OPENBIM-BASED GREEN PUBLIC PROCUREMENT CODE CHECKING: RULE MAPPING, INFORMATION DELIVERY SPECIFICATIONS AND GHERKIN TESTS

Carlo Zanchetta[1], Valeria Galleran[2], Thomas Krijnen[3] and Carlo Pavan[4]
[1]Università degli Studi di Padova, Italy
[2]Building Engineer, Rovigo, Italy
[3]AECGeeks, Eindhoven, The Netherlands
[4]120 grammi, Treviso, Italy

Abstract

This paper investigates openBIM-based Green Public Procurement (GPP) and Environmental Product Declaration (EPD) code checking and the suitability of Industry Foundation Classes (IFC) for supporting such checks. The coverage of Italian national requirements is assessed by concepts in the IFC schema and validation procedures such as Information Delivery Specification (IDS). The work was carried out under a research project funded by Northern Italy Venetian Regional Authority (CAM.bio project). For requirements exceeding IDS capabilities: graph-based checks, calculations, and facets outside IDS scope, a Gherkin-style validation combines human-readable scenarios with imperative code. Validation on IFC models is presented, alongside reflection and conclusions.

Introduction

The problem of implementing a digital strategy to monitor EPD and GPP processes is highly discussed both in relation to openBIM approaches (Aragon et al, 2025) standardization of rule checks (Tomczak, 2022) and alignment with Digital Product Passport (DPP) requirements. (Hamidkhaniha et al., 2025)

Several approaches have been developed to trace EPD and GPP requirements in a digital environment highlighting the necessity to adopt rule checking procedures. The majority of the approaches depends on naming standards and/or the definition of proprietary attributes. This causes the impossibility of standardizing those procedures and overpass national or local approaches. Due to the DPP directive this standardization and national codes alignment is necessary to fully support an EU policy for EPD monitoring, neutral and independent digitalization, computational code checking. (Honic et al., 2024). Based on this background, the study focused two research questions:

- RQ1. Is IFC a standard robust enough to support regulatory GPP requirements validation procedures?
- RQ2. Can IFC-based regulatory verification procedures be fully translated into standardized and vendor-neutral procedures such as IDS?

State of the art

The Industry Foundation Classes (IFC) are the predominant open standard to exchange Building Information by means of digital models (BIM). IFC is often regarded as a comprehensive and rich ontology, but in terms of producing data, the standard has limitations when highly specific information is required and proprietary authoring tools adopt specific dialects. At the same time, individual modeler preferences or the intent behind a data exchange has pronounced consequences on the way data is structured and exported. For example, Noardo et. al (2019) documents such tool limitations in the context of geospatial engineering. Gerbino et al. (2021) specifically looks at some of the difficulties in retaining material data across exports.

Limitations on information level available in IFC standard affect the code checking procedures that can be activated. There is a variety of standards and approaches to define information requirements on BIM models. Tomczak et al. (2022) identifies Spreadsheet, Product Data Templates, Data Dictionaries, IDS, mvdXML, idmXML, Level of information need, IFC Property templates and Linked Data + SHACL. These approaches follow a *declarative* approach, the requirements are stated as the intent that needs to be checked. This as opposed to an imperative approach that states the way how to perform the check. A somewhat hybrid form is Gherkin, which combines a natural language textual scenario description coupled with granular implementation steps in an imperative programming language. Gherkin for IFC has been described first in Moult and Krijnen (2020). It has since then been actively employed in the buildingSMART International (bSI) validation service[1].

When removing approaches without software support, approaches in which the semantics are not computer interpretable or that are more focused on the shape of the data, the approaches include IDS, mvdXML, SHACL and Gherkin. Jaud et al. (2025) specifically highlights the

[1] https://validate.buildingsmart.org

limitations of IDS compared to mvdXML, in particular with respect their precision: where mvdXML enables to stipulate the exact graph pattern to require, IDS is more focused on the interpretation of that pattern in end-user facing semantics and does not differentiate for example whether a property is defined on the occurrence or on the type-level. On the other hand, SHACL, described for its relevance to the AEC domain in Oraskari et al. (2021), requires moving the data to the language of the semantic web (RDF; Resource Description Format) which comes with implications in terms of increased dataset size and lacks a rich ecosystem to perform geometrical analysis (Pauwels et al. 2017). mvdXML has considerably less implementation support than IDS, higher complexity and no version-independence due to its focus on and tight coupling with IFC graph patterns.

The necessity for robust data validation is foundational in particular in the sustainability domain because of its transition toward an automated, BIM-based decision-support ecosystem, driven by increasingly stringent Green Public Procurement (GPP) requirements. Current research emphasizes the integration of Artificial Intelligence (AI) and Life Cycle Assessment (LCA) within the BIM environment to streamline the complex trade-offs required for regulatory compliance (Li et al., 2024). This paradigm shift is most critical during the Early Design Stage (EDS), where model-driven material selection dictates the building's ultimate environmental load and its alignment with mandatory sustainability criteria (Kamari et al., 2022). Such aspects state the necessity to rely on standardized datasets.

Methodology

To support the necessity to adopt standardized and openBIM code checking procedures related to EPD and GPP requirements, a study on the alignment with EPD information requirements was developed highlighting the percentage of attributes that are missing in the current IFC data model. To solve the problem of standardizing GPP/EPD information lacks in the IFC schema buildingSMART Data Dictionary (bSDD) technology was used to pivot on neutral attributes on an independent but standardized extension of the data model. The second part of the study was then dedicated to evaluate IDS robustness in supporting IFC based code checking procedures in the sustainability domain. Using standardized information requirements, an IDS-based code checking procedure was tested. Similarly to what was experimented in the first phase also in this phase the methodology was developed to assess the maturity of the standard by determining the actual percentage of code checking procedures that are supported by IDS .

An operational answer to these limits was developed and tested to demonstrate the applicability of the approach in a Common Data Environment (CDE) developed in alignment to those standards.

The analysis of the percentage of requirements and of procedures covered by bSI standards is then presented and commented on as a result of the experimentation and of the level of robustness of the schema.

The rationale behind the chosen methods is to rely on a procedure that includes at least 67% of EPD domain definitions that can be translated into IFC protocols. Furthermore, it must be demonstrated that the missing information can be integrated using only standardized protocols. Similarly, an information technology that allows for the integration of model compliance verification procedures using only open and neutral protocols is considered as promising. To demonstrate this assumption, in the third phase of the study integration scenarios, to properly support the openBIM code checking procedure, were tested.

Experimentation

First phase: specification

This phase defined the interoperability framework for linking CAM environmental requirements to IFC in a structured and verifiable way. The study focused only on mandatory requirements relevant to the design and construction phases, excluding non-binding provisions related to candidate qualification and procurement. Each criterion was systematically indexed within a spreadsheet, and its requirements were summarized highlighting qualitative and quantitative contents.

Subsequently, each item was evaluated and categorized into one of four levels of 'IFC Compatibility: fully IFC compliant, partially IFC compliant, potentially IFC compliant (partially or fully but characterized by high implementation complexity or requirements related to subjectivity of modeling), and non-IFC compliant (qualitative requirements or elements unrelated to information modeling). This assessment was based on three key factors: the formalization of CAM specifications into information requirements covered by IFC; the effective optimization of data entry and validation processes within current BIM applications; the operational feasibility for the stakeholders involved.

To ensure the practical feasibility and reliability of the proposed framework, the study focused exclusively on '*Fully*' and '*Partially*' compliant criteria. Consequently, both 'Non-IFC Compliant' items and those labeled as '*Potentially Compliant*'—due to their excessive technical barriers and operational complexity—were excluded from the research scope.

Development of a specific IFC Mapping and bSDD

The next step was to determine how the selected criteria could be mapped within the IFC schema. IFC4 ADD2 TC1 (version 4.0.2.1) was adopted as the reference schema, balancing standard maturity with compatibility with current industry tools. The mapping analysis process also aimed to:

- identify specific criteria requiring the development of new Classes or Property Sets (Psets) to address information which were not currently included in the IFC schema;
- identify possible difficulties in data writing procedures for operators with BIM Authoring applications (data assignments to abstract classes such

as groups or systems, document associations to elements, data assignments to material layers).

The analysis revealed that, of the total IFC-compliant criteria developed, 8% of the criteria can be fully mapped according to the schema, while the remaining 92% can be mapped according to other openBIM standards, by defining custom Psets and by developing a bSDD classification dictionary.
It also emerged that more than 50% of the criteria were affected by potential operational issues such as assignment to IfcProject, -Site or -Building which may not exist as such in the native environment, usage of document associations and classifications on objects and materials for which tools have limited support, or decompositions such as material layers.

A significant challenge arises from the diverse nature of CAM criteria, which often stipulate specific requirements for construction materials, such as defined recycled content. These requirements encompass a wide array of numerical, textual, and boolean parameters. Furthermore, certain criteria demand properties not natively supported by the IFC schema, specialized quantity take-offs from filtered object selections, or the inclusion of technical documentation linked directly to model entities.

A technical mapping sheet was developed to define, for each criterion, the relevant IFC entities, materials, properties, and document associations. To support consistent data population and semantic interoperability, CAM information was also structured within a dedicated "CAM Classification" bSDD dictionary covering elements, materials, and properties for the purpose of consistency and adoption in tools that support bSDD.

Analysis of validation procedures of each criterion

Finally, the specification phase defined validation rules and tolls needed for each criterion. Validation was initially formalized using the Information Delivery Specification (IDS) format, but the analysis showed that several criteria exceeded current IDS capabilities:

Validation of document association: most of the criteria required a document association validation related to objects or material layers. However, IDS does not support the verification of external files linked via *IfcDocumentReference*;

- Disjunctive logic (OR conjunction);
- Conditional logic (IF-THEN) validation: where the validation of an object's property depends on conditions defined in a (separated or related) entity;
- Element-to-Material recursion: while IDS can verify that an element associated with "Material X" possesses specific properties, it is impossible to shift focus to the material and apply constraints to the material entity itself;
- Data aggregation and computational criteria: performing project-wide calculations, such as verifying that the cumulative weight of all recyclable materials exceeds 70% of the total project mass.

To overcome these limitations, the research implemented an IDS multi-applicability approach and custom scripts to parse the IFC model, extracting data from both entity and material levels to perform the required validation. The analysis concluded by assigning each criterion to one of three automated validation scenarios tailored to the specific complexity of the data to be verified: IDS-exclusive, script-exclusive, blended validation.

Second phase: validation

To develop the GPP-CAM validation phase, multiple IFC model prototypes were created within an authoring tool according to defined rules and mapping in technical sheet to serve as a baseline for subsequent operations as: test data writing and checking limitations, identify formal control strategies according to the effectiveness of IDS rules, define an overall validation workflow.

An operational guide was then developed to support practitioners and streamline the workflow by sharing a general know-how of IDS rules structure and convey multi-step procedure of validation, including best practices for correct model setup and data population to ensure the accurate generation of the CAM information database according to the CAM Classification bsDD.

Operational and software limitations

A preliminary audit of the informational content of the test models revealed significant deficiencies in data quality and completeness.

Recognizing that many errors stem from the inherent limitations of commercial BIM authoring tools rather than user oversight, the methodology incorporated a 'post-procedural editing' phase (authoring&integration). This ensures that models failing validation due to software constraints can be automatically refined and corrected before the final assessment. Consequently, it became necessary to develop dedicated implementation tools within both desktop applications and CDE.

Recurring limitations in commercial BIM authoring softwares that hinder the direct implementation of CAM requirements included:

- *material classification*: difficulty in assigning standardized classification codes directly to material layers;
- *parameter duplication*: issues with redundant project parameters during the IFC merging phase;
- *entity and group classification*: constraints in mapping classifications to complex element groups;
- *document association*: lack of streamlined workflows for linking external documentation to model objects;
- *generic connectivity*: limitations in creating generic connection relationships between elements;
- *schema conversion*: difficulties in converting IfcGroup entities into specific IFC schema subclasses.

Formal control strategies: Editing for validation

To address these software limitations, the project initially tested advanced IFC model manipulation within the open-source Blender environment. By leveraging the integration of the IfcOpenShell library and the graphical interface provided by Bonsai add-on, the following operations were successfully implemented:

- *document association*: deployment of IfcDocumentReference entities and their corresponding IfcRelAssociatesDocument relationships to ensure formal linkages between external sources (e.g., certifications, technical data sheets) and individual model entities.
- *database structuring*: refinement of the model's internal data structure to ensure full compliance with the IFC schema, thereby correcting inconsistencies generated during export from proprietary software;
- *classification mapping*: assignment of one or more classification codes to model elements using IfcClassificationReference entities and IfcRelAssociatesClassification relationships, facilitating interoperability with external classification systems such as the bSDD.

However, the operational complexity of manual editing highlighted a need for more efficient, less error-prone methods to support large-scale implementation. Consequently, the project shifted toward automated IFC-editing solutions through the development of dedicated 'IfcPatch' recipes in Python. These scripts allow for the automated post-processing and refinement, ensuring a standardized data structure regardless of the constraints imposed by the original authoring tool.

IDS Rule Architecture and Hierarchical Validation

Given the limitations of the IDS format, IDS rules were organized according to a predefined schema where each CAM criterion is decomposed into one or more conditions applicable to the IFC data. For every criterion verifiable via IDS, a multi-specification rule applicability approach was developed to determine compliance. These specifications are structured into three hierarchical levels to enable a step-by-step verification process:

- Level 0 - Applicability: These specifications define the relevance of a CAM rule to the analyzed model. If these conditions are not met, the criterion is either inapplicable to the project or essential preliminary information is absent.
- Level 1 - Entity Classification: These rules ensure that the targeted IFC entities are correctly classified according to the bSDD dictionary. Failure at this level indicates that entities are misidentified, thereby precluding further requirement validation.
- Level 2 - CAM Criterion Validation: These specifications verify whether the IFC model fulfills the actual environmental requirements of the CAM criterion. Non-compliance at this stage necessitates data integration or design adjustments.

This tiered structure facilitates a granular diagnostic process, identifying whether a failure stems from a lack of relevance (Level 0), a classification error (Level 1), or a direct violation of criteria requirements (Level 2).

Orchestrating IDS for conditional logic

The Multiple Applicability Approach was also implemented to overcome the structural constraints of the IDS standard, specifically the lack of support for dependencies or conditional IF-THEN logic. Instead of a single monolithic check, CAM requirements are decomposed into sequential, interdependent IDS files as:

- Triggering Specification: The first IDS file verifies the presence and attributes of the "source" element (e.g., IF an IfcWall is classified as 'LoadBearing').
- Targeted Specification: If the first check is satisfied, a second specification validates the required conditions on the related element (e.g., THEN the associated IfcMaterial must have a specific recycled content property).

Under this approach, CAM Compliance of IF-THEN logic is formally achieved through two valid states:

- Full Compliance: Both specifications are successfully validated, confirming the condition was triggered and the requirement met.
- Non-Applicability: The first specification is not satisfied, indicating the triggering condition is absent (e.g., no LoadBearing walls exist), and thus the requirement is not invoked.

Table 1: Synthesis of validation outcomes

Result level	Status	Meaning
Level 0 Fail	Not applicable	The rule does not apply to the model
Level 1 Fail	Classification Error	Entities are found but not correctly mapped to bsDD
Level 2 Fail	Non-Compliant	Data is present but does not meet CAM thresholds
All Level Pass	Compliant	The model satisfies the CAM criterion

Definition of an Automated Validation Framework

Finally, each criterion was assigned to one of three operational validation scenarios based on its technical complexity:

- Scenario 1: Hybrid Validation (IfcPatch + IDS + Script): The most robust approach. It combines model normalization, standard attribute checking, and complex computational logic for maximum accuracy and traceability.
- Scenario 2: Standard-Centric (IfcPatch + IDS): Suitable for criteria fully expressible through the IDS schema. It ensures high compatibility with international tools but is limited to non-computational requirements.
- Scenario 3: Script-Driven (IfcPatch + Script): Utilized for performance-based criteria that fall outside the IDS scope. It relies on direct analysis of the IFC

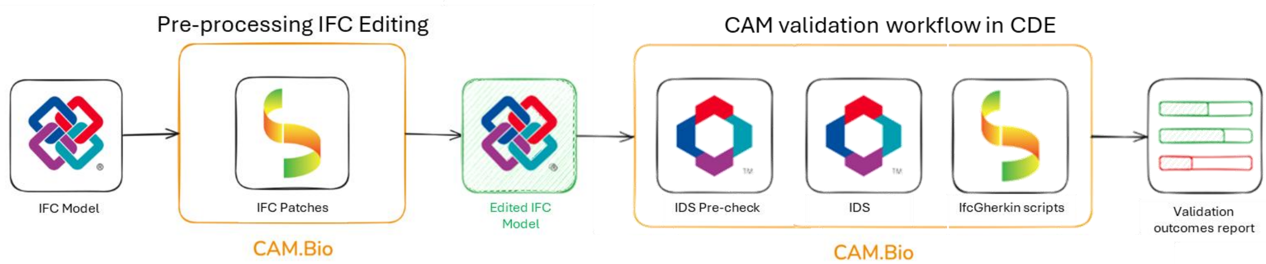

Figure 1: Schematic overview of the contributions of the CAM.bio project wrt. the validation workflow

structure to validate complex environmental constraints and relationships.

This analysis enabled the definition of a comprehensive validation framework that encompasses all potential scenarios, allowing it to adapt seamlessly to each specific criteria by integrating standard protocols with custom computational logic.

- **Phase 1: Pre-processing (IfcPatch):** Uses the IfcOpenShell library to normalize data, correct structural errors, and inject missing relationships or attributes into the IFC model.
- **Phase 2: IDS Pre-check:** A custom module verifies the presence of mandatory information required for formal validation, preventing execution errors.
- **Phase 3: Standard IDS Validation:** Executes .ids files to verify criteria that can be formally expressed via the standard buildingSMART XML schema.
- **Phase 4: Advanced Scripting (IfcGherkin/IfcOpenShell):** Python scripts perform complex verifications unreachable by IDS, such as Boolean logic on materials, cross-entity spatial checks, and geometric quantity aggregations.

The defined four-phase validation process can be made scalable and seamlessly integrated into the CDE platform by encapsulating the entire validation workflow (IfcPatch, IDS pre-check, IDS, and custom scripts) within a Docker microservice exposed via RESTful APIs.

openCDE Platform: Implementation and Visualization

To ensure CAM requirements are manageable by non-technical stakeholders and to demonstrate the feasibility of the proposed framework, the research evolved a CDE prototype. The Xeokit viewer was customized to expose CAM entities often hidden in traditional BIM viewers (e.g. TypeEnums). This included a reorganized tree hierarchy for IfcSpatialZone and IfcBuildingSystem, and direct visibility of material-level properties and external documentation (IfcDocumentReference) reflecting CAM Classification bSDD. The platform includes an IDS Authoring Environment for rule creation and a Validator UI for real-time compliance feedback. Third phase: integration and authoring

Given the limitations and dialects of IFC authoring tools, a process is in place to unify and harmonize the data ingested into the system. In the platform (named CAM.bio – the italian word for "change") this is built around the idea of IfcPatch[2], which is an existing module in the IfcOpenShell family of tools that consists of 'recipes' to fix or alter IFC content and write out a revised version. These recipes are implemented in python using imperative code. In our particular case we built a minimal graph rewriting DSL to declaratively encode the intended transformations. In particular it was found that certain tools were unable to provide classification references to material definitions. As a work-around the modeller was instructed to write these as a material property using particular conventions that would be picked up by the IfcPatch recipe. See Figure 2 for part of the implementation, Figure 3 and Figure 4 for the before and after as a minimal graph rendering.

Model validation

As introduced earlier in the text there has been extensive research on validating information requirements. Many of these can also be used to validate design requirements such as CAM, because they use a similar vocabulary. For this project, several have been considered.

IDS is used in the project, it is limited in scope and expressiveness, but well-tested and widely implemented. An additional approach needs to be selected that is richer

```
select()\
  .where(type='IfcMaterialProperties')\
  .where(Name='ClassificazioneCAM')\
  .select('Material', 'Material').up()\
  .select('Properties')\
  .where(length=1)\
  .select(0)\
  .where(Name='ClassificationCode')\
  .select('NominalValue')\
  .select(0, 'Value')\
  .remove()\
  .construct({'IfcExternalReferenceRelationship': {
    'RelatedResourceObjects': ['{Material}'],
    'RelatingReference': {
      'IfcClassificationReference': {
        'Identification': '{Value}',
        'Description': lambda di:
            bsdd.code_to_description[di['Value']],
        'ReferencedSource': {
          'IfcClassification': {
            'Name': 'CAMClassification'
          }
        }
      }
    }
}})
```

Figure 2: Code produced to rewrite material classifications

[2] https://docs.ifcopenshell.org/ifcpatch.html

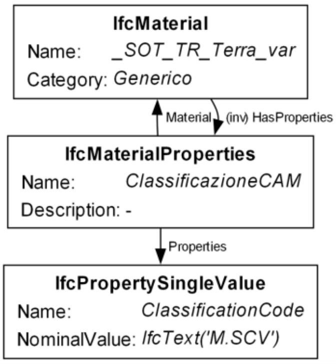

Figure 3: 'Before' situation with classification as property

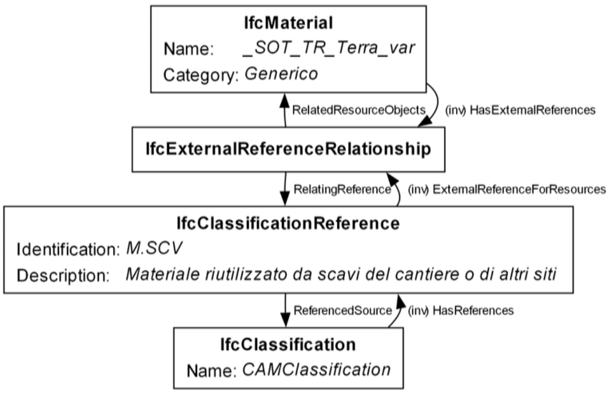

Figure 4: 'After' situation when rewritten

in expressiveness to support implementing the required constraints.

That is also not surprising given that IDS is intended to control the availability of information on top of the modelling mechanisms that IFC and standard model view definitions already propose. mvdXML has the possibility to precisely stipulate graph patterns, but is still lacking many of the constructs needed for the ambitions set out in this paper such as aggregate functions to sum areas together and the ability to assess and compare multiple focus nodes in one scenario.

Naturally, the option exists to implement rules in proprietary software, but this lacks the portability for moving such checks to a web-based platform and is not vendor neutral. Another option is to not use a pre-existing approach at all, and to simply use an imperative programming language to implement the required functionality, which is flexible but comes at a prohibitive cost in terms of shared understanding between domain stakeholders.

The approach selected in the end is Gherkin, which consists of a high level human readable set of scenarios (an example for this project is given in Figure 5) that describe test logic for shared understanding between stakeholders, which are paired with an implementation in (in this case) the Python programming language for the actual execution. As such it can be seen as a middle ground between expressiveness, portability and ease of understanding.

In Gherkin scenarios are broken up into steps, these steps are matched to parameterized implementation functions, as such there obvious advantages: people do not need to understand program code, but can follow the high-level specification; as such there is a shared understanding of the requirements between software engineers as well as other stakeholders; the textual breakdown into statements offers modular reuse of functionality. In order to best understand the technology it's best to look at the code repository that contains the body of rules developed for the validation service[3]. In Figure 6 one sees a table of the functional steps employed in the validation service. This table highlights the relative strength of the Gherkin approach for IFC model verification: Gherkin allows a similar structure of graph pattern building using entity and attribute selection (not too dissimilar from EntityRule and

```
@E00160
Feature: Rule 2.3.3.a - Riduzione effetto isola calore estiva e inquinamento atmosferico

  Scenario: New construction projects and Urban renovation projects must have: green areas
              >= 60% permeable land surface

    Given an .IfcGeographicElement/TERRAIN. or .IfcPavement. or .IfcSlab/PAVING. Or …

    # Filter based on classification
    Given a relationship .IfcRelAssociatesClassification. to .the above. from …
    Given .Identification. ^is^ 'E.TER' or 'E.SSN' or 'E.SPA' or 'E.SSI' or 'E.SCA' …
    # Move back up to the instances
    Given the instances '3' steps up

    Given the property value for 'NetArea'
    Given the sum of all values at depth 0 [stored as 'GreenAreas']

    Given The result for 'SuperficiePermeabile' from 'Rule_232' [stored as
                    'SuperficiePermeabile']

```

Figure 5: Gherkin scenario text implemented for the CAM.bio project for model validation. Given statements correspond to the selection of data from the model (applicability); Then statements perform validation (requirement)

[3] https://github.com/buildingsmart/ifc-gherkin-rules

Predicate / operation	Usage
`An .{entity_opt_stmt}.`	233
`.{attribute}. ^{comparison_operator:equal_or_not_equal}^ {value}`	192
`Its attribute .{attribute}.`	166
`A {file_or_model} with {field} '{values}'`	80
`There must be {constraint} {num:d} instance(s) of .{entity}. …`	58
`A relationship .{relationship}. {dir1:from_to} .{entity}. {dir2:from_to} …`	27
…	…
`It must have no arc segments that use colinear points after taking …`	1
`All edges must form a single connected component`	
`The boundaries of the face must conform to the implicit plane fitted through …`	

Figure 6: Usage count of Gherkin implementation function highlighting very common mvdXML-like traversal operations and specialistic single-use steps for specific imperative validations

AttributeRule in MvdXML) and at the same time using highly specialistic predicates that also involve the wide availability of python modules to build rich behavior for the breadth of disciplines represented in the IFC schema and complexity of downstream verification use-cases.

An example of such a scenario that has been built for the CAM.bio project is depicted in Figure 5. The first lines (line 0-5) are descriptive. In line 8 an initial selection of entity instances is made from the model. Their classification is looked up and filtered. For the elements with matching classifications we sum their areas. This is then compared to the 'SuperficiePermeabile' which is another rule implemented in similar fashion.

Discussion and results

Regarding BIM-compliance analysis, the structural limitations identified within the schema are as follows:

- 44% of the criteria cannot be fully implemented due to the absence of specific entities for environmental and infrastructural contexts (e.g., river ecosystems, groundwater). These entities were introduced in the IFC 4.3 version, rendering the current version insufficient for direct ontological mapping. Future research is to be directed at this.
- The current breakdown of IfcMaterial does not allow for an analytical definition of the internal composition of mixtures or composite materials, complicating the verification of chemical-physical performance criteria.

Regarding requirements that are potentially or partially IFC-compliant but were excluded due to implementation difficulties, the following are highlighted:

Regulatory requirements involving subjective or qualitative evaluations (e.g., prevention, mitigation, analysis of existing greenery) that cannot be translated into standardized parametric properties.

- Compliance with external regulations outside the scope of this study; simplifications were adopted, transitioning to manual verification.
- Comparisons between different temporal phases, which are complex to develop and risk imposing workflows that are disproportionately burdensome for the operator relative to the benefits of the verification. This type of analysis could be further explored in future research.
- Criteria requiring the identification of "technical strategies" that conceptually lack a unique digital identity. In these cases, simplification was preferred to avoid the definition of overly extensive custom IfcPropertySet that would be difficult to standardize.

The analysis revealed that, of the total criteria defined by the CAM regulation, more than 76% are IFC compliant. Considering the research screening for determining the project's scope, based on the exclusion of non-mandatory and difficult-to-implement criteria, it emerges that IFC-compliant CAM criteria represent a total of 56% of the total CAM regulation criteria (46% fully IFC Compliant, 10% Partially IFC Compliant, 21% Potentially IFC Compliant, and 23% Non-IFC Compliant.

This study excluded several criteria from its scope, specifically those pertaining to urban and territorial planning. Such criteria necessitate the calculation of spatial proximities between the project site and local public services, a requirement that can be effectively addressed through enhanced interoperability between IFC and GIS models.

Concerning the IFC-mapping analysis, the study identified that out of the 816 entities comprising the IFC4x2 schema, a subset of 39 entities and their respective TypeEnum is sufficient to fulfill the CAM regulatory requirements. These requirements primarily map to the following categories: IfcMaterial, IfcDocumentReference, IfcSpatialElement, IfcSystem, and IfcBuildingElement sub-classes, including their relative predefined types.

The primary challenge encountered during this mapping phase was the necessity for granular discretization of entities based on their ontological and functional characteristics (e.g., IfcMaterial naming conventions, IfcSystem, and IfcBuildingElement) or according to specific document contents (e.g., IfcDocumentReference). This granularity is essential for executing automated filtering operations and rigorous data validation. Additionally, the definition of custom Property Sets proved critical, especially for capturing data related to recycled, recovered, and by-product content within products and materials.

In this context, the development of a dedicated bSDD was fundamental. To define the requirements dictated by the CAM criteria, the dictionary incorporates 144 class codes, categorized as follows (25 codes for the discretization of document types within the IfcDocumentReference entity; 43 for the characterization of building element features; 8 mapping to various IfcSpatialElement sub-classes; 54 for the classification of diverse IfcMaterial types; 14 defining IfcSystem sub-classes)

The project ultimately identified 104 distinct properties. While 13 of these attributes are natively available within the IFC schema, the remaining 91 were specifically introduced via the cam-bSDD.

Concerning Validation, the analysis of the 38 CAM criteria revealed a high dependency on model normalization and multi-layered validation framework. Specifically, 36 criteria (95%) required corrective support via IfcPatch to ensure data consistency. More than 87% of CAM criteria needed Gherkin as a fallback due to technical requirements and entity comparisons not supported by IDS format only.

Table 2: CAM validation procedure breakdown

Validation Component	Criteria Count (out of 38)	% of Total
IfcPatch Support (Pre-processing)	36	95%
IDS Pre-check	21	55%
IDS-only Validation	5	13%
Script-only Validation	14	37%
Hybrid Validation (IDS + Script)	19	50%

Conclusions

CAM.bio contents were fully aligned with the Italian regulatory framework in force at the time of completion (D.M. 23/06/2022 n° 256). As is often the case in building construction field, new legislation updating CAM criteria has taken effect from February 2026.

In this context, the project serves as a baseline reference to support Italian current implementation and the transition to the updated regulatory framework, and provides an example of structured bSDD dictionary development and as a reference for international researchers.

The study establishes limits of the current version of IFC and IDS in supporting computational and openBIM based code checking procedures based on the information requirements needed by the national implementation of EPD directive.

Acknowledgments

The study was developed in relation to a research project funded by Veneto regional authority (northern Italy area). Further details can be found at the following links:
https://cam-bio.it
https://identifier.buildingsmart.org/uri/cadlinesw/cambio/1.0

References

Aragón, A., Spudys, P., Pupeikis, D., Nieto, Ó., & Garcia Alberti, M. (2025). Bridging interoperability gaps between LCA and BIM: Analysis of limitations for the integration of EPD data in IFC. Buildings, 15(15), 2760.

Gerbino, S., Cieri, L., Rainieri, C., & Fabbrocino, G. (2021). On BIM Interoperability via the IFC Standard: An Assessment from the Structural Engineering and Design Viewpoint. Applied Sciences, 11(23), 11430. https://doi.org/10.3390/app112311430

Hamidkhaniha, S., Martinez, A. G., & Sedaghat, M. (2025, December). Integrating Digital Product Passports and BIM for Circular Construction: A Review of Challenges and Opportunities for Data-Driven Material Reuse and Sustainability Assessment. In IOP Conference Series: Earth and Environmental Science (Vol. 1568, No. 1, p. 012026). IOP Publishing.

Honic, M., Magalhães, P. M., & Van den Bosch, P. (2024). From data templates to material passports and digital product passports. A circular built environment in the digital age, 79.

Jaud, S., Marie, S., & Pinzenoehler, A. (2025). The Quest for the OpenBIM Exchange Requirements Checking Language: Comparing mvdXML, IDS and Gherkin. In EC3 Conference 2025 (Vol. 6, pp. 0-0). European Council on Computing in Construction.

Kamari, Aliakbar & Kotula, Bartlomiej & Schultz, Carl. (2022). A BIM-based LCA tool for sustainable building design during the early design stage. Smart and Sustainable Built Environment. 10.1108/SASBE-09-2021-0157.

Moult, Dion, and Thomas Krijnen. "Compliance checking on building models with the Gherkin language and Continuous Integration." EG-ICE 2020 Workshop on Intelligent Computing in Engineering, Proceedings. 2020.

Noardo, F., Arroyo Ohori, K., Biljecki, F., Krijnen, T., Ellul, C., Harrie, L., & Stoter, J. (2019, June). GeoBIM benchmark 2019: design and initial results. In International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives (No. 2/W13, pp. 1339-1346). ISPRS.

Li, J., Liu, Z., Han, G., Demian, P., & Osmani, M. (2024). The Relationship Between Artificial Intelligence (AI) and Building Information Modeling (BIM) Technologies for Sustainable Building in the Context of Smart Cities. Sustainability, 16(24), 10848. https://doi.org/10.3390/su162410848

Oraskari, J., Senthilvel, M., & Beetz, J. (2021, October). SHACL is for LBD what mvdXML is for IFC. In Proc. of the Conference CIB W (Vol. 78, pp. 11-15).

Pauwels, P., Krijnen, T., Terkaj, W., & Beetz, J. (2017). Enhancing the ifcOWL ontology with an alternative representation for geometric data. Automation in Construction, 80, 77-94.

Tomczak, A., v Berlo, L., Krijnen, T., Borrmann, A., & Bolpagni, M. (2022, November). A review of methods to specify information requirements in digital construction projects. In IOP Conference Series: Earth and Environmental Science (Vol. 1101, No. 9, p. 092024). IOP Publishing.