among software applications used by the various participants in the construction or facility management industry sector. The standard includes definitions that cover data required for buildings and infrastructure works over their life cycle*"(buildingSMART 2024).

The research aims to highlight the need to define information standards for process modeling and test the use of IFC-based process models for construction process planning. Process models are meant to be the digitization of construction processes in the AEC industry. In this scenario, the study highlights how, through the semantic web, information can be shared, standardized, and managed collaboratively. For these reasons and based on the research conducted around LDMs operating on IFC data models, an implementation of these classes in web-based collaborative planning is proposed.

The study is presented in the following order: Section 2 presents the literature review; Section 3 discusses the IFC classes useful for the present paper; Section 4 presents the case study and the methodology used to achieve the stated objectives; Section 5 presents the analysis of the results; Section 6 discusses the conclusions and future works.

2 Literature Review / Background

The literature review is divided into three sections. The first two analyze the application of the BPMN and IFC standards, to highlight how BPMN stands as a standard for digitizing "process logics" but does not contemplate actual process planning. Instead, the IFC data model exhibits this potential and thus also stands as a candidate standard suitable for the temporal management of construction processes. In the third section, it is presented the use of IfcOWL and related ontologies that are most relevant in the implementation of these design and construction processes.

2.1 Time management with BPMN

Business Process Modeling Notation (BPMN) is a standard for representing all the steps in a process. It is a flow chart that graphically shows the detailed sequence of activities and information required to complete a business process. BPMN is a useful tool for logical planning of tasks and should not be confused with other temporal planning tools for tasks, such as Gantt, Pert, etc. (Wix 2007). In the context of BIM-related studies dealing with BPMN, relevant experiments are pointed out below.

The first major experiment proposes a platform for monitoring construction progress in which the work plan is automatically updated by an algorithm, based on site advances or delays and available resources (Messi et al. 2021). BPMN is used to define the logic of the site processes and is integrated with time, cost, and resource information on a process database compiled independently of BPMN.

The second, on the other hand, proposes the use of BPMN-based templates to standardize process models in the AEC field so that they are not constrained to a specific software (Ismail 2022). For planning purposes, process templates should be enriched with additional information regarding required resources and estimated duration of activities, again by integrating the BPMN data model.

In conclusion, a process model can be represented with a BPMN, but it cannot contain time information. In practice, the BPMN schedule expresses only the logic of the process and not the timing of the process.

2.2 Time management with IFC

Some implementations concerning the use of IFC for time scheduling are available in the literature. Examples are given below. It should be noted that these implementations are based on IFC4, which for time-related information exhibits substantial differences from previous versions of the standard.

As a first example, a study is given in which the information required for the construction schedule management information model based on IFC4 was exhibited (Xue et al. 2015). It was shown how the use of the proposed time model enables the effective exchange and sharing of project information. The logical foundations were laid for the development of software for the temporal management of projects.

Subsequently, another IFC-based 4D construction management information model was proposed to promote the construction management of prefabricated buildings by achieving interoperability of information (Yang et al. 2021). Further analyses are needed to formalise a model that also includes cast-in-place elements, which have a more complex construction process.

Finally, a comprehensive IFC-based methodology for exchanging construction progress information has been proposed (Sheik et al. 2023). The proposed method integrates progress information, which includes not only time information, but also cost and other non-standardised information. In addition, a web application was also developed to display the progress information based on the updated IFC model.

Two concepts emerge from this review. The first is that the IFC structure allows temporal information to be exchanged and stored. In contrast to the BPMN standard, which does not allow the storage of such data. The second is that process information not only requires time-related information entities but also cost-related and resource-related ones. The IFC classes used in this article are presented in section 3.

2.3 IfcOWL and other ontologies

Initial scientific contributions reporting the benefits of the semantic web for the construction industry discuss the advantages of using this technology and the importance of using a universal structured language (Pan et al. 2004). In addition, semantic web-related services and early implementation scenarios were proposed (Zeeshan et al. 2004). These premises were concretized by the IFC2x3 version in which the EXPRESS language (Pauwels and Terkaj 2016) was translated into IfcOWL in .xml and .ttl serialization. The structure of IFC lends itself to translation into ontology, given its strongly hierarchical and relation-based structure.

In the literature, IfcOWL is perceived as very complex (Rasmussen et al. 2020; Rasmussen et al. 2019), given its high number of classes, axioms, and properties. In addition, the mechanism of assigning IFC attributes or relationships is cumbersome (Rasmussen et al. 2019). For this reason, other ontologies have been created to simplify its structure. Syntactic and semantic adaptations of the IFC model are then proposed, reducing the data considerably.

Through a literature review, the three simplification logics were identified: different classification systems, geometric simplification, and properties simplification.

Considerations of the approaches in the literature are presented below.

2.3.1 Different classification systems

Some ontologies such as BIMSO (BIM Shared Ontology) (Niknam and Karshenas 2017), use other classification systems. BIMSO is based on the UNIFORMAT II classification system. This ontology identifies the basic elements for planning development and is intended to be extended with other ontologies. The same authors developed BIMDO (BIM Design Ontology), which provides object properties describing the relationships between elements.

The W3C group has developed the BOT ontology (Building Topology Ontology), which describes the topology of buildings, including their floors, spaces, and building elements (Rasmussen et al. 2020). This lightweight ontology is also designed to interface with other ontologies. A practical application is proposed in the IFCtoLBD converter (Oraskari et al. 2018), associated with the PRODUCT and PROPS ontologies.

The problem with the creation of new ontologies is that the correspondence with the IFC schema is lost; hence, the premise of standardisation is lost.

2.3.2 Geometric simplification

Some ontologies such as SimpleBIM (Pauwels Pieter and Roxin A. 2016) simplify the model by removing information about the geometric definition of elements. This is because typically in environments based on linked data there is no intervention in the geometric modification of models.

2.3.3 Properties simplification

In ontologies such as IfcWoD (Web of DataOntology) (Mendes De Farias 2015) and SimpleBIM, (Pauwels Pieter and Roxin A. 2016), the structure of property assignment is simplified. Of all the relationships within the IFC4x3 schema, only 41 are non-abstract and non-deprecated. Of these, only 14 exhibit attributes in addition to Related and Relating, and according to these authors such over-referencing is redundancy.

By simplifying the properties and reducing their number all indications concerning Ranges and Domains are lost. By doing so, the possibility of using Reasoners is lost and one cannot be sure of having written a formally correct IFC.

A further simplification proposed concerns the assignment of attributes. SimpleBIM (Pauwels Pieter and Roxin A. 2016) simplifies the assignment of attributes by assigning them via Data Property. In contrast, IfcOWL assigns attributes via Object Property, making the graph more articulated and complex.

2.3.4 Final considerations

Criticism is often directed at the structure of IFC, not at IfcOWL, which is a faithful translation of the EXPRESS language. The structure of IFC is certainly complex, but this complexity can be resolved through software that simplifies the end-user experience but does not remove information or structure it in another standard. From a methodological point of view, the lightening of the structure should be done through the modularization of the ontology (Terkaj and Pauwels 2017) and using Model View Definition by buildingSMART (Pauwels Pieter and Roxin A. 2016).

3 IFC classes for process information

Below IFC classes inherent to the topics are discussed, to analyse their functions. In the following paragraphs, "action[1]" is understood as "the process of doing something, [...]" and "task[2]" as "a piece of work to be done, [...]".

The information presented in the following paragraphs is taken from the official page of buildingSMART (buildingSMART 2024).

3.1 IfcProcess

IfcProcess is the class designed to describe the process of constructing and storing time information. IfcProcess is an abstract entity, superclass of IfcTask, IfcProcedure, and IfcEvent.

3.1.1 IfcTask

IfcTask is used to describe specific tasks in the construction process. It is defined as a unit of work to be performed in the construction project. It is not limited to construction or installation tasks only but can also be used for other tasks related to design, moving, and commissioning.

Among the attributes of IfcTask is IfcTaskTime, which is fundamental for defining time information. IfcTaskTime allows temporal information to be expressed with attributes describing duration (IfcDuration) and with others defining the date (IfcDateTime). The standard allows time information to be compiled in a non-unique manner. In fact, the compiler has the freedom to use only date attributes, only duration attributes or both. The responsibility for the consistency of the information is left to the compiler.

Since version IFC4, the Concept Usage of IfcTask requires a Root Summary Task to be related to IfcWorkPlan and IfcWorkSchedule. The Root Summary Task is used for data organisation and not for storing typical task information. This in turn nests Summary Tasks, which in turn nest tasks. In addition, a task can be nested by IfcProcedure and IfcEvent, so that the process can be described in much more detail according to the degree of detail of the schedule.

3.1.2 IfcProcedure

BuildingSMART defines "*An IfcProcedure is a logical set of actions to be taken in response to an event or to cause an event to occur*" (buildingSMART 2024). The authors of this article believe that an IfcProcedure is an action that causes or responds to an event and belongs to a logical set of actions and/or events. In fact, an IfcProcedure in the IFC data model must be a single action, not a set of actions.

IfcProcedure contains no temporal information and has no attributes that can contain temporal information.

[1] https://dictionary.cambridge.org/dictionary/english/action
[2] https://dictionary.cambridge.org/dictionary/english/task

3.1.3 IfcEvent

An IfcEvent is an event that occurs in response to or triggers an action or task. Particularly used in schedules, IfcEvent is used to capture information about particular events in the schedule that occur or might occur.

IfcEvent via its IfcEventTime attribute only contains information about the date on which the event occurs and not information about its duration.

3.2 IfcCostItem

IfcCostItem is the IFC entity selected to describe and contain cost information, in a form that allows it to be used within a cost schedule. IfcCostItem is a subclass of IfcControl.

As explained in the section on the IfcTask, a Root Summary Cost is also used here to relate it to the IfcCostSchedule.

Among the attributes of IfcCostItem is IfcCostValue, which defines the total amount of money or the unit cost in case CostQuantities are present.

3.3 Ifc classes applied

The differences between the subclasses of IfcProcess make it possible to manage construction processes with a different degree of detail, depending on the requirements of the design. In the design phase IfcTask can be used, to define the duration of the various tasks, to obtain a time schedule. In the execution phase, the contractor can use IfcProcedure and IfcEvent to define the individual actions to manage the work. In the case study, both scenarios are shown as examples.

The cost model is modeled with the same structure as the process models, so that each individual process node can be controlled.

4 Testing

This chapter shows how to link the various knowledge domains of a case study through the use of the semantic web using IfcOWL as an ontology. A unique RDF graph is produced as output, and the time schedule is represented with a Gantt diagram.

The case study model is extremely simplified and was only developed as a preliminary test. The model consists of a foundation slab, four external walls, a roof, four internal walls, doors and some windows[3]. The model in the version of IFC4_ADD2(buildingSMART 2017).

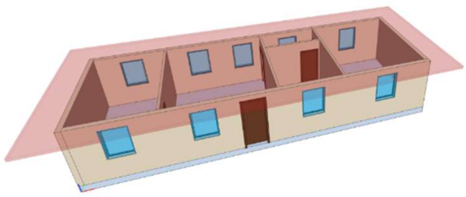

Figure 1. 3D representation of the case study BIM model using the IFC

4.1 Conversion from IFC STEP to RDF graph.

To convert the case study model from IFC STEP to an RDF graph, it is necessary to use a converter. The authors of this article did some export tests with the various tools available in the literature to understand their potential and criticality. The exported files were then opened in Protégé to understand their structure. Pellet as Reasoner was used to check their formal correctness. It is emphasised that the comments made relate to the software versions available at the date of writing this article. In the following paragraphs, the most relevant tools in the literature are commented on.

4.1.1 IFCtoRDF – Desktop

IFCtoRDF - Desktop (Jyrki Oraskari 2020) is a desktop user interface of the IFCtoRDF tool proposed by Pieter Pauwels in command line (Pieter Pauwels 2020). The critical issues encountered are presented below.

[3] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/ResidentialHouse.ifc

The first problem is that by converting the IFC STEP file into the RDF graph, the individuals have attributes and relations written as Annotation Properties, not Property Assertions. This choice allows the graphs to be smaller in size, making it easier to set up queries. However, if Property Assertions are not used, information such as Ranges and Domains, which are essential for the use of Reasoners, are lost.

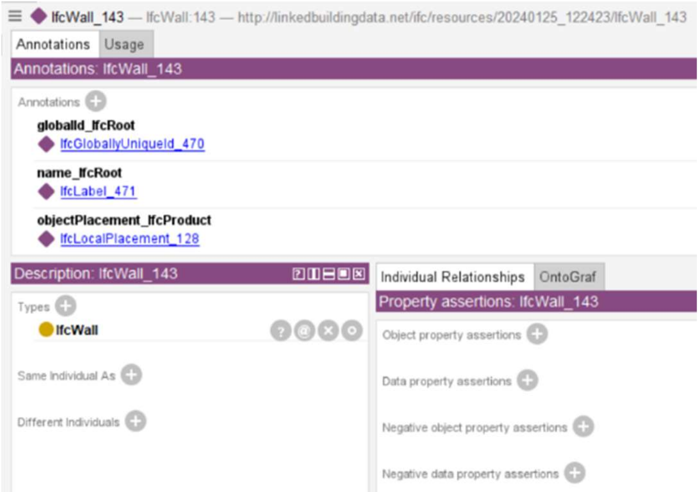

Figure 2. Conversion from IFC STEP to IfcOWL via IFC-to-RDF: properties of the individual IfcWall_14

The second problem is that the classes to which the individuals from the IFC STEP file belong are duplicated without associated properties and descriptions.

4.1.2 IfcSTEP-to-IfcOWL-converters

The IfcSTEP-to-IfcOWL and IfcOWL-to-IfcSTEP converters (Chi Zhang 2021) are based on the IFCtoRDF tool proposed by Pieter Pauwels (Pieter Pauwels 2020) and allow conversion from IFC STEP to RDF and vice versa. This concept guarantees interoperability with all software based on IFC STEP. Indeed, by doing so, it is possible to visualise and modify geometric elements within the modeling environment.

However, like the previous tool, IFCtoRDF converts attributes and properties as Annotation Properties. In addition, there are problems with the import of ontologies, probably due to the lack of maintenance of the tool.

4.1.3 IFC to LBD

IFC to LBD (Jyrki Oraskari 2024) is a specific tool for converting IFC STEP files into an RDF graph based on the BOT ontology. The PRODUCT and PROPS ontologies can also be used if selected in the user interface.

This tool (Oraskari et al. 2018) operates in two steps. Initially, it uses the IFC-to-RDF converter internally to obtain an IfcOWL Abox graph. In the second step it uses the IfcOWL Abox graph as a basis for obtaining an Abox graph on BOT (and possibly PRODUCT and PROPS). In the user interface there is a snap called ‘Create and link IfcOWL’ that allows the IfcOWL Abox[4] graph to be saved. The other export options are not relevant for this article as the graph exported to BOT (and possibly PRODUCT and PROPS) was not considered. In fact, this tool was used to obtain the graph based on IfcOWL. The tool supports from version IFC2x3 to IFC4_ADD2. The most recent version available was therefore used. Therefore, the model used is the IFC4_ADD2 version.

Opening the RDF graph in Protégé and activating the Reasoner notifies an error on the IfcCompoundPlaneAngleMeasure and IfcArcIndex. The cause of the error is related to the Object Property called list:hasNext. The former are individuals related to the geographical coordinates of the building and the latter describe a single circular arc segment. These classes are not of primary interest for this article. To proceed with the experimentation, the Object Property called list:hasNext on these individuals was simply removed. Once removed, the Reasoner detected no further inconsistencies.

[4] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/HouseRDF#

4.2 RDF process and cost model production

The main BIM software does not allow the writing of non-geometrical elements such as IfcTask, IfcProcedure, IfcEvent, IfcCostItem, etc... For the compilation of these classes, there are certain methods, here are some examples.

In collaborative mode, query languages can be used, e.g. SPARQL. This requires writing the query code manually. It is better if this environment is integrated into software that allows the use of Reasoners. This method requires specific programming skills.

Acting in desktop-based logic, it is possible to use applications that allow these classes to be compiled, such as BlenderBIM, a free add-on, open source and Native-IFC AECO toolkit. This method is the quickest, and the software's interface allows easier assignment of tasks. The software does not allow formally incorrect files to be written, and it prevents logical errors by the compiler. It also does not require specific programming skills. Once written to IFC STEP files, it is possible to convert them with the same converter used previously. Alternatively, you can work directly on the IFC file through Python's IfcOpenShell libraries. Or you can write them manually to OWL/RDF ontology compilation software, for example Protégé. This method is not subject to formal errors thanks to Reasoners. However, it remains subject to possible logical errors on the part of the compiling user. To solve this problem, it is necessary to implement rules through languages such as SWRL (Semantic Web Rule Language) or SHACL (Shapes Constraint Language). Given the premises and the remarkable simplicity of these examples, it was deemed appropriate to use this method.

From the same version of IfcOWL used by IFC-to-LBD, Abox graphs containing the time and cost information were made. The values used are for illustrative purposes only and do not reflect a real situation.

As for the time management domain, a graph was produced for each process model. In the beginning, a graph was produced for the Root Summary Task[5], to which the process model of the slab[6], external walls[7], roof[8], internal walls[9], doors[10] and windows[11] are to be related. By joining these process models, a graph with all the tasks necessary for the realization of the case study building is obtained. The same thing was done for the cost models. For simplicity, only the cost model for the Root Summary Cost[12] and for the slab[13] were made. The process and cost models proposed in this article are simple and for illustration only.

One possible application is where entities that publish price lists share RDF graphs describing detailed and structured process models. In addition to sharing a common knowledge base, they allow practitioners to select the desired process models, and compose their own RDF graph to manage projects. Indeed, in the AEC industry, although projects are difficult to repeat, the same cannot be said for process and cost models. For example, with some exceptions, the process model for the construction of a cast-in-place reinforced concrete wall remains standard regardless of the project.

4.3 Importing to GraphDB

Ontotext GraphDB is a free graph database and knowledge discovery tool compatible with RDF and SPARQL. For space issues, all queries formulated in SPARQL are reported and annotated within a text file in a public server[14].

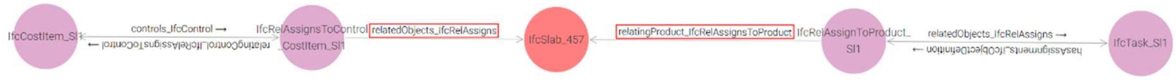

Figure 3. RDF graph after being imported and related into GraphDB

[5] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/ProcessModels/SummaryTaskRoot#
[6] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/ProcessModels/SlabTask#
[7] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/ProcessModels/ExternalWallsTask#
[8] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/ProcessModels/RoofTask#
[9] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/ProcessModels/InternalWallsTask#
[10] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/ProcessModels/DoorTask#
[11] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/ProcessModels/WindowTask#
[12] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/CostModels/SummaryCostRoot#
[13] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/CostModels/SlabCost#
[14] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/SPARQL_Query.txt

4.3.1 Union of process and cost models

The graphs of the process and cost models were imported within GraphDB. Through a query in SPARQL they were joined together, forming an overall graph of the process[15] and cost[16] models.

4.3.2 Relating Building Element and process and cost models

The RDF graph of the case study building is imported. The relationships that go to assign tasks or costs to products through IfcRelAssignsToProduct or IfcRelAssignsToControl relationships have been written into the relevant graphs. Through a query in SPARQL they are related to their product (highlighted with a red rectangle) and through further queries, unnecessary individuals and triples are removed.

Note that the graph shown in Figure 3 (HouseRDF, TaskRDF and CostRDF) describe three different knowledge domains, and reflect a possible real-world situation in which the modeling, time management and economic management software are different. One of the goals of the semantic web is to relate data from different sources. Once these entities are related, a unique RDF graph is obtained[17].

4.3.3 Output

Through a query, a table showing the attributes and relationships of each task is obtained[18].

Table 1. Extract of the table obtained by a query in SPARQL showing attributes and relationships of the IfcTask

Task	Name	Predecessor	Child	Schedule Start	Schedule Finish
inst_witsk: IfcTask_SWi1	Summary Task of all windows' tasks	inst_dotsk: IfcTask_SDo1	inst_witsk: IfcTask_Wi1	13/8/24 8.00	13/8/24 16.00
inst_dotsk: IfcTask_SDo1	Summary Task of all doors' tasks	inst_waitsk: IfcTask_SWai1	inst_dotsk: IfcTask_Do1	12/8/24 8.00	12/8/24 16.00
...	...	...	...	...	...

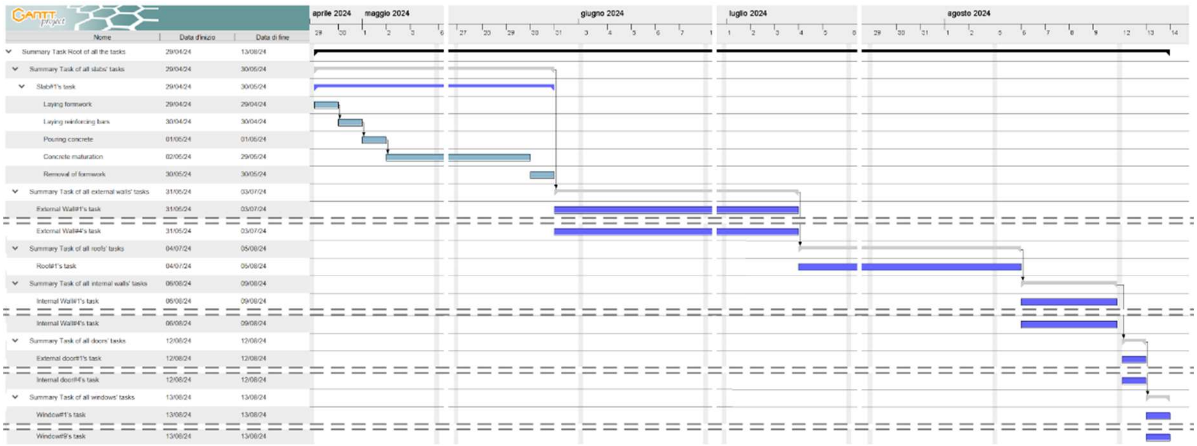

Figure 4. Gantt diagram of the case study using Gantt Project

The resulting table was later reworked so that it could be imported into GanttProject, a free and open-source Gantt charting software.

5 Analysis of results

Conversion to RDF graphs according to the IfcOWL ontology results in a significant increase in the size of the source file. This fact does not necessarily imply an increase in response time by software working with RDF graphs compared to those working with IFC STEP. In fact, since they are two different types of formats, they also have different read and write speeds. Further research is needed to understand which of these two formats performs better.

[15] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/TaskRDF#

[16] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/CostRDF#

[17] https://lie.dicea.unipd.it/Publications/2024/CIB_W78/TempSched/Tab for GanttProject.csv

An analysis was conducted to count the number of individuals in each Ifc class, through the implementation of a query on SPARQL. This query was applied to each model in the source dataset to understand which classes had the most individuals and in what percentage. Next, the results obtained for each model were averaged. Note how more than 85% of the individuals pertain to IFC classes for the geometric definition of the elements. Usually, in the context of linked data this information is not considered since 3D visualization is not a priority. Therefore, the workflow is slowed down by the presence of information that is not used. It is necessary to continue the research to arrive at a modularization of the ontology. In this way, it is possible to work only on the data of interest by creating modules divided by discipline and then rejoining them with the overall graph.

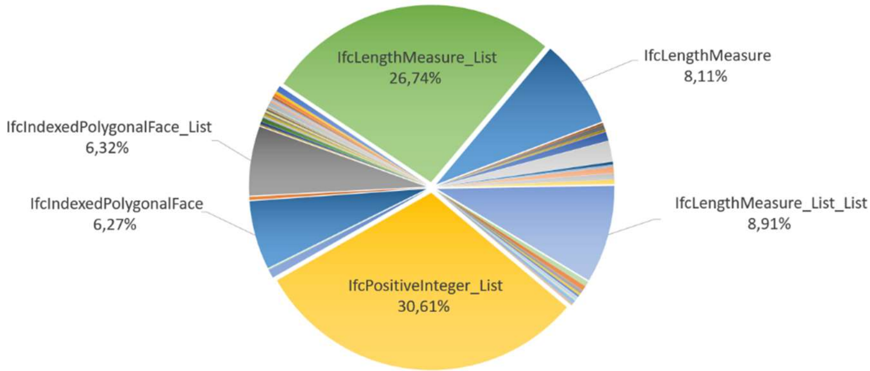

Figure 5. Percentage of the average number of individuals within RDF graphs of real case models

6 Conclusions and future works

One of the biggest issues in the AEC industry is the lack of interoperability and structuring of information. This leads to procedures that share information in a heterogeneous, unstructured way and without standard terminology. To overcome this problem in 1994, the IFC initiative was born to produce a shared standard for the entire infrastructure industry.

The first contribution of this paper is to discuss and define the role of BPMN in the temporal definition of projects. This standard allows for a clear description of the logical sequence of activities, but it does not allow one to go in and define their timing. The IFC standard must be used to compile this information.

The second contribution of this article is to discuss the benefits of the semantic web. Through the semantic web, information can be shared, standardized, and managed collaboratively. The main goal of the semantic web is to relate data from different sources using a structured, common language. As a result, cloud collaboration based on linked data models (LDM) is becoming a reference to support collaboration among different stakeholders. In collaborative mode, query languages can be used (i.e. SPARQL), so the dataset of information can be queried to extract the information needed.

The third contribution of this paper is to discuss the web ontology for the AEC industry. There are various ontologies in the literature to simplify the structure of IFC and improve its performance. Doing so, however, fails the premise of standardization and the use of a common structured language. The proposal of new ontologies expresses the need to work with more user-friendly and lighter data structures. There is a need to develop software, apps, and websites that overcome the complexity of the standard through a simple user experience that helps the user. This does not renounce the completeness of the information exhibited by the IFC scheme. From the performance point of view, it is also necessary to continue the research for modular use of IfcOWL. By doing so, it is possible to work with reduced amounts of data without losing information. As shown in Figure 5, about 85% of the individuals in an RDF graph belong to classes related to the geometric definition of elements. This data is usually not used in the linked data environment, but it is necessary to keep it in case someone wants to return to the modeling environment. As future work, it is intended to develop a tool that allows conversion from IfcOWL-based RDF graph to an IFC STEP file. In addition, developers are encouraged to continue to maintain the conversion tools (IFC STEP to IfcOWL) to use the most up-to-date versions of IFC.

The fourth contribution of this paper is a simple case study demonstrating the application of RDF Abox graphs based on process and cost models. This approach facilitates information reuse, significantly saving time and enhancing knowledge sharing about building processes.

The experiment is based entirely on the IFC data model and can also be reproduced using desktop-based tools based on the IFC STEP format. However, although even the desktop-based approach ensures data standardization and interoperability, it does not allow for web-based collaborative work. The workflow shown in this article is too cumbersome for real practical application. Further future work is the development of software and/or web apps that simplify the workflow.

References

buildingSMART. (2017). *IFC4 ADD2* [online]. Available from: https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/ [accessed 17 May 2024].

buildingSMART. (2024). *IFC4X3_ADD2* [online]. Available from: https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/content/scope.htm [accessed 14 May 2024].

Chi Zhang. (2021). *IfcSTEP-to-IfcOWL-converters* [online]. Available from: https://github.com/BenzclyZhang/IfcSTEP-to-IfcOWL-converters [accessed 17 May 2024].

Niknam, M. and Karshenas, S. (2017). *A Shared Ontology Approach to Semantic Representation of BIM A Shared Ontology Approach to Semantic Representation of BIM Data Data Mehrdad Niknam*. Available from: https://epublications.marquette.edu/civengin_fac.

Ismail, A. (2022). *BIM integrated and reference process-based simulation method for construction project planning*.

Jyrki Oraskari. (2024). *IFCtoLBD* [online]. Available from: https://github.com/jyrkioraskari/IFCtoLBD [accessed 17 May 2024].

Jyrki Oraskari. (2020). *IFCtoRDF-Desktop* [online]. Available from: https://github.com/jyrkioraskari/IFCtoRDF-Desktop [accessed 17 May 2024].

Mendes De Farias, T. (2015). *IfcWoD,Semantically Adapting IFC Model Relations into OWL Properties*.

Messi, L., Spegni, F., Carbonari, A., Ridolfi, L. and Vaccarini, M. (2021). *Process-based simulation models using BPMN for construction management at runtime*.

Oraskari, M., Pauwels, J., Vergauwen, P. and Klein, M. (2018). *The IFC to linked building data converter*.

Pan, J., Anumba, C.J. and Ren, Z. (2004). *Potential application of the semantic web in construction*.

Pauwels, P. and Terkaj, W. (2016). EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*, 63, pp.100–133.

Pauwels Pieter and Roxin A. (2016). *SimpleBIM : from full ifcOWL graphs to simplified building graphs Citation for*. Available from: www.tue.nl/taverne.

Pieter Pauwels. (2020). *IFCtoRDF* [online]. Available from: https://github.com/pipauwel/IFCtoRDF [accessed 17 May 2024].

Rasmussen, M.H., Lefrançois, M., Schneider, G.F., Pauwels, P. and Janowicz, K. (2020). *BOT: the Building Topology Ontology of the W3C Linked Building Data Group*. Available from: https://www.microsoft.com/microsoft-365/.

Rasmussen, M.H., Lefrançois, M., Pauwels, P., Hviid, C.A. and Karlshøj, J. (2019). Managing interrelated project information in AEC Knowledge Graphs. *Automation in Construction*, 108.

Sheik, N.A., Veelaert, P. and Deruyter, G. (2023). Exchanging Progress Information Using IFC-Based BIM for Automated Progress Monitoring. *Buildings*, 13(9).

Terkaj, W. and Pauwels, P. (2017). *A Method to generate a Modular ifcOWL Ontology*.

Wix, J. (2007). *Quick Guide Business Process Modeling Notation (BPMN) Quick Guide*.

Xue, W., Wang, Y. and Man, Q. (2015). Research on information models for the construction schedule management based on the IFC standard. *Journal of Industrial Engineering and Management*, 8(3), pp.615–635.

Yang, B., Dong, M., Wang, C., Liu, B., Wang, Z. and Zhang, B. (2021). Ifc-based 4d construction management information model of prefabricated buildings and its application in graph database. *Applied Sciences (Switzerland)*, 11(16).

Zeeshan, A., Assistant, R., Chimay, A., Darshan, R., Associate, R., Patricia, C. and Lecturer, S. (2004). *Semantic Web Based Services For Intelligent Mobile Construction Collaboration*. Available from: http://www.lboro.ac.uk/cice/.