www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

ENHANCING ACCURACY IN COST ESTIMATION: STRUCTURED COST DATA INTEGRATION AND MODEL VALIDATION

SUBMITTED: April 2024
REVISED: December 2024
PUBLISHED: December 2024
GUEST EDITORS: Vito Getuli, Farzad Rahimian, Nashwan Dawood, Pietro Capone, Alessandro Bruttini
DOI: 10.36680/j.itcon.2024.058

Jacopo Cassandro, PhD Candidate,
Polytechnic of Milan, Italy;
jacopo.cassandro@polimi.it

Claudio Mirarchi, Researcher,
Polytechnic of Milan, Italy;
claudio.mirarchi@polimi.it

Carlo Zanchetta, Researcher,
University of Padua, Italy;
carlo.zanchetta@unipd.it

Alberto Pavan, Associate Professor,
Polytechnic of Milan, Italy;
alberto.pavan@polimi.it

SUMMARY: *Cost estimation for tendering is one of the leading causes of legal disputes in the architecture, engineering, construction, and facilities management (AEC/FM) industry. To date, there are no standardized procedures for verifying cost estimation accuracy. The association between cost data and geometric model objects is currently done manually, leading to time losses and inaccuracies in cost estimates. This study defines a new architecture for cost items within the Industry Foundation Classes (IFC) data model to ensure structured cost data, which is currently expressed in unstructured natural language. It encompasses the definition of a cost domain in IFC, potential relationships it may have with an IFC geometric information model, and the validation of these relationships through semi-automated verification procedures of structured data (geometric-cost). A specific case study, focused on a structural IFC model, is examined to assess current and future applications. Additionally, rules for BIM information requirements will be defined through the Information Delivery Specification (IDS) to ensure clear understanding for both humans and computers. This will specify which data must be included in the geometric model to ensure validation and verification of the uniqueness of associated cost data. The results demonstrate the feasibility of defining structured cost elements in IFC and verifying their association with geometric data to ensure coherence and uniqueness in cost estimation.*

KEYWORDS: *BIM, IFC, cost ontology, compliance checking, cost item, Information Delivery Specification (IDS), AEC/FM.*

REFERENCE: *Jacopo Cassandro, Claudio Mirarchi, Carlo Zanchetta & Alberto Pavan (2024). Enhancing accuracy in cost estimation: structured cost data integration and model validation. Journal of Information Technology in Construction (ITcon), Special issue: 'Managing the digital transformation of construction industry (CONVR 2023)', Vol. 29, pg. 1293-1325, DOI: 10.36680/j.itcon.2024.058*

1. INTRODUCTION

Accurate cost estimation is critical for effective planning and budgeting of construction projects in the architecture, engineering, construction, and facilities management (AEC/FM) industry. It serves as a key role in project management, guiding decisions across all stages of construction.

However, this process is often characterized by human errors, discrepancies in pricing and association with the various components of the project, and inaccurate quantities calculation. These issues undermine the reliability of cost estimates, compounded by the absence of standardized procedures, which further exacerbates the situation (Adeli et al., 2001; Lu et al., 2016).

In recent years, the integration of Building Information Modeling (BIM) has presented promising opportunities to enhance cost estimation methodologies. Despite its widespread adoption, challenges persist in maintaining the accuracy and consistency of cost associations with project elements. While previous research, such as by Wu et al. (2014), Sacks et al. (2018), Elghaish et al., (2020), and Olatunji et al. (2021), has illuminated BIM's potential to revolutionize cost estimation, fundamental approach to computation remains largely the same.

Currently, the 5D BIM cost management software retrieves information from models exported in the open format Industry Foundation Classes (IFC), containing unstructured cost data to be linked with the definition of the cost estimate.

The approach to cost estimation for public tenders differs across various regions or states. In Italy, the method involves utilizing a price list, different for each region, which is a list of cost items that serve as the foundation for economic offers and payment regulations in public contracts. In the specific case of this research the cost items are collected within the price list of public works of the Lombardy Region, a document that is based on unstructured data and cost items described in natural language.

This article addresses the persistent gap between BIM's expansive capabilities and the assurance of precise cost relations, underlining the urgent need for innovative solutions to address this critical issue.

In the AEC/FM industry, data exchange relies on the standard IFC, standardized by ISO 16739-1 (ISO 16739-1:2024, 2024), which establishes a robust framework for information exchange between various systems (Froese T et al., 1999). The IFC scheme is a rich, extensive, and complex schema. The complexity of the IFC data model is due to its need to represent objects and their interrelations through various AEC subdomains, which require a complete and connected representation (Eastman et al., 2010). BuildingSMART International's publication of the IFC standard plays a crucial role in facilitating BIM data exchange among stakeholders, given its open specification. Within the IFC data model, several entities represent building management information, including *IfcConstructionManagementResource*, *IfcWorkPlan*, *IfcTask*, *IfcScheduleTimeControl*, *IfcCostSchedule*, *IfcCostItem*, and *IfcCostValue*.

This study seeks to lay the groundwork for developing a new structured cost domain within openBIM through a standardized structure of cost items and implementing a verification procedure to ensure correspondence between unit cost items and objects in information models. By considering cost items as a structured class in the IFC data model (*IfcCostItem*), relationships between geometric objects and cost items can be established. This structured approach enables validation of the correspondence between cost items and geometric objects, ensuring the accuracy and uniqueness of established cost-object relationships and subsequent cost estimation. To this end, to validate information accuracy, an Information Delivery Specification (IDS) file was created initially to define geometric model requirements. This has been fundamental to guarantee the correct relation of cost entities and analyzing data in subsequent phases.

The paper is developed as follows. First, in the section "2. BACKGROUND", an analysis of the existing literature on construction management using IFC and the current techniques of model checking is presented. After that, in the section "4. COST DATA ARCHITECTURE", the key aspects underlying the research and what emerged from their detailed analysis are reported; among these, there is the study of the class *IfcCostItem* and *IfcConstructionResource*, the study of IDS standard, the analysis of the current cost items in the price lists and the study of the new structure hypothesized for the cost items in a project that the research group is developing with Lombardy Region.

Subsequently, the section "5. EXPERIMENTATION AND RESULTS" shows the experimental part with the definition of the architecture of the unit cost items in IFC and the development of a BIM model characterized by geometric and cost related entities. Currently, no BIM software allow to write these new cost entities and relate them to geometric elements, so the IfcOpenShell library has been used to develop this section. The result of the code has been validated through the IFC viewer and IFC reader.

The last step of this research was the validation of the data model generated through the definition of the information requirements that the geometric model must contain through the development of the IDS. This allows the correct execution of the next phase of verification and validation of the relationships between geometric entities and cost entities defined in precedence. The paper concludes with a discussion and conclusion of the research in the sections “6. DISCUSSION” and “7. CONCLUSION”.

2. BACKGROUND

2.1 Model Checking and Existing Applications

One of the most relevant aspects, which characterizes the BIM process, is the validation of the model and the information contained in it to ensure an adequate exchange of data. There are currently various forms of project validation or model control, which include verification against geometry (clash detection), validation against design parameters (e.g., specifications within the BEP, BIM validation), and compliance checks with regulations (code checking).

The main objective of these checks is to verify the accuracy and consistency of all the information in the model, ensuring compliance with the established rules.. However, despite the facilities offered by software and tools, the validation process remains time-consuming, expensive, and prone to errors (Dimyadi & Amor, 2013).

Model checking is crucial in information modeling and management (Ciribini et al., 2015) for data validation (Ghannad et al., 2019). Research indicates that in standard design processes, only 5-10% of the project's information content undergoes systematic verification (Trebbi et al., 2020).

Nowadays the verification of the models takes place through clash detection that involves verifying geometric interference, which could pose issues during construction if not addressed preemptively within the 3D model (Akponeware & Adamu, 2017). These interferences are typically categorized as "hard clashes", where two objects physically collide and occupy the same space, and "soft clashes", where objects are close but do not collide.

Code checking entails ensuring that the digital model complies with relevant regulations (Trebbi et al., 2020). By leveraging specialized software for these controls you can improve various aspects of building design by reducing time and error and increasing efficiency and model quality (Greenwood et al., 2010). It is essential to verify the compliance of the model with regulatory and technical requirements, making automatic controls of the frequency of information and uniqueness highly valuable in the AEC sector (Solihin & Eastman, 2015). Additionally, compliance with "Employer Information Requirements" (E.I.R.) or "BIM Execution Plan" (B.E.P.) may need to be confirmed.

Early initiatives in automated code compliance checking include the Singapore project CORENET (COnstruction and Real Estate NETwork) (*CORENET*, 2024), which focused on integrating all lifecycle phases. Similarly, in the USA, SMARTCodes emerged (*Codes That Support Smart Growth Development | Smart Growth | US EPA*, 2024), and Autodesk Revit introduced plugins like UpCodesAI, supporting parts of the International Building Code and other standards from various jurisdictions (*UpCodes | Searchable Platform for Building Codes, Assemblies, and Building Products*, 2024).

Model checking in the AEC industry is commonly facilitated by standalone applications like Solibri Model Checker, SMARTcodes, ePlanCheck, AEC3 Compliance, or EDM Model Server (Ismail et al., 2023). Clash detection is a typical application of model checking, used to verify if various elements, such as different types of pipes, intersect within the model. Another example is ensuring that door widths comply with accessibility codes outlined in regulations or national standards. Among the most utilized tools for model data verification are Solibri Model Checker and Navisworks.

Solibri Model Checker (SMC) stands out as a leading BIM software application, enabling designers to visualize and address issues within the design model both before and during construction. It offers a comprehensive suite of functionalities, leaving users with minimal room for manual intervention.

While SMC comes pre-loaded with rules based on the Norwegian State Administrative Agency handbook, users have the flexibility to customize these rulesets to suit specific project requirements. However, creating entirely new rules within SMC is somewhat restricted, as it requires access to the API, which is not publicly available.

Navisworks is another widely adopted tool for clash detection and model coordination across various disciplines.

This software specializes in detecting intersections or conflicts between elements within the 3D model, facilitating the swift identification and resolution of construction or design issues. By identifying clashes early in the project lifecycle, Navisworks contributes to reducing errors and costs during project execution.

2.2 IFC & Construction Management

The Industry Foundation Classes (IFC), developed by BuildingSMART, stand as an open and interoperable standard aimed at enhancing interoperability across diverse domains within construction engineering projects. The IFC ensures "interoperability between industrial processes of all different professional sectors in civil engineering projects by allowing IT applications used by all project participants to share and exchange information about the project" (BuildingSMART, 2024).

Since its inception in 1997, IFC has been instrumental in facilitating the sharing and exchange of project information among various computer applications utilized by different stakeholders. The official version until the end of 2023 was IFC4 ADD2 TC1, which achieved ISO standard status in 2018 (ISO 16739-1:2018). In the early months of 2024, the latest version, IFC version 4.3 ADD2 - 4.3.2.0, received ISO approval and was subsequently published in April 2024 (ISO 16739-1, 2024), laying the groundwork for IFC software certification.

The data model is designed to be vendor-independent and can be used in a variety of hardware devices, software platforms, and interfaces for different use cases. In the AEC sector, data is exchanged through the IFC format. This format ensures compatibility between different software and allows for the exchange of information among project members. It also enables storing the information as part of process libraries (Jiang et al., 2019)(Fürstenberg, et al., 2021). Structured in EXPRESS data specification language or XML, the IFC standard delineates actors, processes, controls, resources, and products within the construction domain. Its hierarchical organization comprises four conceptual layers: Resource layer, Core layer, Interoperability layer, and Domain layer. Entities within this model can be interconnected and characterized by a set of attributes, facilitated by the *IfcRelationship* entity. This flexible structure allows for related information to be stored either within or outside the project data.

In the field of construction management, IFC comprises entities such as *IfcWorkPlan* (for schedule planning), *IfcTask* (representing construction tasks), *IfcScheduleTimeControl* (providing task time information), *IfcCostSchedule* (about cost planning), *IfcCostItem* (defining unit items for cost estimation), and *IfcResource* (including construction resources like material, product, labor, and equipment resources).

IFC plays a crucial role in the exchange of information throughout the life cycle of construction projects and in building management. Is a vendor-neutral data scheme in the AEC industry and enables the continuous exchange of geometric and semantic information between stakeholders and software solutions (Fürstenberg, et al., 2021).

Numerous researchers have studied the potential of the IFC data model in the field of construction information management. Froese T et al. (1999) conducted an extensive analysis of project planning and cost estimation data.

Their findings underlined the versatility of the IFC standard in effectively representing various aspects of construction projects, including costs, construction processes, resources, products, and project documentation. On this basis. Fu et al. (2006) introduced the concept of the nD model, utilizing IFC to outline information relevant to planning and cost estimation. Similarly, Ma et al. 2010) proposed a framework for BIM-based Construction Cost Estimating (CCE) software, grounded in Chinese standards, to simplify cost estimation processes. Zhiliang et al. (2011) have deepened the integration of the IFC standard into construction cost estimation practices for tenders in China.

Their study highlighted the need for extensions, such as proxy elements and property sets, to fully exploit IFC's potential in accurately expressing construction cost estimation data. In addition, Ma et al. (2013) addressed key issues related to cost estimation of semi-automated and specification-compliant Tendering of Building Projects (TBP), leveraging IFC data from design models into architectural and structural engineering projects.

In a bid to enhance collaboration and information sharing efficiency, Liao et al., (2014) proposed a methodological approach to develop a collaborative construction prototype model on BIM software. Meanwhile, Xu et al., (2013) have developed an innovative method to extract data from the BIM model and use it to make the cost estimation for the project, leveraging the IFC standard. Furthermore, recent studies by Wu et al. (2014), Sacks et al. (2018), Elghaish et al. (2020), Olatunji et al. (2021), and Fürstenberg, et al. (2021) have further explored the potential of BIM to augment cost estimation practices.

3. SCOPE AND METHODOLOGY

Within the scope of the research, several key challenges related to defining and validating the relationships between costs and model objects are identified. First, it is essential to establish a cost ontology and a subsequent standardized procedure for the verification and validation of costs related to the model objects. Accurate structuring of cost data within the classes defined in the IFC data model is necessary to define a new open BIM cost domain, overcoming the approach based exclusively on the association of attributes.

One of the main challenges is ensuring that the architectures of cost items are more complex and structured than a textual representation in natural language. Overcoming this limitation is crucial to enable better readability and understanding of cost data by computer systems. Furthermore, the research deals with the need to validate data across domains, such as the geometric and cost domains, present within the same model.

The research focuses on the key aspects of:

- What are the key factors for structuring cost data?
- How can we establish a process for verifying and confirming data between the geometric and cost domains?

For this reason, the possibility of structuring project economic management information directly within the IFC data model is being investigated.

The research focuses on managing cost data within a new cost database structured and implemented in an openBIM format following the IFC data model to ensure data validation across different domains.

Currently, in Italy, in the specific case within the Lombardy Region's price list of public works, cost items are characterized by unstructured textual description, leading to potential challenges and errors in associating costs.

To address this, the research focuses on structuring the cost items into the IFC data model and relating them with their respective geometric objects of a specific IFC case study. Additionally, rules for BIM information requirements are established through the IDS to ensure clarity for both humans and computers. This facilitates defining the necessary data in the geometric model to ensure accurate cost relation and subsequent validation and verification of cost data uniqueness.

This section describes the framework and methodology of the proposed system. The proposed methodology assumes that the BIM model is exported in IFC format. The IFC format was chosen because it is an open and interoperable data format widely recognized as a comprehensive and widely used data model in the AEC/FM sector. This ensures that our model can be used across various software platforms universally.

Figure 1 shows the logical process used in this research. The logical process is divided into three steps: (1) Data Creation, (2) Data Integration, and (3) Data Validation.

Step (1) is to create a geometrical BIM model, export it in IFC format, and then verify it according to the requirements established in the IDS. Once step (1) is finished, the IFC model is analyzed through the IfcOpenShell library to allow the extraction of the *IfcElement* entities to be related to the cost items and then integrated with the new *IfcCostItem* entities.

The cost items will be selected from an external cost database, developed previously and which will contain cost items structured according to a specific architecture in IFC. A new cost domain consisting of interoperable and openBIM cost classes within IFC has therefore been defined.

At the moment the cost database contains only entities useful for the research (cost item of concrete masonry, formwork, reinforcing bar, etc.) but in the future, it may contain all the cost items necessary for a cost estimation. In example, it may contain all the cost items currently stored in the price lists of Lombardy Region or the cost items developed by a practitioners.

These entities were developed by code and their implementations is possible through the creation of new *IfcCostItem* entities and subsequent storage in the cost database. After step (2) a new BIM model is saved in IFC format containing both geometric and cost data. At this point, the logical process foresees the validation of the relations defined in step (2). The purpose of step (3) is to identify any conflicts or errors and report them to resolve them, ensuring a detailed and useful BIM model for the definition of an accurate cost estimate.

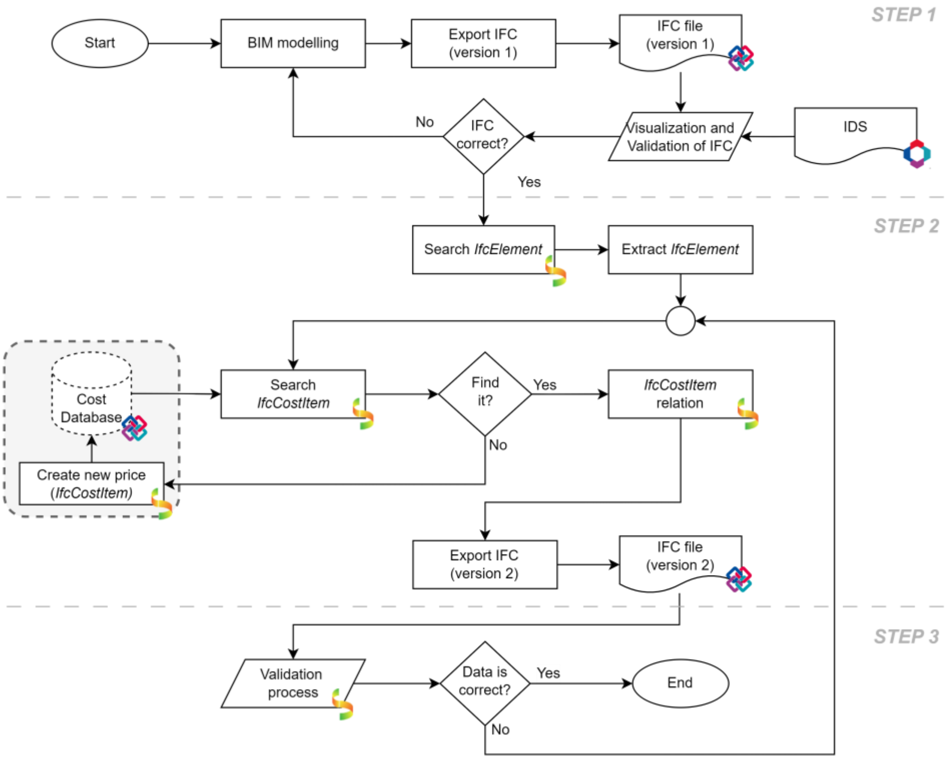


Figure 1: Research methodology framework

4. COST DATA ARCHITECTURE

Within IFC some entities allow to represent the information of the construction management. Among these are: *IfcWorkPlan* which allows program planning, *IfcTask* that defines the construction task, *IfcScheduleTimeControl* that identifies the time information of the task, *IfcCostSchedule* that collects cost planning, *IfcCostItem* that allows you to describe the unit item to be used in cost estimation, and *IfcResource* that includes construction resources (material resources, product resources, labor, and equipment).

4.1 IfcCostItem

IfcCostItem is the entity that represents the cost of assets and services, the execution of works by a process, lifecycle cost, cost estimates, budgets, and more in the IFC standard. It is a non-geometric entity, a subclass of *IfcControl*. The abstract entity *IfcCostItem* describes a cost or financial value with descriptive information that describes its context in the form that enables it to be used within a cost schedule (BuildingSMART, 2024).

IfcCostItem is also described through a set of attributes. Some of them are inherited from the hierarchically superior entities, such as *Name* and *Description,* instead, the attributes *PredefinedType*, *CostQuantites* and *CostValue* are the proprietary attributes of the class (Figure 2).

The Name attribute serves to offer a shared value facilitating the grouping of separate instances within a grouping arrangement, whereas the Description attribute holds the text utilized for describing the element in a cost schedule. (BuildingSMART, 2024).

An *IfcCostItem* can link one or many *IfcCostValue* instances, which can represent a unit cost, total cost, or a unit cost along with one or multiple quantities utilized for generating the total cost. An *IfcCostValue* may represent an original value, or a value derived from formulas. Each instance of *IfcCostValue* can also have a category. Many possible types of cost value can be identified such as “labor”, “material”, “equipment”, “overhead”, “profit”, “estimated cost” and so on. These *IfcCostValue* are stored in the *CostValue* attribute.

The quantities can be given as individual quantities, or those quantities are provided as element quantities by one or many building elements. *IfcPhysicalQuantity* entity represents the physical quantity of a certain item used in construction projects, which can be used to calculate the costs associated with the materials, labor, equipment, and construction work needed to complete the project. The *IfcMeasureWithUnit* entity can be used to represent a quantity and unit of measure, which can be used in cost calculations. These *IfcPhysicalQuantity* are stored in the *CostQuantities* attribute.

Another crucial aspect is that an *IfcCostItem* can define various relationships. It can be nested to form cost assemblies through the *IfcRelNests* relation. Additionally, it can be linked to an *IfcProduct* via the *IfcRelAssignsToControl* relation. Moreover, it may be associated with a product through the *IfcRelAssignsToProduct* relation or with a resource through the *IfcRelAssignsToResource* relation.

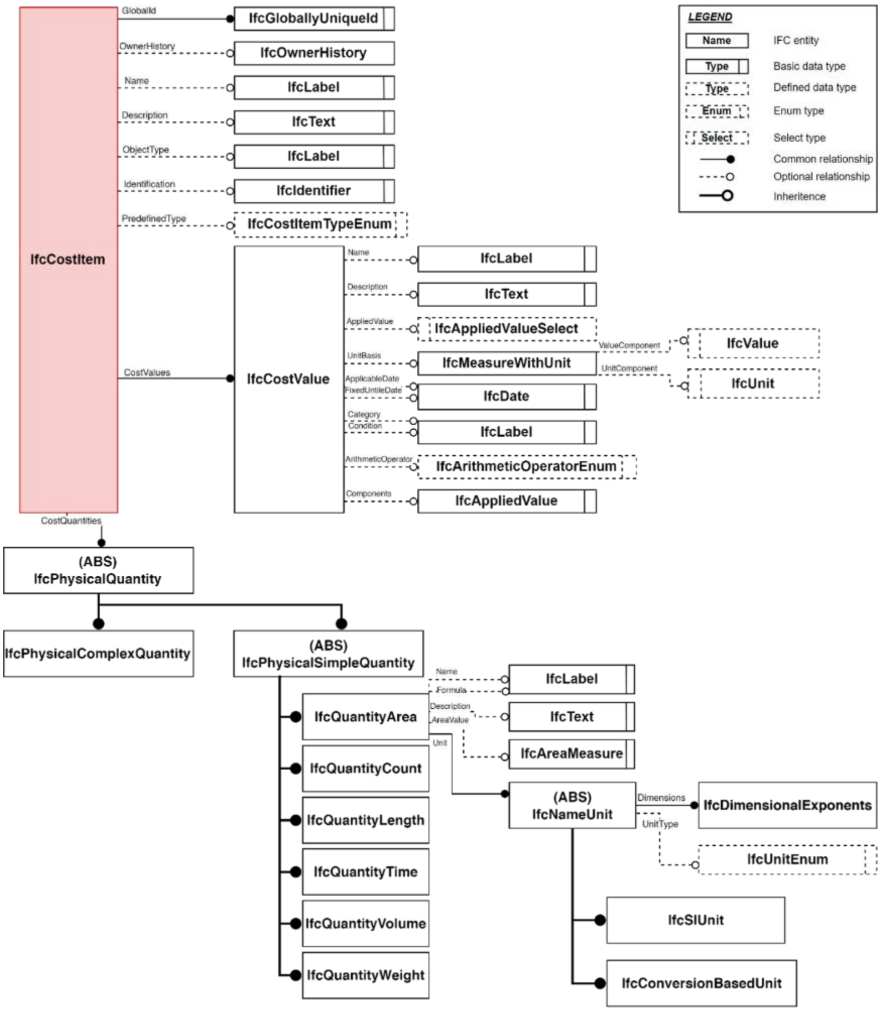


Figure 2: Direct attributes of the IfcCostItem entity

4.2 IfcConstructionResource

IfcConstructionResource entity is a subclass of *IfcResource* and itis used to represent the resources required for construction activities (materials, labor, and equipment) and each resource can be further classified based on various attributes such as cost, availability, and performance (BuildingSMART, 2024). Inside *IfcConstructionResource* there are *IfcConstructionMateriaResource*, *IfcConstructionProductResource*, *IfcEquipmenteResource*, *IfcCrew*, *IfcLaborResource* (Figure 3). All these entities can be used to create a comprehensive data structure for construction cost estimation and storage of price items in a price list thus creating a link between the theorized cost ontology and IFC cost domain.

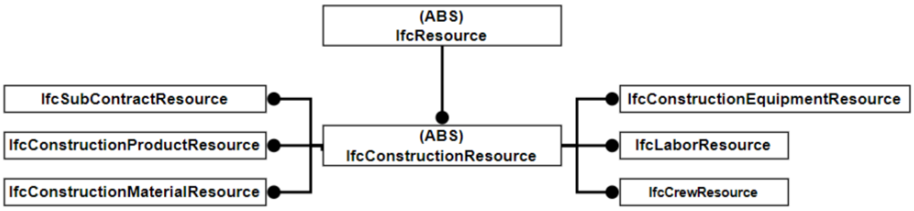


Figure 3: IfcResource and relative sub-entity

4.3 Information Delivery Specification

The Information Delivery Specification (IDS) serves as a standardized framework developed by BuildingSMART to delineate the requisite level of information essential for a specific project (BuildingSMART, 2023). IDS establishes the information criteria necessary for a geometric model to facilitate seamless data exchange, ensuring readability by humans and interpretability by machines. It outlines the methodology for delivering and exchanging various components, properties, values, and units of measurement within a project. An IDS file may encompass multiple independent requirements, devoid of interdependencies, enabling the creation of reusable blocks adaptable across different projects. It is the standard used to define the Level of Information Need (LoIN) (*EN 17412-1*, 2020) and the Exchange Information Requirements (EIR) (*ISO 19650-1*, 2018).

Traditionally, information requirements have been disseminated through Excel spreadsheets or PDF documents, which lack machine interpretability and pose readability challenges due to their voluminous nature. IDS, on the other hand, focuses on specifying information delivery requirements in a structured manner, ensuring the accurate transmission of the necessary information in projects, and improving automated workflows by providing machine-readable data. Moreover, the formulation of an IDS plays a pivotal role in standardizing diverse modeling approaches prevalent in the industry. For instance, it helps reconcile disparities in modeling techniques, such as the usage of slabs versus landings, fostering consistency and interoperability across projects.

Therefore, this document has a crucial role, since it offers a validation of the IFC for both the client and the modeler. It also serves as a contractual basis for ensuring the correct provision of information, and can be considered a "legal document". The IDS can create specific requirements for each use case, adapting to an organization's projects.

The process of compiling and validating the IDS (Figure 4) involves (1) the creation of the IDS by the client, (2) the sending of the IDS to the modeller, (3) the creation of the model, (4) the validation of the data by the modeller against the IDS, (5) sending the IFC model to the client, (6) validation of the data by the client against the IDS.

Each IDS consists of three main components:

- *Description*: This section provides an overview of the rationale behind the specification's significance within the project context, along with instructions on its implementation. Its purpose is to enable users to comprehend the necessity for the requested information effectively.
- Applicability: The specification outlines the category of IFC model objects to which it applies. This delineation ensures clarity regarding the scope of application within the project.
- Requirements: This part defines the information needed for the objects specified in the previous section. It details the properties or classifications required, facilitating precise data acquisition.

The applicability and requirements are described using facets, which serve to precisely describe the information applicable to a single entity. When included in the requirements section, a facet defines the necessary information for an entity to comply with the specification using fixed parameters, ensuring clarity and interpretability by computer systems.

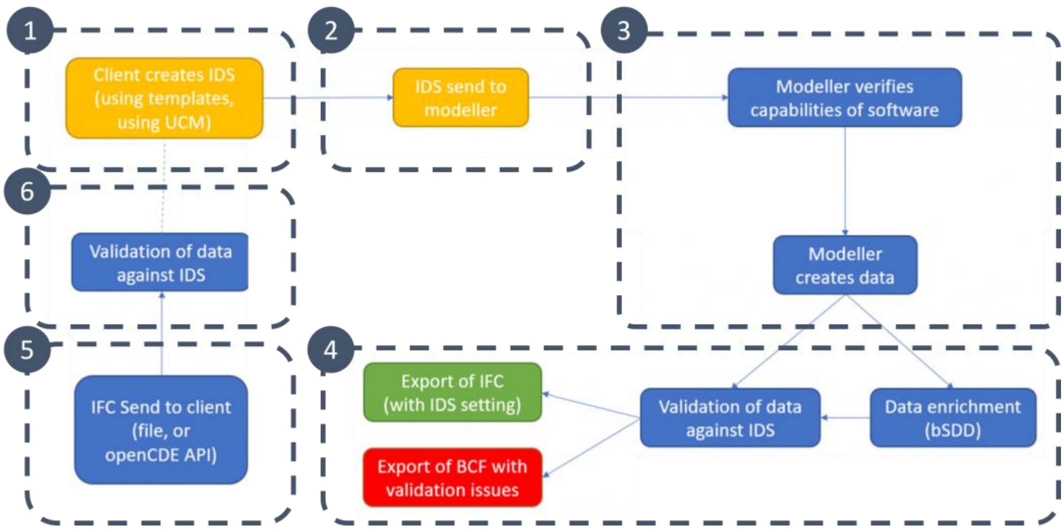


Figure 4: Workflow of IDS

4.4 Price list

The price list is a catalogue of all cast items that can be used in the public tendering. In the specific case of this research, the cost items used are collected within the price list of the Lombardy Region.

These cost items, Figure 5, typically include:

- A unique code for each item of construction works and resources [1];
- A description of the item in natural language [2];
- The unit of measurement for the construction works and resources [3];
- The unitary price for the construction works and resources [4];
- The impact of labor, equipment, and materials on the unit price [5].

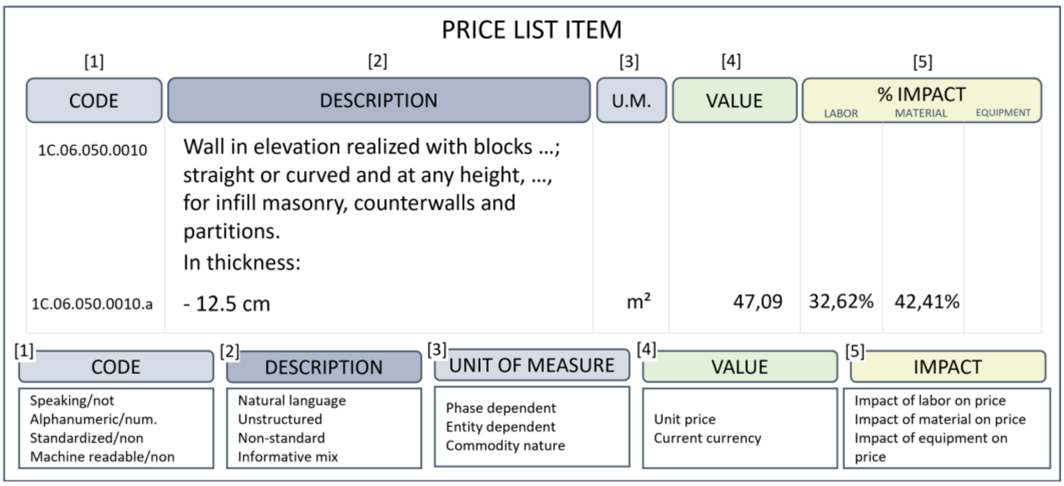


Figure 5: General structure of the current Price List in Italy

Although these protocols are specific to Italy, the general methodology can be adapted to any cost estimation approach. The fundamental factors influencing construction costs - such as labor, equipment, materials, overheads, and profits - remain consistent across different methodologies. These factors collectively contribute to the total project amount.

4.5 Cost Items Breakdown

Currently, cost items are characterized by descriptions in natural language withunstructured data and without a specific meaning, resulting in unclear and unrelated to each other. This makes the data unclear and not readable by a digital tool and on the other hand leaves free and subjective interpretation of the data by users (professionals, architects, quantity surveyors, etc.). This causes errors and time wasted in understanding, choosing, and evaluating cost items to be associated with geometric objects. An example of the current cost item description is visible in Figure 6. It is immediately visible that the cost item information is not ordered but placed in different positions of the text; which can generate interpretations and errors of assessment.

Analysis of the cost item shows that:

- The information related to the work to be realized is found both at the beginning of the description ("masonry in elevation") and the end ("for masonry infill, counterwalls and partitions: thickness 12.5 cm"); in Figure 6 are highlighted in yellow.
- The information related to the activity to be performed ("realized") and the modes of realization of the work ("straight or curved and at any height", "with horizontal holes") are scattered at the beginning, at the center and the end of the description; in Figure 6 are highlighted in purple.
- The information related to the resources that characterize the work can be found both at the beginning of the Description ("with blocks of alveolar bricks according to UNI EN 771") and at the end ("with blocks 25 x 25 cm", "a percentage of perforation of 60/70%"); in Figure 6 are highlighted in green.
- The information related to the included components is inserted among all that has been reported above; in Figure 6 are highlighted in blue.

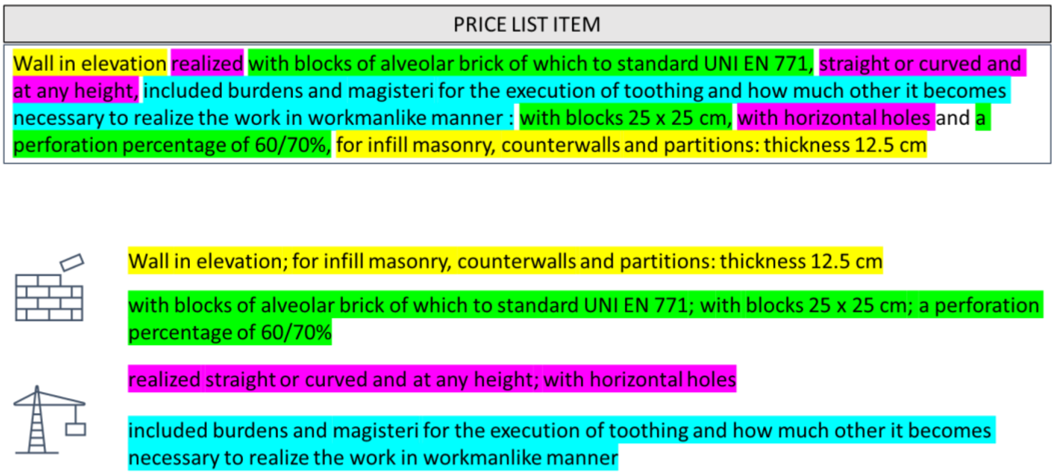


Figure 6: Example of non-linear description of a Price Item

For this reason, a new ontology was developed by the research group to standardize the cost item (Figure 7). First of all, two main groups have been defined to structure the data: one on elementary resources and one on construction works. Elementary resources (material resources, labor resources, equipment resources) are the basic elements for defining construction works. On the other hand, the construction work is the result of the process of combining work and activity ("*Laying of masonry layer*").

The "work" corresponds to the physical part of the construction work, what will be realized and created, for example, the "*layer of masonry*"; the work consists of elementary resources such as materials ("*hollow clay blocks*") or products ("*cement mortar*").

The "activity" corresponds to the action that must be performed to carry out the construction work, for example in new construction the "*laying or creation*" of the masonry layer, or in a maintenance activity the "*restoration or demolition*"; the activity will consist of elementary resources such as equipment ("*tower crane*") and labor ("*construction worker*").

The peculiarity of the structure allows to identify and associate each work ("*layer of masonry*") with different activities ("*laying, restoration, or demolition*"). This structure follows the logic of price analysis and allows the definition of the entire structure of the items of construction work from its elementary components (material resources, equipment resources, and labor resources).

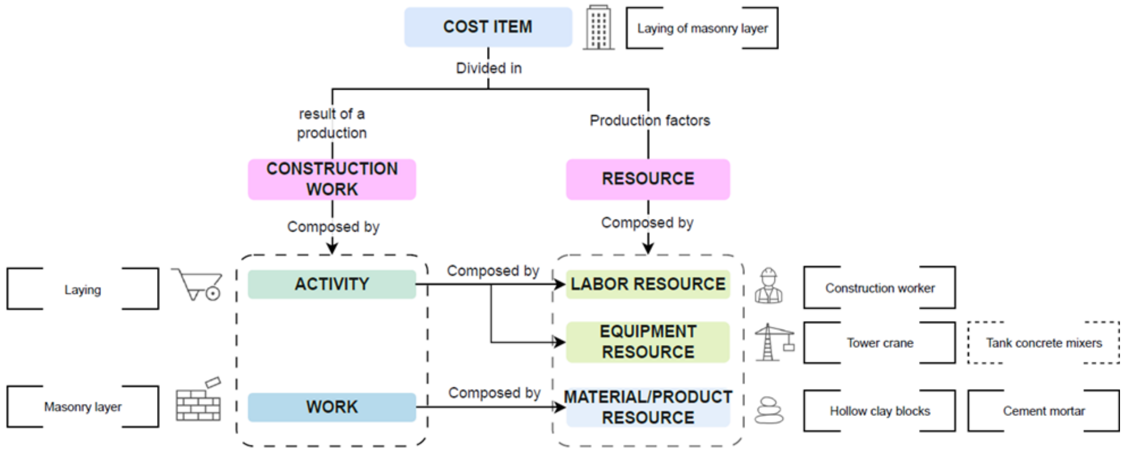


Figure 7: Conceptual structure of the cost item with an example of "Laying of masonry layer"

After having defined the logical structure that relates the various members of a cost item it has been necessary to define the structure for the definition of every single cost item. A standardized structure with defined attributes is necessary for breakdown and reorganization according to standardized rules of cost items. The description of cost items turns into an ontology that is more granular than a simple natural language description. This allow to:

- Place and store the information in specific attributes;
- Ensure validation and comparison activities between the cost class and the object class to which it is related;
- Consider the cost items not only as strings of text understandable by humans but as entities replicable and understandable by the machine.

Figure 8 shows a simplified example of the data sheet that defines a material resource. It is visible that the attributes allow to store:

- The *general information* related to the object to which a price will be associated:
  - Gender defines the type of resource (material, equipment or labor);
  - Family defines the group within which the resource is turned and for which work will be used (masonry, slab, insulation, etc.);
  - Category defines the sub-group of resources; in the material resources it defines the nature of the resource and if this is a system (element already mounted and only to be laid; for example, the monoblock window), a kit (set of disassembled elements, supplied jointly and to be mounted on site; for example ventilated facade), product (single element; for example, perforated block) and finally the accessory (completion element; for example door locks or handles).
- The *detailed information* related to the object to which a price will be associated:
  - Type defines any object types (for example, for the block if it is pierced, full or semi-full);
  - Material defines the material that constitutes the object (for example, cement, wood, brick, etc.);
  - Size and Performance Parameters define the size and performance characteristics of the object (for example, length, width, height, transmittance, etc.).
- The *additional information* related to the object to which a price will be associated:
  - Included or excluded define what is included or excluded from the price associated with the supply of the item (for example, in the supply price of the door can be included or excluded door handles, door locks, door covers, etc.);
  - Tech specs define the peculiarities of each object and specific characteristics.

MATERIAL RESOURCE

GENDER	FAMILY	CATEGORY	OBJECT	TYPE	MATERIAL	USE	SIZE PARAM.	PERFORMANCE PARAM.	INCLUDED	EXCLUDED	TECH SPECS
MATERIAL RESOURCE	*masonry*	*products*	*block*	*drilled*	*brick*	*masonry*	*length [cm] = 25 | width [cm] = 12 | heigth [cm] = 7*	*nd*	*nd*	*nd*	*nd*

Figure 8: Example of Data Sheet for the breakdown of a Material Resource Cost Item

Figure 9 shows a simplified example of the data sheet that defines construction work. In this case, two data sheets will allow to store the information of the work (physical entity) and the activity. The combination of these data sheets allows to define the description of the construction work. It is visible that the attributes allow to store:

- The *general information* related to the object to which a price will be calculated from the price of resources through the price analysis (gender, family, category)
- The *detailed information* related to the object to which a price will be from the price of resources through the price analysis (type, material, use, function, size/performance/physical parameters, etc.)
- The *additional information* related to the object to which a price will be from the price of resources through the price analysis (included, excluded, tech specs)
- The *resource* defining a relationship with the resource data sheets that are nested within the work (material and product resources) and activity (equipment and labor resources) defined according to the logic of price analysis.

Data sheet attributes are similar to ensure greater standardization and structuring of data. In this way, it becomes possible to interpret the information not just for humans but also for machines, through the use of standardized and structured ontology and semantics.

CONSTRUCTION WORK

GENDER	FAMILY	CATEGORY	OBJECT	MATERIAL	USE	SIZE PARAM.	INCLUDED	EXCLUDED	TECH SPECS	RESOURCE
WORK	*architecture*	*wall system*	*layer of masonry*	*brick*	*counterwalls | partitions*	*thickness [cm] = 12.5*	*burdens and magisteri ...*	*nd*	*nd*	*brick blocks 25x25 cm*

GENDER	FAMILY	CATEGORY	OBJECT	TYPE	USE	FUNCTION	INCLUDED	EXCLUDED	TECH SPECS	RESOURCE
ACTIVITY	*production*	*realization*	*laying*	*manually*	*architecture. wall system. layer of masonry*	*nd*	*nd*	*nd*	*nd*	*construction worker*

Figure 9: Example of Data Sheet for the breakdown of a Construction Work Cost Item (Work + Activity)

5. EXPERIMENTATION AND RESULTS

5.1 Cost Items in IFC

Starting from the definition of the new structure for the cost items, described in section 4.5, a method to represent cost information using IFC has been developed and validated. This could provide the basis for the information exchange resources among information systems and the development of a new structured and standardized cost domain. A previous study was done, by the same research group, using a simple case study to verify and validate the possible implementation (Cassandro et al., 2023).

A new architecture of cost items has been developed to consider them as a class *IfcCostItem* structured and more granular than a description in natural language. This entity is present in the IFC data model and contains a set of attributes where information can be instantiated. This entity can be related to different entities to define the new structure of cost items in IFC.

Cost items are structured according to a granular architecture based on relational logic, described in section 4.5. This architecture, translated into the IFC standard, is characterized by relations of nesting between the different

IfcCostItem entities starting from the elementary resources, through the work and the activity, and then to the construction work. Figure 10 shows an example of a conceptual scheme for the cost item "casting of a concrete masonry". Starting from the elementary resources (ready mixture of concrete, concrete pump, vibrator, and construction worker) that define the work (concrete wall layer) and the activity (casting with pump) it is possible to define the unitary construction work item (casting with pump of a concrete wall layer).

However, the entities IfcCostItem do not allow a complete and independent description of the cost items, for which it is necessary to activate the relations with other IFC entities (relations between *IfcControl* and *IfcConstructionResource*, *IfcElement*, *IfcTask*).

In this way, an ontology for cost items based on *IfcCostItem* entity was created. Since this entity is not supported by current BIM software, it is necessary to define this entity through the code. Starting from this structure, the new IFC cost items have been developed with code. The method was implemented using the IFC4_ADD2_TC1 - 4.0.2.1, IfcOpenShell, an open-source library (IfcOpenShell v0.7.0) and Python 3.10, (IfcOpenShell Contributors, 2020).

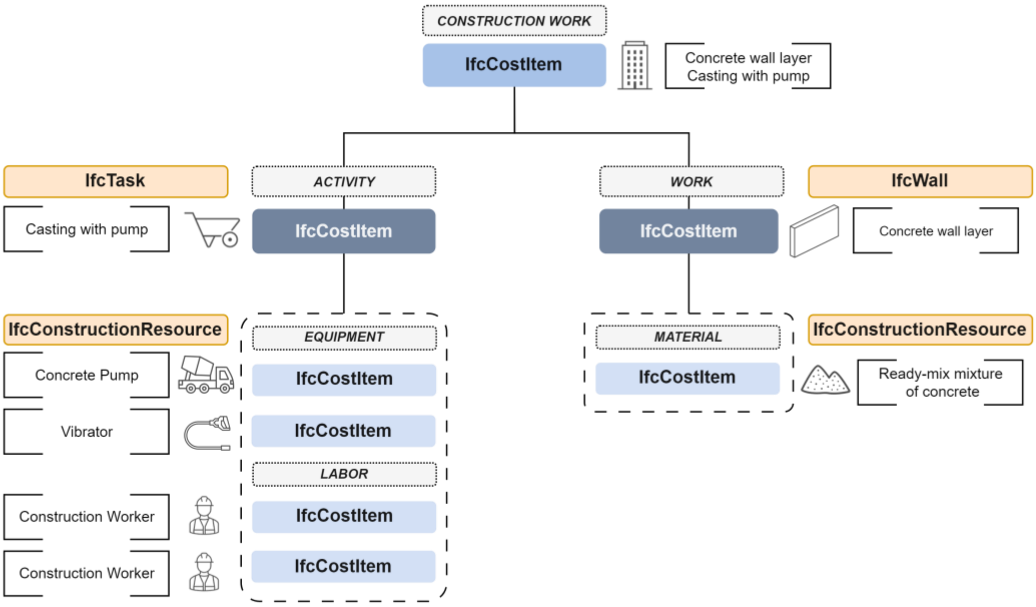


Figure 10: Conceptual framework of "Casting of Concrete Wall Layer"

The entity *IfcCostItem* represents the current cost item contained in the price list in a standardized format, open and interoperable. Then, every unitary *IfcCostItem* contains specific information and data based on the original items given by the price list. *IfcCostItem* entities are established with both inherited attributes (GlobalId, Name, Description, ObjectType, and Identification) and proprietary attributes (PredefinedType, CostQuantities and CostValues). Each useful attribute of the *IfcCostItem* is characterized by a specific definition fundamental to structure the data in an open and interoperable format:

- The *GlobalId* attribute allows the assignment of a *globally unique identifier* for each cost item.
- The *Name* attribute allows the definition of a label that *briefly describes the cost item*. It is a string that allows the immediate understanding of the voice by man with the meaning of natural language.
- The *Description* attribute allows the definition of a text describing the cost element in its entirety. It contains the *entire description of the cost item* (declaratory) and is an alphanumeric string of characters for information purposes that is intended to be read and understood by a human being.
- The *Identification* attribute allows the assignment of the *price list code* to the *IfcCostItem*. This attribute is an identifying designation given to a control and it is the identifier at the occurrence level.
- The *ObjectType* attribute denotes a particular type that further indicates the object. Use is established at the level of instant subtypes and in this research defines the *gender of the cost item* (Construction Work, Material Resource, Labor Resource, ecc.). This attribute is enhanced because the enumeration of the

PredefinedType attribute is set to USERDEFINED.

- The *PredefinedType* attribute describes a list of the available types of cost items. Currently, for the class IfcCostItem, enumeration types are not defined in the standard; there are only the types USERDEFINED and NOTDEFINED. Therefore, the enumeration of the PredefinedType attribute is always set to USERDEFINED. For this reason, the ObjectType attribute is enhanced.
- The *CostValues* attribute defines the *unit price of the cost element* that must be multiplied by the total of CostQuantities if provided. It is an amount of money characterized by the price of the elementary resource or construction work to which the overheads and profits must be added to define the unit price. It is characterized only by a list of inherited attributes (there are no own attributes) that allow the definition of a name, a description, an amount of an applied value (monetary amount), the number and the unit of measure on which the unit cost is based (1 cubic meter, etc.), the date from which and until which a value applied is applicable (period of validity of the item which for price lists appears to be annual with extensions of up to a further six months), the type of cost used (material, labor, overheads, profit, list price, etc.), the condition under which a cost value applies, the arithmetic operator applied to the values of the components if there are components (such as "ADD" to indicate the sum between unit amount, overheads and profit), a list of values of optional components from which AppliedValue is calculated (unit amount, overheads and profit).
- The *CostQuantities* attribute defines the component *quantities of the same type* for which the total quantity for the cost item is calculated as the sum. If CostQuantities are provided, values indicate unit costs, otherwise values indicate total costs.

The *IfcCostItem* entity has limitations regarding the information that can be added to its attributes. While some information about the price item can be included, other details related to the physical object cannot be directly contained within the cost entity. Therefore, it becomes necessary to establish relationships with specific IFC entities, groups of properties - PSET (*IfcPropertySet*), or sample elements (describe the physical aspect of the unitary object) to insert these values. This is because a computer class-based price list is created, where each item in the price list is an independent cost class that can be associated with the corresponding model object.

Through the HasAssignments attribute, inherited from the *IfcObjectDefinition* class, it is possible to activate the *IfcRelAssignsTo* relation and associate the instances of *IfcObject* and its various 1st level subtypes, such as *IfcProduct*, that contains for example, inside the *IfcElement*, the entity *IfcWall*. The sample element is exactly the unit value of the resource specified in the price list (such as the "Layer of Wall"). Each of these sample elements has specific physical information, geometric information, performance information, material information, and so on, that allow to define to what corresponds the cost value associated with a cost item. Through the IsDefinedBy attribute, inherited from the *IfcObjectDefinition* class, it is possible to activate the *IfcRelDefinesBy* relation and associate the *IfcPropertySet*. The *IfcPropertySet* is assigned via the *IfcRelDefinesByProperty* relation and contains the properties (*IfcComplexProperty* or *IfcSimpleProperty*) holding a set of values (*IfcPropertySingleValue*, *IfcPropertyEnumeratedValue*, *IfcPropertyListValue*, etc.) for each instance of cost items.

Both the sample element, the sample task, the sample resource, and the property sets hold key information that defines the cost item. This information is crucial for characterizing the cost item and will be useful in the next phase of the study. This information is described through the identification of IFC entities and the groups of properties (PSET) that explain them within the IFC standard. In the case of the concrete wall layer, to define a complete and detailed cost item architecture, we will need to clarify the following information:

- *The material of the sample entity* through the entity *IfcMaterial*; that constitutes the wall; will be defined vocabularies of terms associated with the entity as it is possible to indicate the concrete material with different synonyms including concrete, cement conglomerate or others.

The general properties of the sample entity through the relative property group (IfcPropertySet); in the case of the masonry layer (IfcWall) is represented by the "Pset_WallCommon" describing the general properties of the wall. For example, whether it is structural or not, etc (

- Table 1).

The specific properties of the sample entity through the relative property group (IfcPropertySet); such as the concrete properties described through the "Pset_ConcreteElementGeneral" property group already present in the standard and which defines the strength class, the exposure class, the structural class of *concrete, the construction method (prefabricated or cast in place), etc (*

Table 1: Example of general properties related to a sample element

GroupName	*PropertyName*	*Template / Value*	*Example*
Pset_Wall Common	Status	*P_ENUMERATEDVALUE / IfcLabel /PEnum_ElementStatus*	*NEW*
	FireRating	*P_SINGLEVALUE / IfcLabel*	-
	ThermalTransmittance	*P_SINGLEVALUE / IfcThermalTransmittanceMeasure*	-
	IsExternal	*P_SINGLEVALUE / IfcBoolean*	*TRUE*
	LoadBearing	*P_SINGLEVALUE / IfcBoolean*	*TRUE*

- Table 2).

Table 1: Example of general properties related to a sample element

GroupName	*PropertyName*	*Template / Value*	*Example*
Pset_Wall Common	Status	*P_ENUMERATEDVALUE / IfcLabel /PEnum_ElementStatus*	*NEW*
	FireRating	*P_SINGLEVALUE / IfcLabel*	-
	ThermalTransmittance	*P_SINGLEVALUE / IfcThermalTransmittanceMeasure*	-
	IsExternal	*P_SINGLEVALUE / IfcBoolean*	*TRUE*
	LoadBearing	*P_SINGLEVALUE / IfcBoolean*	*TRUE*

Table 2: Example of specific properties related to a sample element

GroupName	*PropertyName*	*Template / Value*	*Example*
Pset_Concrete ElementGeneral	ConstructionMethod	*P_SINGLEVALUE / IfcLabel*	*In-Situ*
	StructuralClass	*P_SINGLEVALUE / IfcLabel*	*S1*
	StrengthClass	*P_SINGLEVALUE / IfcLabel*	*C20/25*
	ExposureClass	*P_SINGLEVALUE / IfcLabel*	*XC1*

- *The base quantities of the sample entity* through the relative base quantities (*IfcQuantitySet*); the unit price for the concrete masonry layer is indicated per cubic meter and then the base quantities describing the sample element will be inserted. For example, for the masonry layer (*IfcWall*) the standard provides the quantities of length, width, height, footprint area, side area, and volume. All these quantities will be defined by unit value "1" as they indicate the price for a sample element of "1" cubic meter (Table 3).

Table 3: Example of base quantities related to a sample element

GroupName	*PropertyName*	*Template / Value*	*Example*
Qto_Wall BaseQuantiti es	Width	*Q_LENGTH*	*1 m*
	Length	*Q_LENGTH*	*1 m*
	Height	*Q_LENGTH*	*1 m*

NetFootprintArea	*Q_AREA*	*1 m²*
NetSideArea	*Q_AREA*	*1 m²*
NetVolume	*Q_VOLUME*	*1 m³*

The defined structure, which follows the logic of price analysis, will therefore allow to recreate all the relationships that contribute to the definition of a unit cost item through the use of IFC entities and the relationships between them. The validation of the output code is visible in an IFC graphic viewer (FZKviewer) or an IFC text viewer (IfcQuickBrowser). *IfcCostItem* represents an abstract entity without a graphic representation, then in the IFC graphic viewer will be visible only its attributes and the relationships it has with other entities.

It is therefore possible to see all the relationships that exist between the various entities of cost (construction work, resources, and sample element/task) in this new cost items architecture.

In each file containing the new cost items in IFC format and created according to the developed architecture, the following key elements can be identified:

1. *Elementary Resources:*

   These resources define what is needed to carry out the construction work, such as materials, equipment, or labor. Examples include construction workers, concrete pumps, and ready-mix concrete. Each resource includes several attributes, such as the amount needed (*IfcPhysicalQuantity*), the unit cost (*IfcCostValue*), and its relationship with other IFC entities. Specifically, this includes the physical entity (*WORK*) that will be realized.

2. *Activity and Work:*

   These entities break down the construction work into time-based activities (e.g., pump casting) and physical entities (e.g., concrete wall layer). They help organize resources for price analysis by defining the unit time (activity) and unit physical quantity (work) required for the construction work. These entities also contain attributes like the unit quantity (time for activities and physical quantities for work), cost values (*IfcCostValue*), and their relationship with other entities. For example, the material resources (such as ready-mix concrete) relate to the work (casting concrete with a pump), which will eventually form part of the physical entity (concrete wall layer).

3. *Construction Work:*

   This entity represents the cost item in the price list, see Figure 7, such as "casting with the pump of concrete wall layer". It includes all relevant information for defining the work, such as unit quantities (*IfcPhysicalQuantity*), cost values (*IfcCostValue*), and relationships with other entities. For instance, the cost value of construction work is calculated by adding the cost values of the work and activity (referred to as "A"), plus overheads (15% of A, in the case of the Lombardy Region public works price list, with a legal range of 13% to 17% in Italy), and the profit (10% of A+B). The final cost is the sum of "A", "B", and "C".

Figure 11, Figure 10, Figure 12, Figure 13, and Figure 14 illustrate the relationships between these entities, showing how the cost values are defined and validated during the experimental phase, as shown in Figure 15 and Figure 16.

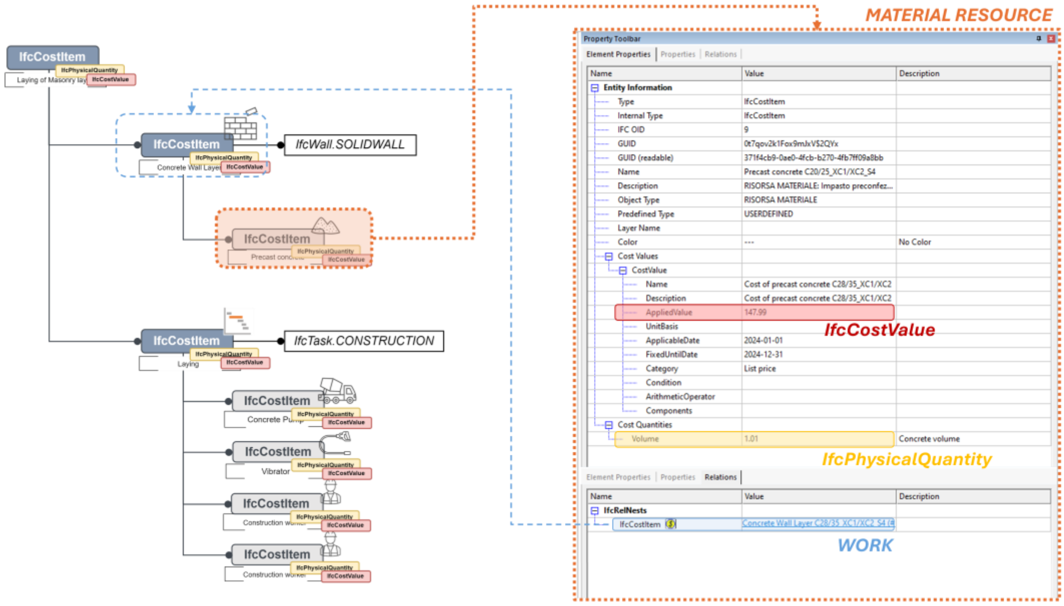


Figure 11: Validation of the cost item in IFC of a material resource

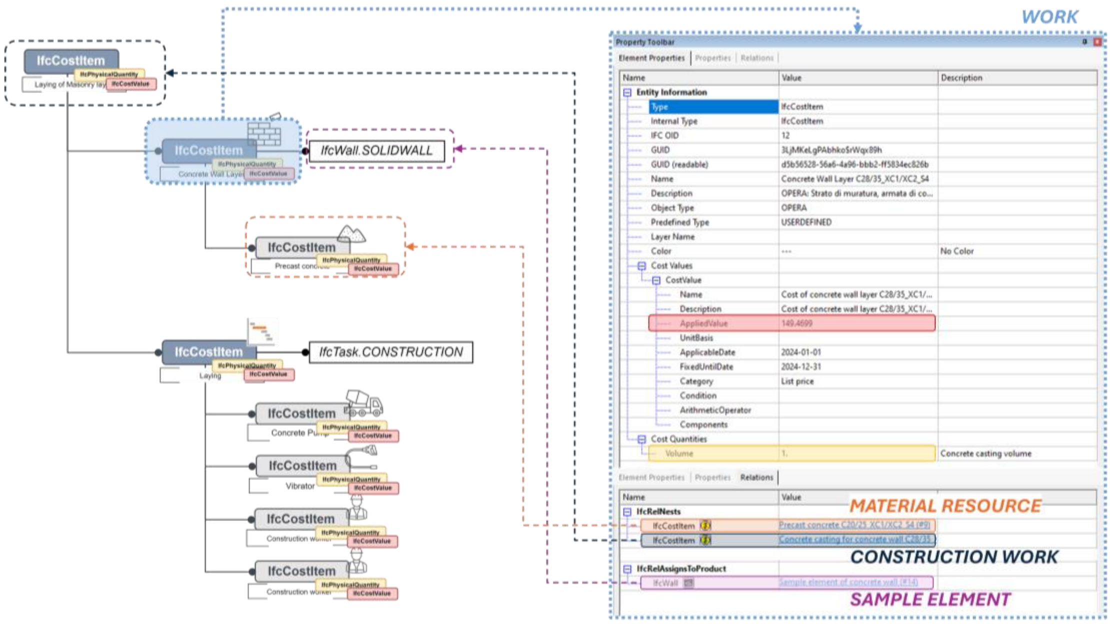


Figure 12: Validation of the cost item in IFC of a work

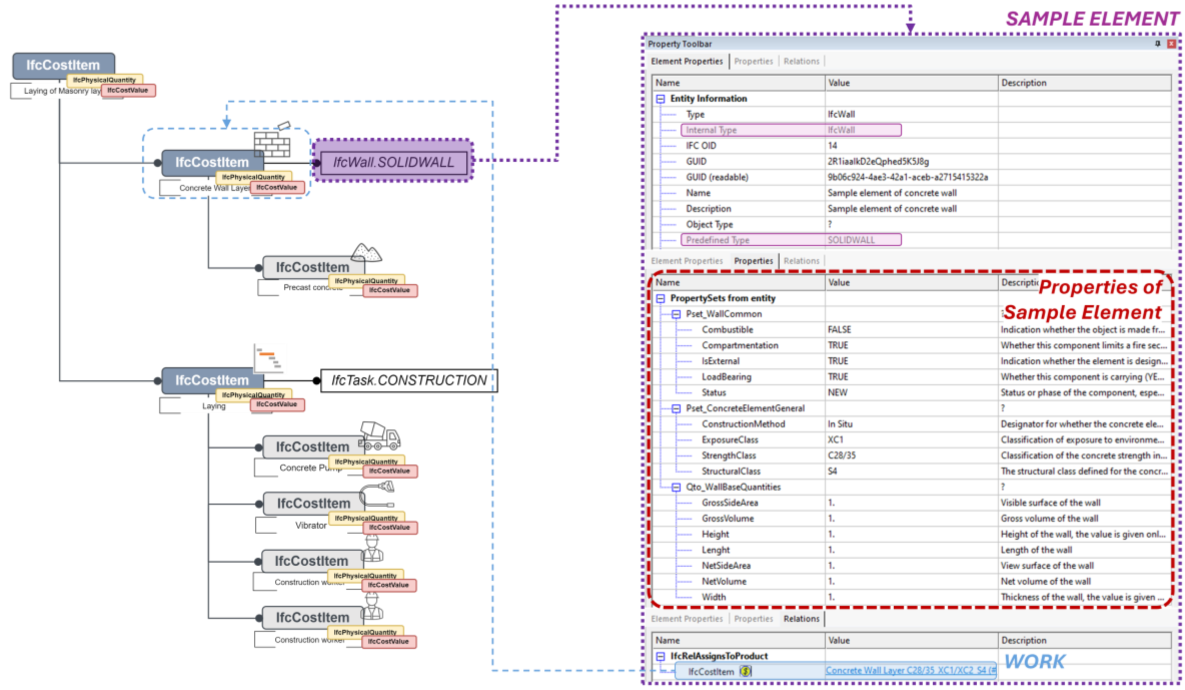


Figure 13: Validation of a sample item in the cost item in IFC

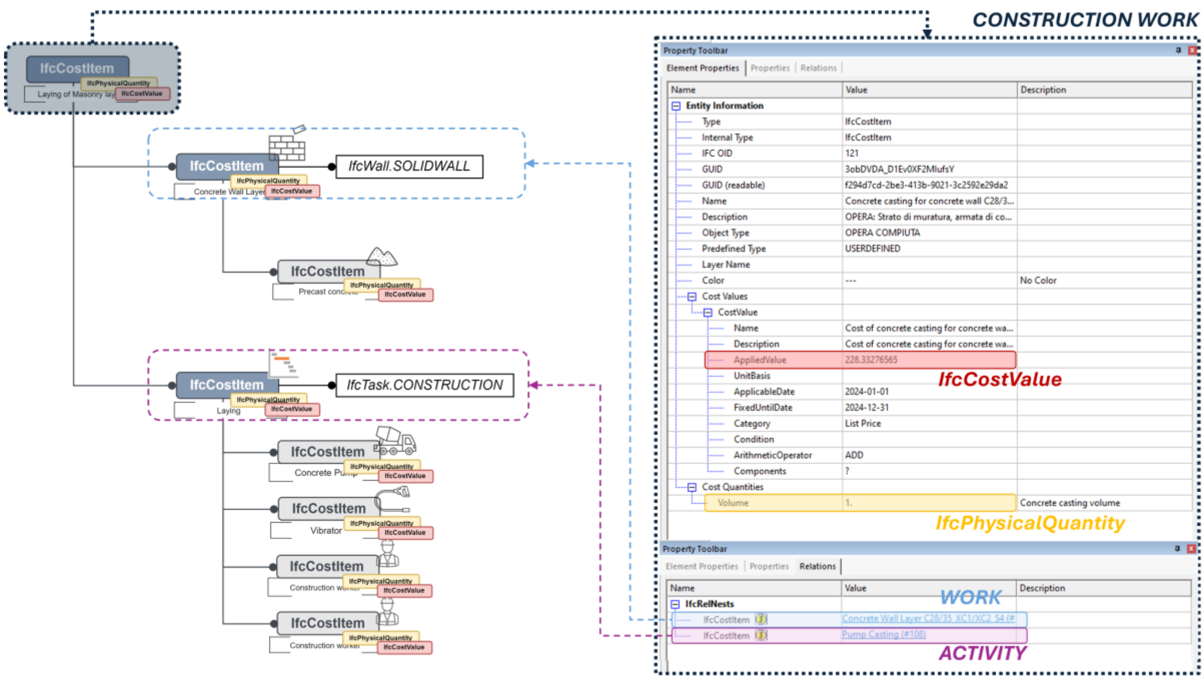


Figure 14: Validation of the cost item in IFC of a construction work

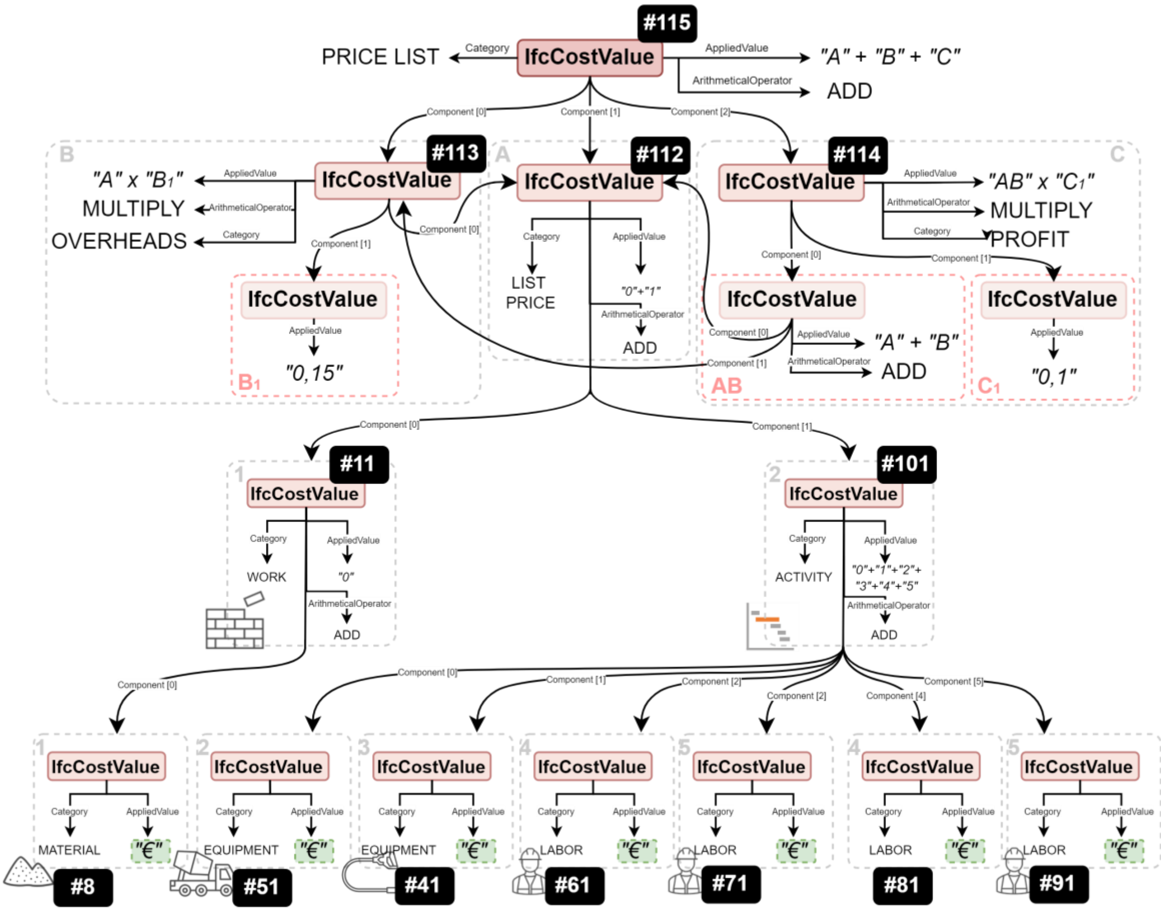


Figure 15: Structure of the cost value in IFC of a construction work

```
#115=IFCCOSTVALUE('Price List Cost of concrete casting for concrete wall C28/35_XC1/XC2_S4','Price List Cost of concrete casting for concrete wall C28/35_XC1/XC2_S4',IFCREAL(228.33276565),#1,'2024-01-01','2024-12-31','List Price',$,.ADD.,(#112,#113,#114));
  #1=IFCSIUNIT(*,.VOLUMEUNIT.,$,.CUBIC_METRE.);
  #112=IFCCOSTVALUE('Cost of concrete casting for concrete wall C28/35_XC1/XC2_S4','Cost of concrete casting for concrete wall C28/35_XC1/XC2_S4',IFCREAL(180.50021),#1,'2024-01-01','2024-12-31','Construction Work',$,.ADD.,(#11,#101));
    #1=IFCSIUNIT(*,.VOLUMEUNIT.,$,.CUBIC_METRE.);
    #11=IFCCOSTVALUE('Cost of concrete wall layer C28/35_XC1/XC2_S4','Cost of concrete wall layer C28/35_XC1/XC2_S4. Abstract entity of work',IFCREAL(149.4699),$,'2024-01-01','2024-12-31','Work',$,.ADD.,(#8));
      #8=IFCCOSTVALUE('Cost of precast concrete C28/35_XC1/XC2','Cost of precast concrete C28/35_XC1/XC2',IFCREAL(147.99),$,'2024-01-01','2024-12-31','Material',$,$,$);
    #101=IFCCOSTVALUE('Cost of activity to cast wall layer','Cost of activity to cast wall layer. Abstract entity of activity',IFCREAL(31.03031),#6,'2024-01-01','2024-12-31','Activity',$,.ADD.,(#41,#51,#61,#71,#81,#91));
      #6=IFCSIUNIT(*,.TIMEUNIT.,$,.SECOND.);
      #41=IFCCOSTVALUE('Cost of concrete vibrator','Cost of concrete vibrator',IFCREAL(9.67),#6,'2024-01-01','2024-12-31','Equipment',$,$,$);
      #51=IFCCOSTVALUE('Cost of powered pump','Cost of powered pump',IFCREAL(3.33),#6,'2024-01-01','2024-12-31','Equipment',$,$,$);
      #61=IFCCOSTVALUE('Cost of 3 level construction worker','Cost of 3 level construction worker',IFCREAL(31.23),#6,'2024-01-01','2024-12-31','Labor',$,$,$);
      #71=IFCCOSTVALUE('Cost of 2 level construction worker','Cost of 2 level construction worker',IFCREAL(29.02),#6,'2024-01-01','2024-12-31','Labor',$,$,$);
      #81=IFCCOSTVALUE('Cost of 1 level construction worker','Cost of 1 level construction worker',IFCREAL(26.07),#6,'2024-01-01','2024-12-31','Labor',$,$,$);
      #91=IFCCOSTVALUE('Cost of 1 level construction worker','Cost of 1 level construction worker',IFCREAL(26.07),#6,'2024-01-01','2024-12-31','Labor',$,$,$);
  #113=IFCCOSTVALUE('Overheads','Overheads in the Lombardy Regione Price List (15%)',IFCREAL(27.0750315),$,'2024-01-01','2024-12-31','Overheads',$,$,$);
  #114=IFCCOSTVALUE('Profit','Profit in the Lombardy Regione Price List (10%)',IFCREAL(20.75752415),$,'2024-01-01','2024-12-31','Profit',$,$,$);
```

Figure 16: Validation of the cost value in IFC text reader of a construction work

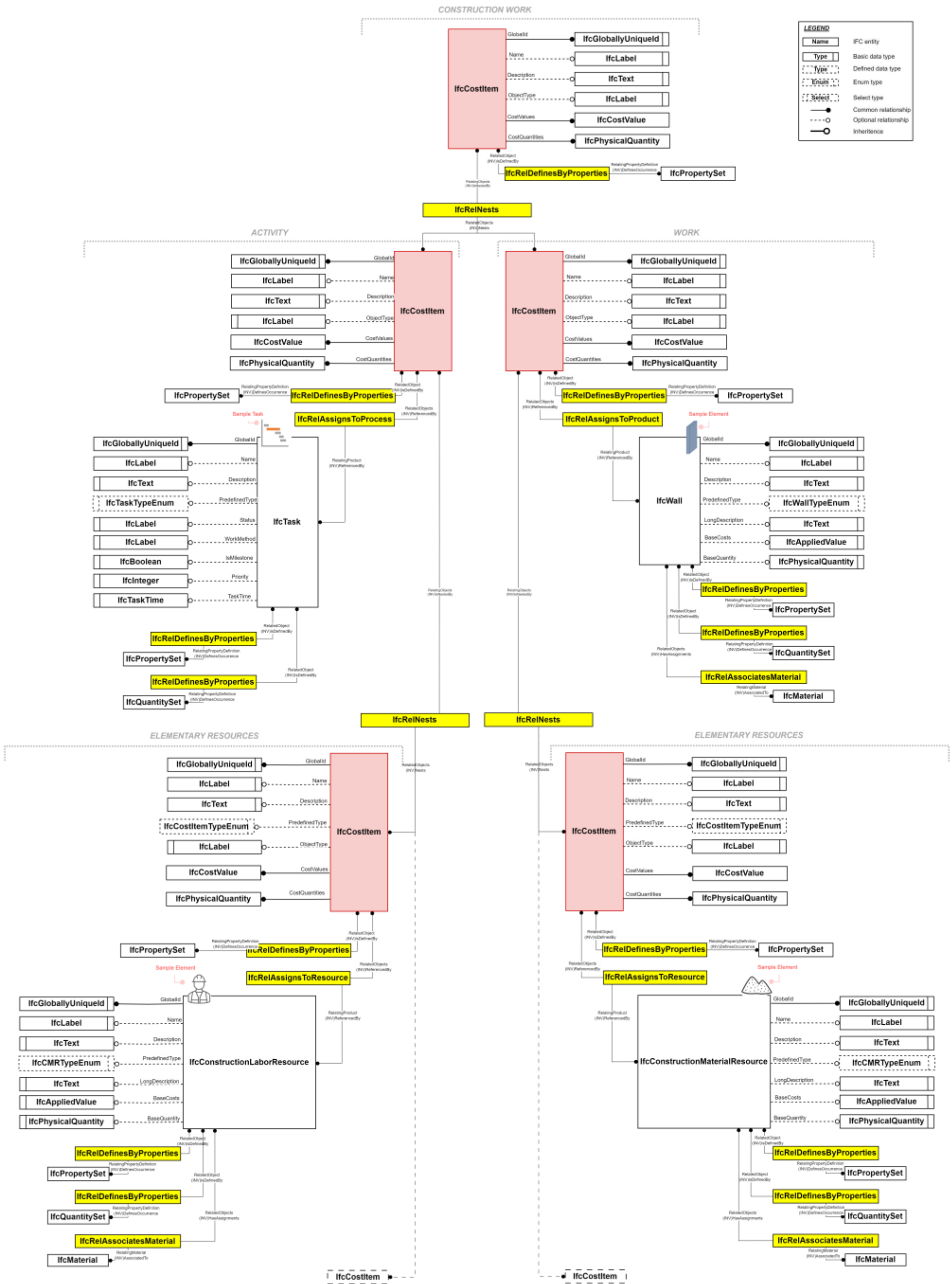


Figure 17: Architecture of IfcCostItem of "Concrete Wall Layer"

Figure 17 shows schematically the architecture defined and validated for the unit cost item "casting of a concrete masonry" as explained above.

From Figure 17 it is visible that the entity cost item of the construction work is related to two sub-entities cost items (work and activity); these are in turn related to the sample element or sample task and with the cost items of the elementary resources that constitute them.

5.2 Model Requirements

After conducting a detailed analysis of cost items and having structured them, the essential information necessary to ensure a correct association of cost items to the geometric model has been identified, the subsequent verification and validation of data between geometric entities and cost items.

The geometric model must include specific details, termed "Facets" in the standard IDS, to meet these requirements. Each facet comprises an applicability section, specifying the types of objects to which the facet applies, and a requirements section, outlining necessary properties or classifications for those objects. Metadata, including name, description, and instructions, precedes the applicability section to clarify the facet's purpose and implementation guidelines. For instance, we require all *IfcWall* entities within the geometric model to have both the "PredefinedType" attribute, adhering to predefined values such as PARTITIONING, SOLIDWALL, MOVABLE, NOTDEFINED, or USERDEFINED, and the “Name” and “ObjectType” attribute set to an unspecified value (Figure 18 and Figure 19).

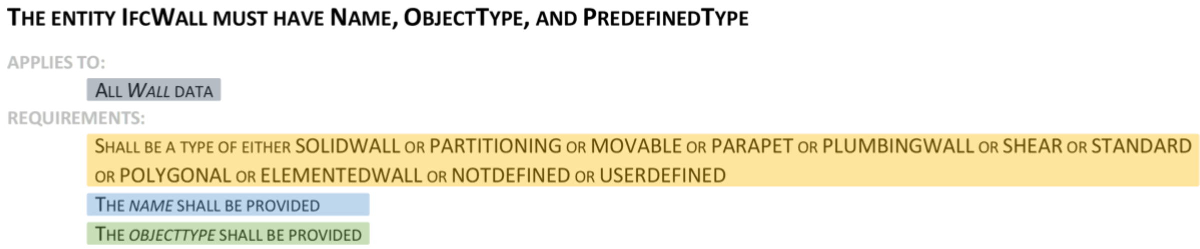


Figure 18: Simplified IDS user visualization

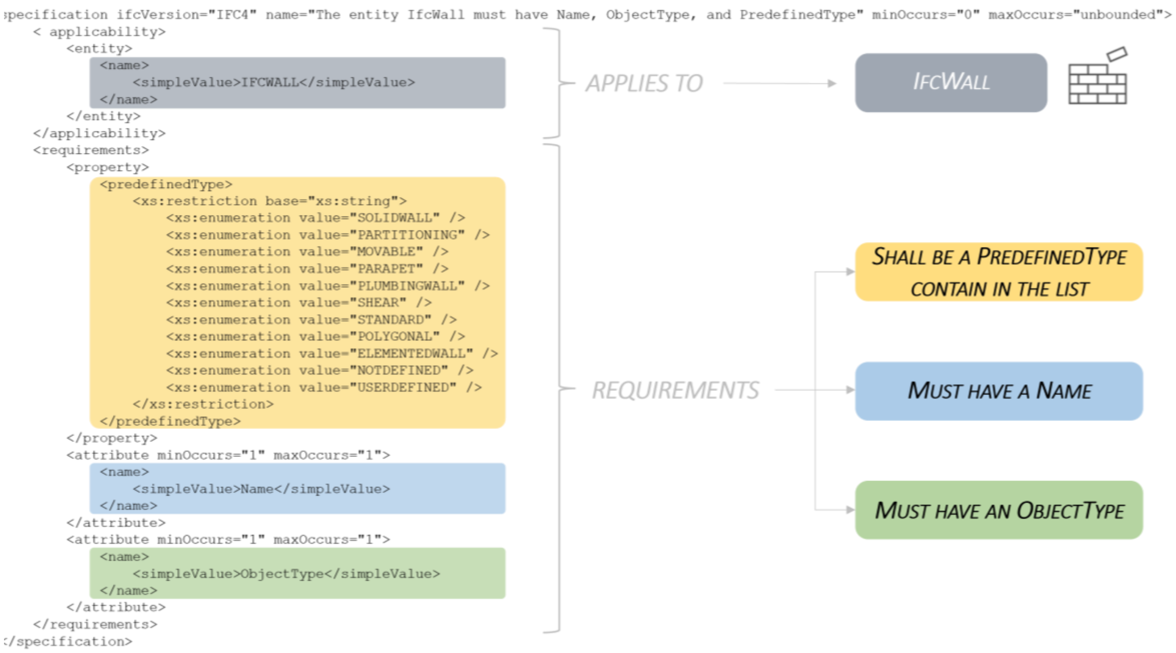


Figure 19: IDS document in machine-readable xml format

Table 4 consolidates the rules governing BIM information requirements. The example includes specifications for modeling structural walls. The IDS requirement is characterized, as explained in section 4.3, by three main parts (description, applicability, and requirements). While the first (description) helps the immediate understanding of the requirement by humans, the remaining two (applicability and requirements) allow the understanding and validation of the data by the machine. Specifically, an IDS of a concrete wall is characterized by:

- *Applicability* where any object that is exported as *IfcWall*.SOLIDWALL with a Loadbearing value set to “TRUE” must meet specific criteria.
- *Requirements* that describe all the requirements that *Applicability* must meet. Must contain an attribute "Name" valued; must contain the “Pset_ConcreteElementGeneral” with inside properties “ConstructionMethod”, “StrengthClass”, “ExposureClass”, “StructuralClass”,

“ReinforcmentVolumeRatio”, “ReinforcmentVolumeRatio” enhanced with specific data (In Situ, C28/35, XC1, etc.); must contain the “Pset_WallCommon” with within the general properties “FireRating” and “IsExternal” enhanced while the properties “LoadBearing” and “Status” enhanced with specific data (TRUE and NEW); finally, it must contain the “Qto_WallBaseQuantities” with all the basic quantities defined to enhanced standards (“Height”, “Width”, “Length”, “NetFootprintArea”, “NetSideArea”, “NetVolume”, etc.).

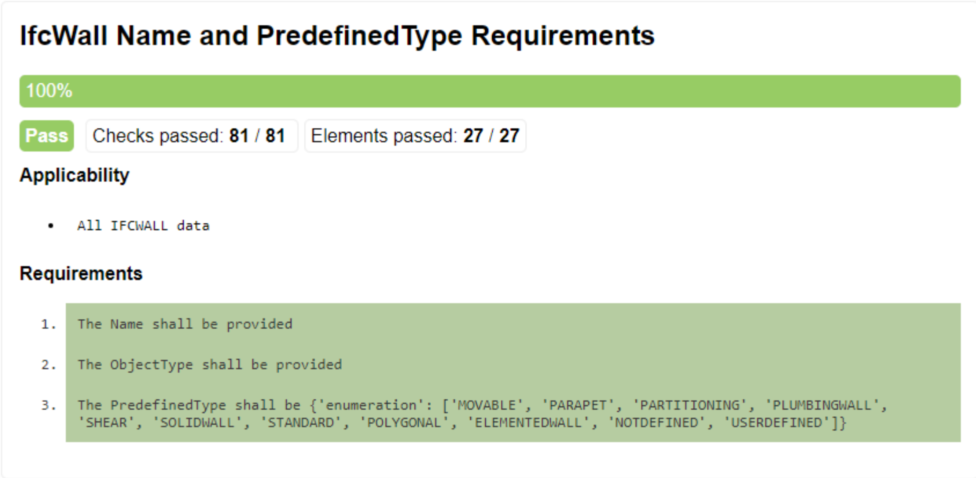
The requirements that require the use of specific values in the parameters, must be made explicit and inserted exactly as defined in standards; as regards the requirements that provide for the sole existence of the parameter, without the insertion of the specific value, it will be necessary their presence indistinctly of what inserted within them (in Table 4 indicated with "-").

Table 4 is an example of an IDS requirement; for all the requirements, please refer to the GitHub page (section 0).

Table 4: Example of requirements for modeling structural concrete walls

	Applicability	Requirements		
Requirement	*IfcWall*.SOLIDWALL	Attribute	Name	-
	Loadbearing TRUE	Pset_ConcreteElementGeneral	ConstructionMethod	In-Situ
			StrengthClass	-
			ExposureClass	-
			StructuralClass	-
			ReinforcementVolumeRatio	-
			ReinforcementStrengthClass	-
		Pset_WallCommon	FireRating	-
			IsExternal	-
			LoadBearing	TRUE
			Status	NEW
		Qto_WallBaseQuantities	Height	-
			Width	-
			Length	-
			GrossFootprintArea	-
			NetFootprintArea	-
			GrossSideArea	
			NetSideArea	
			GrossVolume	
			NetVolume	
			GrossWeight	
			NetWeight	-

Verification of these requirements and the geometric model's accuracy was conducted using a developed code starting from “IfcTester” (Dion Moult, 2024). This is a library useful to create and read IDS files and validate IFC models to verify compliance. In addition, you can generate reports of the performed validation, saving them as web page, JSON, or BCF files. This process enabled the validation of information within the model and the identification of any differences from the initially defined requirements. From the verification carried out through the developed IDS, it has been possible to validate the correctness of the information inserted in the geometric model and to correct it. Below, Figure 20 shows the validation of general data of wall objects, *IfcWall*, through the IDS visible in Figure 18. All the objects (27 instances of *IfcWall*) have been verified and each of these instances meets the three defined requirements. Instead, Figure 21 shows the validation of requirements for concrete walls, through the IDS visible in Table 4. In this case, none of the objects (15 instances of *IfcWall* with the specific ObjectType) fully meet the requirements; among the 15 instances, only 148 requirements, compared to the total of 195 checked, are passed. It will then be necessary to make corrections to the geometric model and to perform the IDS verification again to ensure consistency with the established requirements and pass the validation.



IfcWall Name and PredefinedType Requirements

100%

Pass | Checks passed: **81** / **81** | Elements passed: **27** / **27**

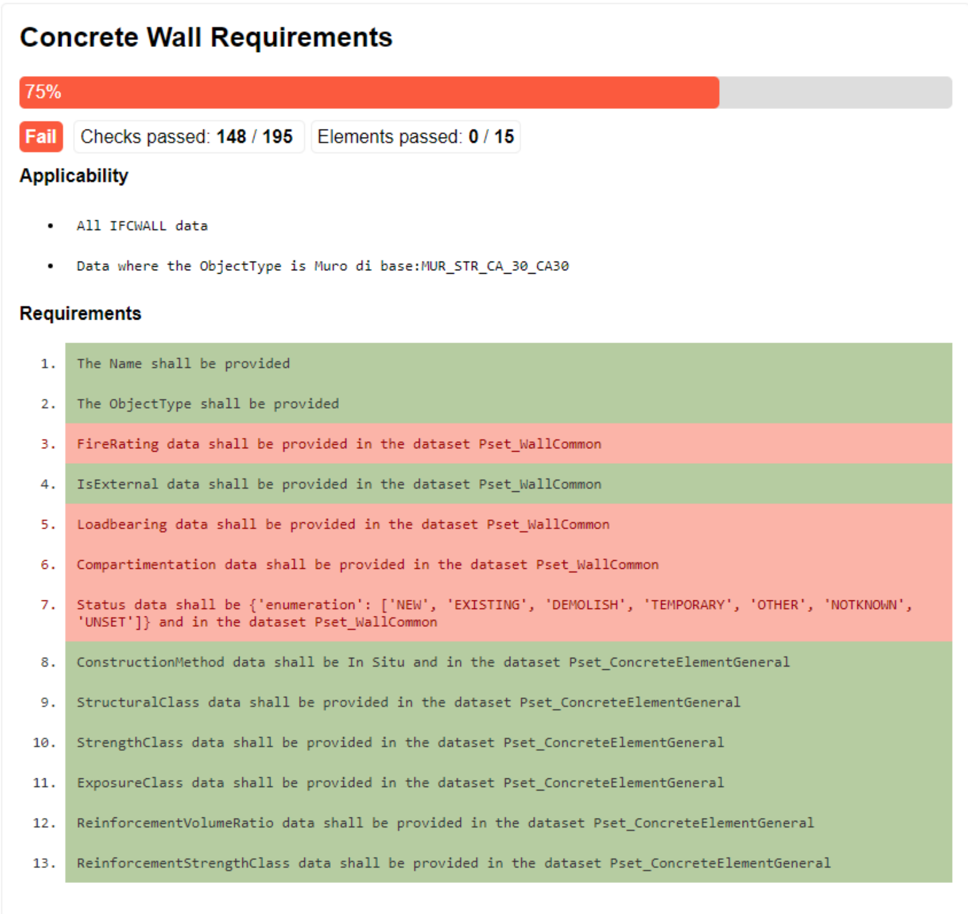
Applicability

- All IFCWALL data

Requirements

1. The Name shall be provided
2. The ObjectType shall be provided
3. The PredefinedType shall be {'enumeration': ['MOVABLE', 'PARAPET', 'PARTITIONING', 'PLUMBINGWALL', 'SHEAR', 'SOLIDWALL', 'STANDARD', 'POLYGONAL', 'ELEMENTEDWALL', 'NOTDEFINED', 'USERDEFINED']}

Figure 20: Validation of general data of wall objects



Concrete Wall Requirements

75%

Fail | Checks passed: **148** / **195** | Elements passed: **0** / **15**

Applicability

- All IFCWALL data
- Data where the ObjectType is Muro di base:MUR_STR_CA_30_CA30

Requirements

1. The Name shall be provided
2. The ObjectType shall be provided
3. FireRating data shall be provided in the dataset Pset_WallCommon
4. IsExternal data shall be provided in the dataset Pset_WallCommon
5. Loadbearing data shall be provided in the dataset Pset_WallCommon
6. Compartimentation data shall be provided in the dataset Pset_WallCommon
7. Status data shall be {'enumeration': ['NEW', 'EXISTING', 'DEMOLISH', 'TEMPORARY', 'OTHER', 'NOTKNOWN', 'UNSET']} and in the dataset Pset_WallCommon
8. ConstructionMethod data shall be In Situ and in the dataset Pset_ConcreteElementGeneral
9. StructuralClass data shall be provided in the dataset Pset_ConcreteElementGeneral
10. StrengthClass data shall be provided in the dataset Pset_ConcreteElementGeneral
11. ExposureClass data shall be provided in the dataset Pset_ConcreteElementGeneral
12. ReinforcementVolumeRatio data shall be provided in the dataset Pset_ConcreteElementGeneral
13. ReinforcementStrengthClass data shall be provided in the dataset Pset_ConcreteElementGeneral

Figure 21: Validation of requirements for concrete walls

5.3 Cost-Object Relation

After the development of the unit cost items and verification with the IDS of the geometric model, a code has been developed to be able to relate the cost items to the geometric objects obtaining, therefore, a new BIM model containing both the geometric information and the information of cost.

The code is characterized by 3 steps: (1) querying the IFC geometric model (Figure 22), (2) querying the IFC cost item database (Figure 23), (3) creating the relationship between cost item and geometric objects and creating of new BIM model (Figure 24).

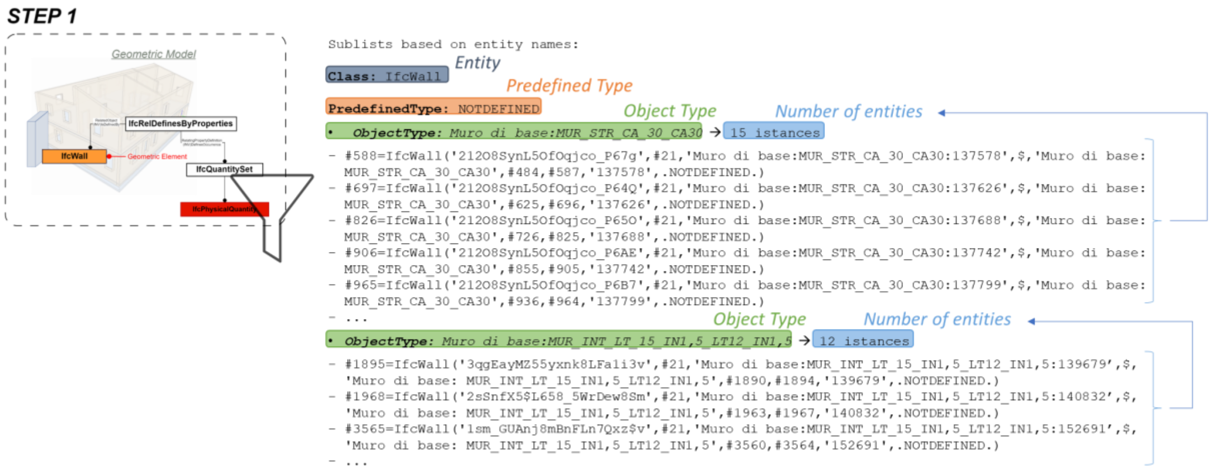


Figure 22: Example of two groups of geometric objects divided by type in the same entity identified by the code

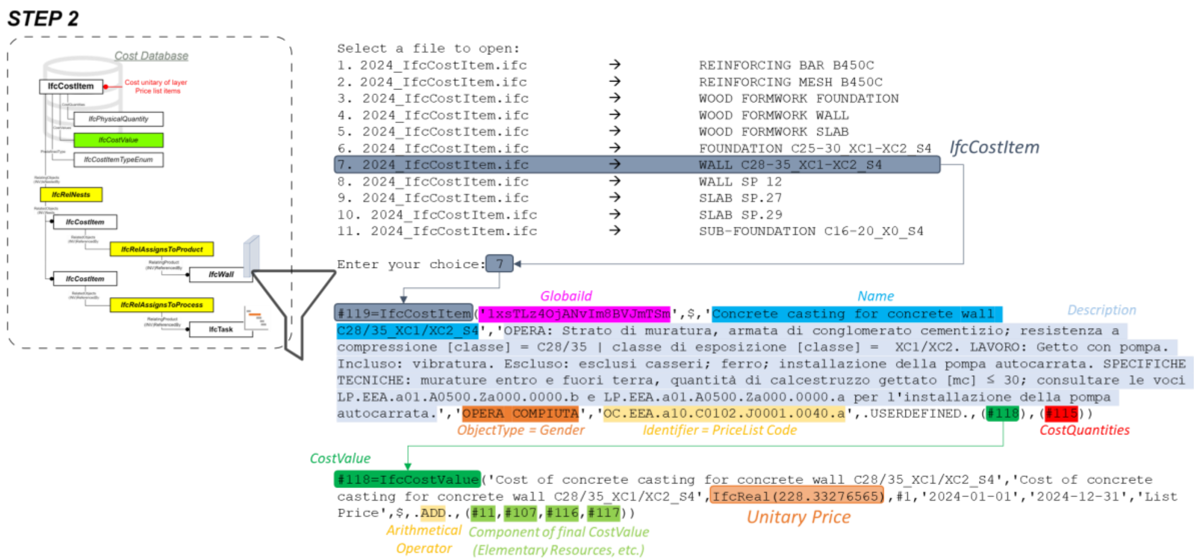


Figure 23:Example of a query of the cost item database

Step (1) involves a first query of the geometric model to identify the objects to which the cost items must be reported for a successive estimate of the costs.

The model query allows to automatically define lists of objects divided by IFC entities, enumerative typology (PredefinedType), and finally by object type. Groups of geometric objects divided by type will then be identified, also indicating the number of instances (Figure 22). At this point, after selecting the list of instances (ObjectType) of interest, begins the process step (2).

The previously created cost database containing cost items in IFC is queried for the choice of unit cost item (currently contains items useful for experimentation, but in the future could contain all the items of a price list). The unit cost item is then selected and related to the list of geometric objects (Figure 23). As a result of these inputs, a new entity *IfcCostItem* will be created. This is step (3) of the process. In these new cost items, the attributes Name, Description, Object Type, Identification, PredefinedType, and CostValues are inherited from the unit cost item and instead in the attribute CostQuantities the amount of information is retrieved from geometric objects (Figure 24).

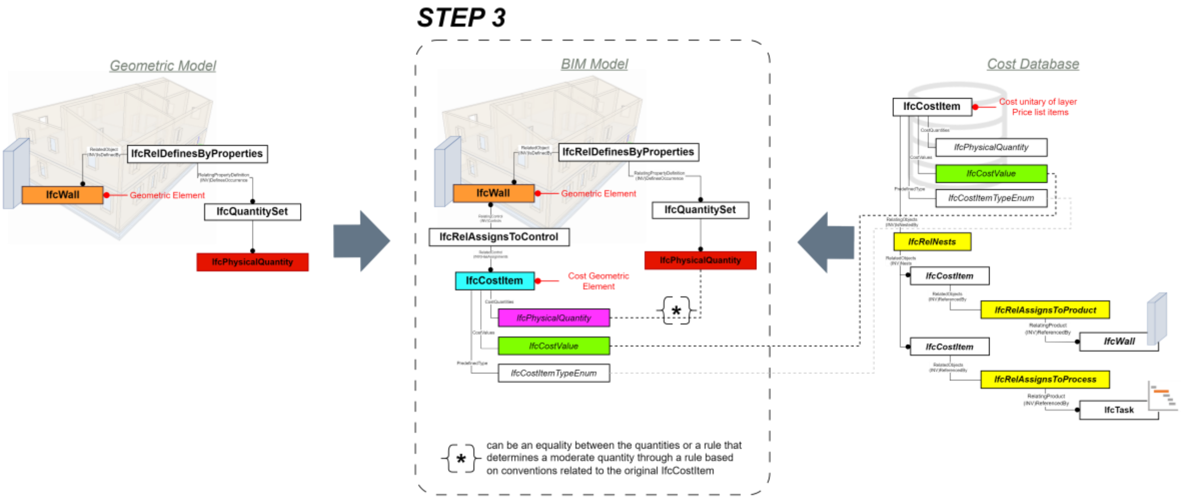


Figure 24: Example of new BIM model with cost and geometric information

However, the quantities do not exactly match those defined in the geometric object (physical quantities) but will be moderate based on measurement rules contained in the unit cost item (these measurement rules are not defined manually by the user as commonly happens).

These quantities are essential for a correct cost estimate and a correct bill of materials (for example, we do not consider the net volume quantity of the wall, but the net volume quantity multiplied by a scrap of 10%). The code will then allow the iteration of operations in the same group of instances or passing to the following groups defined above.

After the phase of the relation between cost items and geometric objects, it will be possible to save the new IFC model containing now also the cost information related to the objects.

In Figure 25 the model without cost data is represented and in Figure 26 the new BIM model is validated and enriched with cost data.

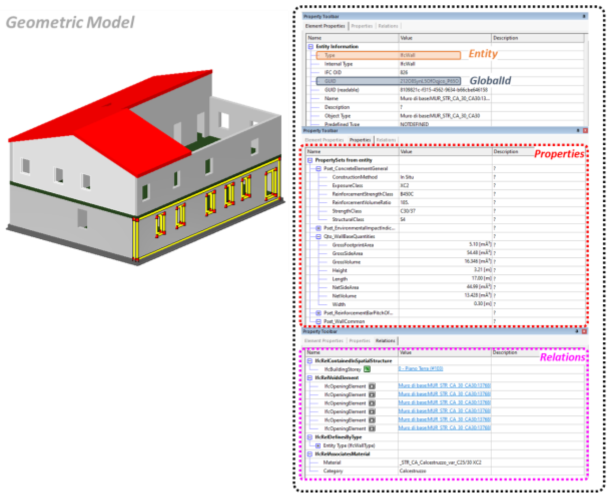


Figure 25: Geometric model without cost data

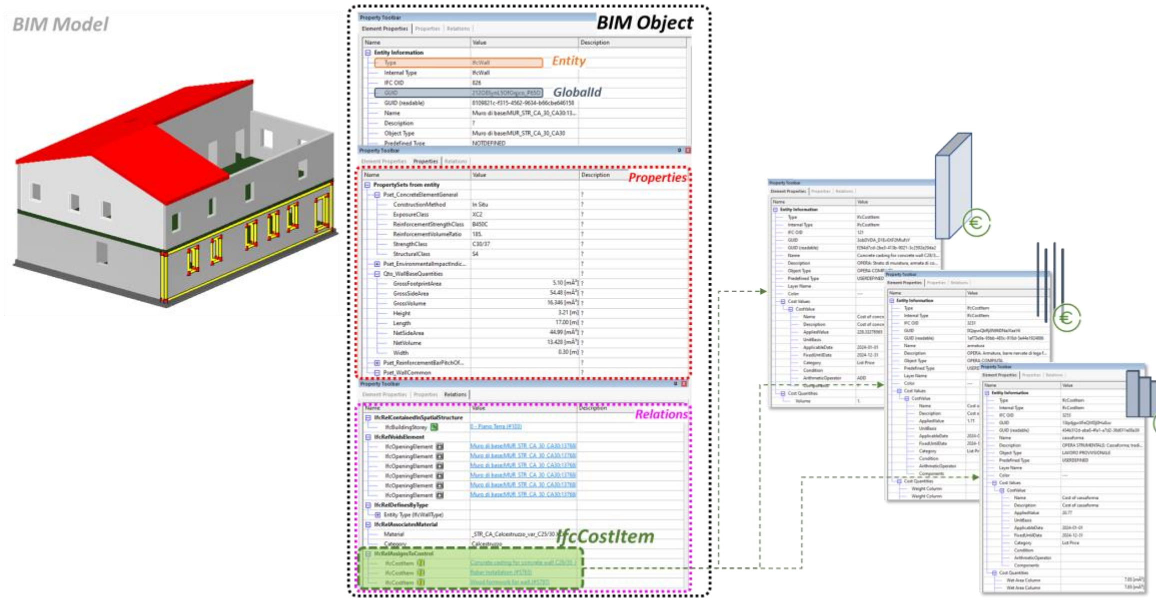


Figure 26: New BIM model with cost data

The validation of the model was done with a graphical IFC viewer (FZKviewer). Are visible new relationships (*IfcRelAssignsToControl*) that allow to relate the geometric object (*IfcWall*) with cost items (*IfcCostItem*) for the definition of the cost estimate (cost of casting concrete wall, laying formwork and laying rebar).

5.4 Verification of the uniqueness and completeness of data

A code developed using Python 3.10 and IfcOpenShell, an open-source library (IfcOpenShell v0.7.0), allowed an analysis of the new BIM model, in IFC format. It was possible to validate the correctness and uniqueness of the relationships between cost items and geometric objects defined in section 5.3.

The validation process discussed in this study is different from the current methods. Nowadays, verifications are performed only within the same domain. However, this research aims to verify the accuracy and correctness of information between two different domains, the geometric domain, and the cost domain.

Figure 27 shows schematically how the validation process allows to verify the correctness of the information. The analysis involves a detailed checking of the information within geometric objects of the 3D model against corresponding data stored in the cost database, including PropertySets, Materials, etc. While the cost item architecture is not directly integrated into the geometric model, each cost entity (*IfcCostItem*) within the model holds key data for cost estimation, such as the Identification attribute. This attribute contains the price list code useful to find the cost item in the cost database.

This approach ensures a relationship between cost estimation items in the BIM model and unit cost items in the cost database without burdening them with unnecessary data.

The data analysis is characterized by three fundamental steps: (1) search for all costs in the BIM model and related geometric objects to which they are related, (2) analysis of the data contained in the cost and geometric entities (attributes, property sets, etc.), (3) verification of the correctness between the data and development of the final report.

Once the code is started, it will automatically perform all the analyses necessary for the verification of the items in the BIM model. Specifically for the case study of this research, different levels of verification and control of the correctness of the data have been set according to the fundamental characteristics for which the cost item is defined.

The analysis has therefore allowed to analyze specifically the information present in the cost items (this information is fundamental for the definition of the price) with the respective attributes of the geometric objects to which they have been related.

Therefore, different verification codes have been developed based on the different cost items and objects to be verified.

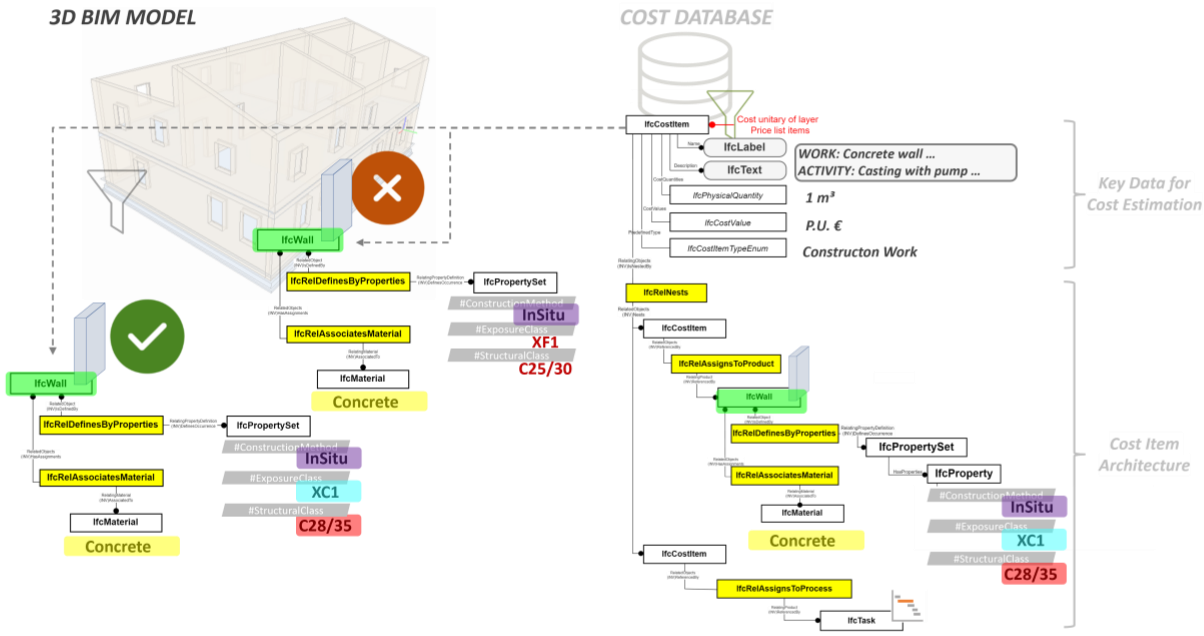


Figure 27: Example of checking the correctness of the cost item relationship with geometric objects

Below, in Table 5, are schematically listed examples of data that have been the subject of verification for an *IfcWall*. What is depicted is an example of what has been elaborated and developed. Testing was performed on all entities of the analyzed model (*IfcRoof*: 1 instance, *IfcWall*: 27 instances, *IfcSlab*: 6 instances) and has allowed to establish the correctness or less of what was reported during the phase of the relation of the cost items described in the section 5.3.

Table 5: Examples of data that are subject to verification for an IfcWall

Cost Item	Object	Data Verification	Data Type
Casting of concrete wall	IfcWall	Entity	*Class Name*
		PredefinedType	*Class Attribute*
		Thickness of wall	*Quantity Set*
		ConstructionMethod	*Property Set*
		StructuralClass	*Property Set*
		StrengthClass	*Property Set*
		ExposureClass	*Property Set*
		Status	*Property Set*
		Loadbearing	*Property Set*
Laying of Wood Formwork	IfcWall	Entity	*Class Name*
		PredefinedType	*Class Attribute*
		Height of wall	*Quantity Set*
		Status	*Property Set*
		Loadbearing	*Property Set*
Laying of Reinforcing Bar *	IfcWall	Entity	*Class Name*
		PredefinedType	*Class Attribute*
		ReinforcementVolumeRatio	*Property Set*
		ReinforcementStrengthClass	*Property Set*
		Status	*Property Set*
		Loadbearing	*Property Set*

* This procedure is used because the provided geometric model does not contain the reinforcement bar modeling. If they had been present it would have to be verified with the data of the *IfcReinforcingBars*.

For each object to which cost items have been related, a code has been developed because the validation rules differ according to the type of construction work and object to which they relate. For the casting of a concrete wall, it is essential, for example, to verify the type of concrete used and whether the information in the geometric model corresponds to those of the cost related.

For other construction work, such as laying the formwork, it is essential to check whether the same is suitable for use on walls, foundations, or floors because the price and type of formwork differ by type of processing and application.

After querying the IFC model, results are validated, and inconsistencies are flagged. In Table 6 you can view the attributes and PSET validated between geometric objects and cost objects and the result of the validation (✔ = Validated, ✖ = Not Validated, ND = "Not Defined"). From these results, users can therefore:

- Maintain the cost-object relationship, recognizing any inconsistencies.
- Adjust the original cost item via the cost database query, choosing between proposed changes or creating a new cost item if necessary. However, this latter option remains a future development of the study.

Furthermore, a report detailing the analysis results for each relation is generated. Figure 28 illustrates an example of the data obtained from the verification between a cost item for concrete casting for a structural wall and its geometric object (wall). The generated report provides insights into the consistency of data between the geometric object and its associated cost item, as depicted in Table 6.

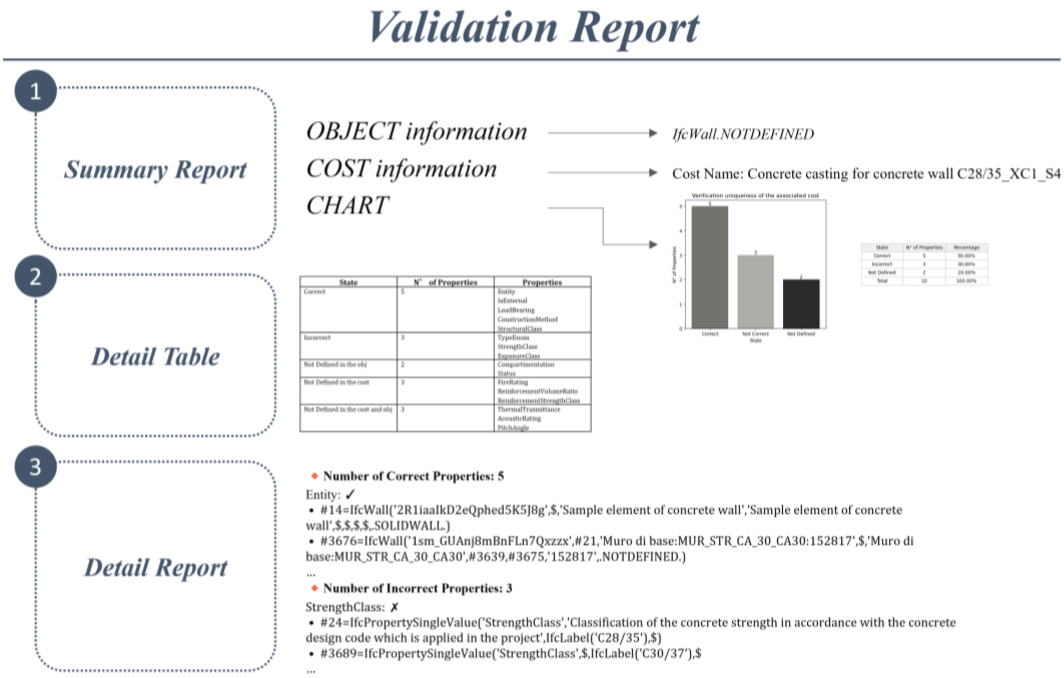


Figure 28: Example of the Validation Report developed by the code

The validation report, as visible in Figure 28, is characterized by 3 main parts:

- The first part contains a *Summary Report* that briefly describes the entire analysis, explaining the information of the analyzed geometric object (Entity, PredefinedType, GlobalId, Name), of the cost item related to it (Entity, PriceListCode, GlobalId, Name) and a summary graph of the validation with the percentage of "correct", "incorrect" and "not defined" data. These data are fundamental for defining the price of the cost item but are not included in the geometric object.
- The second part contains a *Detailed Table* showing the states of verification ("correct", "incorrect", "not defined in the obj", "not defined in the cost", "not defined in the cost and obj"), the number and list of properties that are contained in the states.
  The states differ in:

- "Correct" indicates all properties that are suitable between cost item and geometric object.
    - "Incorrect" indicates all properties that are not suitable between cost item and geometric object.
    - "Not defined in the objects" refers to the properties that are included in the cost item and are essential to determining the price of the construction work. These properties are not found in the analyzed geometric object.
    - "Not defined in the cost" refers to the properties that are not included in the cost item and are not essential to determining the price of the construction work. These properties are found in the analyzed geometric object.
    - "Not defined in the cost and obj" refers to the properties that are not included in the cost item and the analyzed geometric object.
- The third part contains the *Detailed Report* of all the analyses performed; each section of the detailed report is characterized by the number of properties ("correct", "incorrect", etc.), the type of properties (Entity, PredefinedType, ConstructionMethod, etc.) and the complete description of the IFC entities that have been verified for the validation of the correctness and uniqueness of the relationship between cost item and geometric object. These are shown in their standard version, with their attributes (Figure 29).

Table 6: Verified parameters between the geometric object (IfcWall) and associated cost item

Entity	**Attribute/PSets**	**Parameter Name**	**Geometric Object**	**Cost Item**	**Check**
IfcWall	Attribute	Entity	IfcWall	IfcWall	✓
		PredefinedType	NOTDEFINED	SOLIDWALL	✗
	Pset_ConcreteElementGeneral	ConstructionMethod	In Situ	In Situ	✓
		StrengthClass	C30/37	C28/35	✗
		ExposureClass	XC2	XC1	✗
		StructuralClass	S4	S4	✓
		ReinforcementVolumeRatio	100	-	-
		ReinforcementStrengthClass	B450C	-	-
	Pset_WallCommon	FireRating	-	-	-
		IsExternal	TRUE	TRUE	✓
		LoadBearing	TRUE	TRUE	✓
		Status	-	NEW	ND
		AcousticRating	-	-	-
		PitchAngle	-	-	-
		ThermalTransmittance	-	-	-
		Compartmentation	-	TRUE	ND

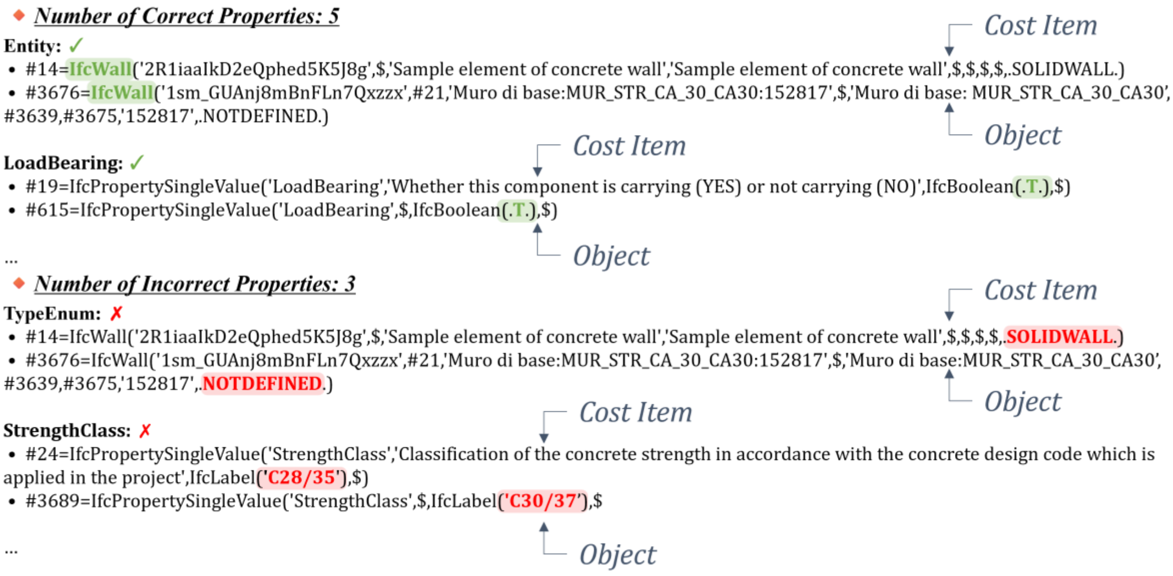


Figure 29: Detailed Validation between Cost Data and Geometric Data

6. DISCUSSION

This study is part of a larger research project, in collaboration with Lombardy Region, aiming to digitize and standardize all cost items currently in the Lombardy Region Price List and identify a new cost domain for the creation of a new regional digital pricing platform web based. These new structured and standardized cost data can be directly related to other data such as those in the 3D BIM model. This allows the creation of relationships between different geometric and cost entities characterized by information and different architectures; which guarantee also to check the correct relationship created between cost elements and geometric objects to optimize the cost estimation.

Given the high uncertainty and inaccuracy in cost estimation processes, it is crucial to:

- Structure cost items in a currently natural language to define machine-readable data.
- Define information requirements that the model must contain (Level of Information Need) for correct cost estimation.
- Define a semi-automated control procedure for these information requirements to avoid errors and save time. This is achievable through a standardized document (IDS) for verifying information in IFC models.
- Verify the correctness and uniqueness of cost elements related to geometric objects to ensure greater accuracy in cost estimation.

Currently, cost estimation is a critical task in the AEC/FM sector. Therefore, to support, verify, and improve its quality in public tenders and reduce human mistakes, the study proposes to structure the current cost elements in a standardized scheme and to define an open and interoperable cost domain. This allows cost information to be related to geometric objects without the need to create specific attributes within the geometric model that contain codes or references to unit cost items. In this context, current practices often involve linking codes in geometric elements to corresponding keys with various cost items (Pavan et al., 2017) for performing the subsequent cost estimation in 5D BIM software.

Initially, the research structured and validated a new cost item architecture (cost entity) in IFC format to define a cost domain. These new entities have allowed to enrich the geometric model and create a BIM model characterized by the presence of entities belonging to different domains (cost and geometry) related to each other. In this way, it is possible to create a link between "n" cost entities and a single geometric object. Each geometric object can be characterized by several construction works with different cost values that contribute to its realization.

Consider for example a concrete wall. It is necessary to estimate the costs of reinforcement laying, formwork laying, and concrete casting. The new architecture guarantees a greater granularity of data than a simple description in natural language, as in the current Italian price lists. It will also allow to define in detail the elementary resources of a cost item, the necessary quantities of them, and their cost value according to the logic of price analysis. Currently, the new architecture of cost items is based on the hardcoded relationships and properties that belong to the IFC standard.

Then, the study defines and develops a procedure to validate the correctness of cost data related to geometric objects within the IFC data model. This scientific research has led to technological attempts through writing Python code and using the IfcOpenShell library. The results obtained are real, effective, and scalable.

Nowadays, it is not possible to perform the verification of the correctness and uniqueness of the association between two different domains, cost and geometry. Typically, verification takes place exclusively within the geometric model considering only the domain of geometries. The research aims to validate data across multiple domains, in this specific case the geometric and the cost domains. These are related together and contained within the same model. This leads to numerous challenges in both cost estimation and construction phases, resulting in cost increases and potential disputes between clients and enterprises.

However, to obtain these results is required to use the code developed because current commercial applications do not support user-friendly implementations. Furthermore, this methodology cannot be applied to all IFC models, because they must be correctly modeled and contain relevant information for subsequent verification. If the geometric model lacks necessary information, such as quantities for cost estimation or crucial information for verifying costs (concrete strength class, etc.), it becomes impossible to perform proper estimation and verification procedures due to missing data.

In fact, during the experimentation phase, problems were found in the initial IFC model exported by BIM authoring software. This has led to the impossibility of carrying out a correct analysis of the model and the subsequent application of the hypothesized methodology through the developed codes. In the specific case, represented in Figure 30, it is possible to notice as erroneous modeling leads to the export of only part of the data associated with the geometric object (IfcWall with only the quantity "NetVolume" of the masonry) and not the totality of the data defined by the standard and associated with the geometric object (IfcWall with the basic quantities "Length", "Width", "Depth", "NetVolume", etc.).

These problems are crucial because if the initial data is incorrect or missing, it will be impossible to define relationships and verify their correctness solely on the data that are present in the geometric model. Therefore, data validation already from the modeling phase (using machine-readable tools, such as IDS) becomes even more important to ensure the delivery of a correct geometric pattern containing essential information for its use.

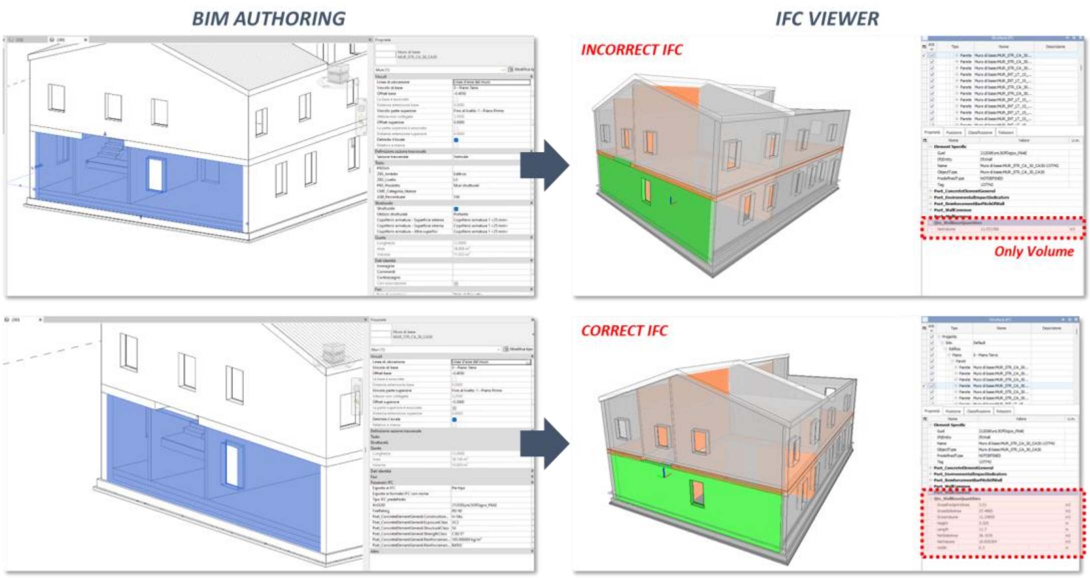


Figure 30: Comparison between correct and incorrect IFC export

7. CONCLUSION

The research highlights the crucial importance of defining a cost domain based on structured information. This allows the creation of relationships between the entities of the cost domain and that of geometries and the subsequent verification during the cost estimation phase for effective management of cost data in construction projects in the AEC sector. Initially, the study explores the definition of a new cost data architecture based on IFC classes and their relationships. This allows the definition of structured cost items in an open format, characterized by a more granular architecture than a cost item based on natural language. Additionally, it enables the creation of relationships between cost entities and geometric objects (IFC being the most widely used exchange format in the AEC sector, as noted in section 2) without the need to create attributes within the geometric objects. This new cost item structure subsequently ensures the application of a semi-automated verification method of the relationships created between two different domains, geometric and cost. This allows quick and effective control to ensure that the cost information in the updated BIM model is aligned with the previous geometric model.

The research included the development of a prototype using Python and the IfcOpenShell library, demonstrating the effective applicability of the proposed method based on the relationship between cost entities and geometric objects and the subsequent validation of the correctness of these relationships. This leads to a significant reduction in delivery time and cost estimation validation from manual to semi-automatic process. This also allows the possibility to extract, starting from the created BIM model in IFC format, key information for the entire estimation or a partial cost estimation (only the costs of the concrete masonry layers). Greater interoperability of data between the different actors in the construction process through the use of the IFC format will also be ensured.

Despite the numerous advantages this application offers, some limitations were identified in the proposed method. These include data standardization and identification of model requirements verifiable through IDS; if the model does not comply with the specified requirements, subsequent cost verification cannot be performed.

A detailed analysis of the IFC data model information is also essential for a clear understanding and proper export of the geometric object. Geometric objects cannot be identified only based on class attributes (Name, Description, TypeEnum, etc.), but it is necessary to deepen the relationships they create with other entities (*IfcMaterial*, *IfcPropertySet*, etc.). An example is distinguishing between lean concrete and slab foundations, both classified as *IfcSlab*.BASESLAB, by analyzing the LoadBearing value (TRUE or FALSE) in the Pset_SlabCommon. Moreover currently through the information hardcoded present in the standard IFC, it is not possible to define the total information that characterizes the cost items, so it will be necessary to implement the property sets associated with the entities in the standard for greater specificity.

Finally, despite the method being utilized in a specific case study focusing on specific elements such as walls, floors, foundations, and roofs, the methodology can be applied to several case studies. Ongoing and future developments will include testing in different areas of construction and large-scale models and applications to verify their reliability. In addition, ongoing research is also exploring the possibility of verifying times defined in the time schedule, as the latter must also reflect the defined unit times within the cost analysis of each cost item.

AUTHOR CONTRIBUTIONS

Conceptualization, J.C., C.M., C.Z. and A.P.; Methodology, J.C.; Code, J.C.; Validation, J.C., C.M., C.Z. and A.P.; Writing—original draft, J.C. and C.M.; Writing—review and editing, J.C., C.M., C.Z., and A.P.; Visualization, J.C., C.M., C.Z., and A.P.; Supervision, C.M., C.Z., and A.P. All authors have read and agreed to the published version of the manuscript.

DATA AVAILABILITY

The data presented in this study are openly available on GitHub at https://github.com/Cassa97/IFC-Cost-Item-Validation.git under a Creative Commons Attribution 4.0 International License.

ACKNOWLEDGEMENTS

This paper is an extended version of our previous work which was presented at 23rd International Conference on Construction Applications of Virtual Reality (CONVR 2023), Florence, Italy. The authors acknowledge the support and feedback from Prof Farzad Rahimian and Dr Vito Getuli as the Chairs of the International Scientific Committee of CONVR 2023.

REFERENCES

Adeli, H., Karim, A., & York, N. (2001). Construction Scheduling, Cost Optimization and Management. In *Construction Scheduling, Cost Optimization and Management*.

Akponeware, A. O., & Adamu, Z. A. (2017). Clash Detection or Clash Avoidance? An Investigation into Coordination Problems in 3D BIM. *Buildings 2017, Vol. 7, Page 75*, *7*(3), 75. https://doi.org/10.3390/BUILDINGS7030075

BuildingSMART. (2023). *Information Delivery Specification (IDS)*. https://technical.buildingsmart.org/standards/ifc/

BuildingSMART. (2024). *Industry Foundation Classes (IFC) - buildingSMART International*. https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/

Cassandro, J., Donatiello, M. G., Mirarchi, C., Zanchetta, C., & Pavan, A. (2023, July 10). *Reliability of IFC classes in ontology definition and cost estimation of public procurement*. https://doi.org/10.35490/EC3.2023.230

Ciribini, A., Ventura, S. M., & Bolpagni, M. (2015). Informative content validation is the key to success in a BIM-based project. *Territorio Italia*.

Codes That Support Smart Growth Development | Smart Growth | US EPA. (2024). https://19january2021snapshot.epa.gov/smartgrowth/codes-support-smart-growth-development_.html

CORENET. (2024). https://www.corenet.gov.sg/general/about-us.aspx

Dimyadi, J., & Amor, R. (2013). Automated Building Code Compliance Checking – Where is it at? *19th International CIB World Building Congress*, 172–185.

Dion Moult. (2024). *IfcTester*. https://pypi.org/project/ifctester/#description

Eastman, C. M., Jeong, Y.-S., Sacks, R., & Kaner, I. (2010). Exchange Model and Exchange Object Concepts for Implementation of National BIM Standards. *Journal of Computing in Civil Engineering*, *24*(1), 25–34. https://doi.org/10.1061/(ASCE)0887-3801(2010)24:1(25)/ASSET/7364E9D5-44CB-40F2-B694-BFFD9C38CE42/ASSETS/IMAGES/LARGE/10.JPG

Elghaish, F., Abrishami, S., Hosseini, M. R., & Abu-Samra, S. (2020). Revolutionising cost structure for integrated project delivery: a BIM-based solution. *Engineering, Construction and Architectural Management*, *28*(4), 1214–1240.

EN 17412-1. (2020). https://store.uni.com/en-17412-1-2020

Froese T, Grobler F, Ritzenthaler J, Yu K, & Akinci B. (1999). Industry foundation classes for project management - a trial implementation. *Electronic Journal of Information Technology in Construction*.

Fu, C., Aouad, G., Lee, A., Mashall-Ponting, A., & Wu, S. (2006). IFC model viewer to support nD model application. *Automation in Construction*, *15*(2), 178–185.

Fürstenberg, D., Wikström, L., Hjelseth E. & Lædre, O., (2021). Enabling automation of BIM-based cost estimation by semantic web technology. *Conference CIB W782021, 11-15 October 2021, Luxembourg*, 937-945. https://itc.scix.net/pdfs/w78-2021-paper-094.pdf

Ghannad, P., Lee, Y. C., Dimyadi, J., & Solihin, W. (2019). Automated BIM data validation integrating open-standard schema with visual programming language. *Advanced Engineering Informatics*, *40*, 14–28. https://doi.org/10.1016/j.aei.2019.01.006

Greenwood, D., Lockley, S., Malsane, S., & Matthews, J. (2010). *Automated compliance checking using building information models*. RICS. https://researchportal.northumbria.ac.uk/en/publications/automated-compliance-checking-using-building-information-models

IfcOpenShell Contributors. (2020). *IfcOpenShell-Python-Documentation*. https://ifcopenshell.github.io/docs/python/html/ifcopenshell-python.html

Ismail, A. S., Ali, K. N., Iahad, N. A., Kassem, M. A., & Al-Ashwal, N. T. (2023). BIM-Based Automated Code Compliance Checking System in Malaysian Fire Safety Regulations: A User-Friendly Approach. *Buildings 2023, Vol. 13, Page 1404*, *13*(6), 1404. https://doi.org/10.3390/BUILDINGS13061404

ISO 16739-1:2018 - Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries — Part 1: Data Schema (2018). https://www.iso.org/standard/70303.html

ISO 16739-1:2024 (2024). https://www.iso.org/standard/84123.html

ISO 19650-1. (2018). https://www.iso.org/standard/68078.html

Jiang, S., Jiang, L., Han, Y., Wu, Z., & Wang, N. (2019). OpenBIM: An Enabling Solution for Information Interoperability. *Applied Sciences (Switzerland)*, *9*(24).

Liao, L., Man, Q., Teo, E. A. L., Li, L., & Li, X. (2014). Improving construction schedule and cost information feedback in building information modelling. *Proceedings of Institution of Civil Engineers: Management, Procurement and Law*, *167*(2), 91–99.

Lu, Q., Won, J., & Cheng, J. C. P. (2016). A financial decision making framework for construction projects based on 5D Building Information Modeling (BIM). *International Journal of Project Management*, *34*(1), 3–21.

Ma, Z., Wei, Z., & Zhang, X. (2013). Semi-automatic and specification-compliant cost estimation for tendering of building projects based on IFC data of design model. *Automation in Construction*, *30*, 126–135.

Ma, Z., Zhang, X., Wu, S., Wei, Z., & Lou, Z. (2010). Framework Design for Bim-Based Construction Cost Estimating Software. *Proceedings of CIB W78: 27th International Conference*.

Olatunji, O. A., Lee, J. J. S., Chong, H. Y., & Akanmu, A. A. (2021). Building information modelling (BIM) penetration in quantity surveying (QS) practice. *Built Environment Project and Asset Management*.

Pavan, A., Giani, Matteo., & Mirarchi, Claudio. (2017). *BIM: metodi e strumenti : progettare, costruire e gestire nell'era digitale*. Tecniche nuove.

Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). BIM Handbook. In *BIM Handbook* (3rd ed). John Wiley & Sons, Inc.

Solihin, W., & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction*, *53*, 69–82. https://doi.org/10.1016/J.AUTCON.2015.03.003

Trebbi, C., Cianciulli, M., Matarazzo, F., Mirarchi, C., Cianciulli, G., & Pavan, A. (2020). Clash detection and code checking BIM platform for the Italian market. *Research for Development*, 115–125. https://doi.org/10.1007/978-3-030-33570-0_11

UpCodes | Searchable Platform for Building Codes, Assemblies, and Building Products. (2024). https://up.codes/

Wu, S., Wood, G., Ginige, K., & Jong, S. W. (2014). A technical review of BIM based cost estimating in UK quantity surveying practice, standards and tools. *J. Inf. Technol. Constr.*

Xu, S., Liu, K., & Tang, L. C. M. (2013). Cost estimation in building information models. *ICCREM 2013: Construction and Operation in the Context of Sustainability - Proceedings of the 2013 International Conference on Construction and Real Estate Management*, 555–566.

Zhiliang, M., Zhenhua, W., Wu, S., & Zhe, L. (2011). Application and extension of the IFC standard in construction cost estimating for tendering in China. *Automation in Construction*, *20*(2), 196–204.