buildings

MDPI

Article

Coverage-Accounted, Provenance-Aware Validation of IFC Building Models for Public Safety Management

Jaewook Lee [1], Chorong Park [2], Wonho Cho [1] and Hyun Sang Choi [2,*]

[1] Department of Architecture, Gachon University, 1342 Seongnam-daero, Sujeong-gu, Seongnam 13120, Republic of Korea; juklee531@gmail.com (J.L.); blackjoker8822@gmail.com (W.C.)
[2] Korea Institute of Civil Engineering and Building Technology (KICT), 283 Goyang-daero, Ilsanseo-gu, Goyang 10223, Republic of Korea; chorong@kict.re.kr
* Correspondence: hyunsang@kict.re.kr

Abstract

Industry Foundation Classes (IFC) files are the main deliverable for the digital safety management of public buildings, but a file that passes a schema check is not necessarily usable. Most checking tools reduce model quality to a single pass/fail result, blame the model for rules that do not apply or are themselves wrong, and drop the objects they could not process. We present a validation framework that keeps these questions distinct: it checks each element separately against the IFC4 schema and the project's information requirements; reports one of seven verdict states that distinguishes model defects from inapplicable, unspecified, or defective requirements; accounts for every element so that none disappears from the audit; and labels all simplified geometry as authored or derived. Applied to eight production models from two authoring pipelines (5994 elements; 88,499 element–rule pairs), the verdicts were reproducible, agreed with an independent oracle on all 39,334 pairs it could decide, and detected eight of ten injected defect types in every case. Separating the two checks reclassified 274 valid elements of one model from violations to rule gaps, and every model from one pipeline arrived without authored axes. The framework supports procurement-side quality assurance and data-readiness screening for aging public buildings.

Keywords: building information modeling (BIM); industry foundation classes (IFC); automated compliance checking; information requirements; information delivery specification (IDS); data provenance; coverage accounting

Check for updates

Academic Editor: Rafiq Muhammad Choudhry

Received: 25 September 2026
Revised: 13 October 2026
Accepted: 22 October 2026
Published: 26 October 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

In many countries, public building stocks are aging faster than the institutional capacity to inspect them. In Korea, buildings older than 30 years already account for a substantial share of the national stock, and municipal governments carry the statutory duty of monitoring their safety with limited engineering staff [1]. Scan-to-BIM pipelines are beginning to be used for structural performance assessment [2] and image-based defect inspection [3] on exactly this stock of deteriorated buildings. Therefore, digital safety management initiatives aim to replace document-based reporting with building information models that software can query, check, and reuse, and linking risk-management information to the IFC building model is being explored for the same purpose [4]. Within openBIM, the Industry Foundation Classes (IFC, ISO 16739) [5] serve as a vendor-neutral format in which such models are submitted and exchanged across proprietary authoring tools [6]. The premise

of any automated model-based safety workflow is that a submitted IFC file actually carries the identities, properties, and geometries required by downstream tasks.

That premise routinely fails—and it fails in ways that a single conform/does-not-conform verdict cannot express. Automated checking of a submitted model must answer at least three different questions: whether a value is valid with respect to the IFC schema; whether it satisfies the information requirements of a particular project or client—in this study, version 8.3 of the Korean sBIM information-requirement system [7]; and whether the model is ready to support downstream analyses such as structural screening. These questions have different authorities and remedies. A schema violation is the exporter's defect, an unmet project requirement may be the modeler's, and an unanswerable analysis question may be nobody's—the requirement may not apply to that element, may never have been specified for that value, or may itself be written incorrectly in the requirement table.

Conventional rule-checking pipelines conflate these layers in four ways: first, schema validity and project conformity are merged, so a value that is perfectly valid in IFC is flagged as an error merely because a project list has no entry for it; conversely, defects of the requirement specification itself are silently charged to the model author. Second, binary verdicts cannot represent non-applicability or rule gaps: demanding a reinforcement property of a steel beam or scoring a placeholder value such as 999 as "present," produces compliance rates that are precise but meaningless. Third, objects whose geometry fails to generate or whose representations are unsupported tend to disappear from summary statistics altogether—a silent omission that overstates the coverage exactly where the model is most problematic. Fourth, when the authoring tool does not export an axis representation, lightweight geometry is derived instead—for example, a member centerline inferred from a body extrusion—and the derived element is usually indistinguishable from authored data downstream, so the confidence of any subsequent interpretation cannot be traced.

These failures are not hypothetical. In the eight production models studied here, every model delivered through one of the two authoring pipelines—architecture, structure, and fire protection alike—arrived without a single specific entity classification. Placeholder loads of 999 appeared in dozens of spaces, and, as we show, an entire curtain-wall class was excluded from judgment by a checker that coupled verdict coverage with geometry generation. Detecting this class of problems requires an accounting discipline that refuses to allow any object to disappear from the audit, together with verdicts that separate a model's faults from the checking system's own gaps.

We argue that the trustworthiness of automated IFC validation is established not by maximizing the share of objects that receive a verdict but by explicitly recording four separations: defects of the model versus defects or gaps of the rule system, what applies versus what does not, geometry that is authored versus derived, and objects that were processed versus those that were not. This study operationalized these separations in a validation framework with four components:

1. A layered validation architecture that evaluates IFC schema validity and sBIM v8.3 information requirements as two separate axes;
2. A seven-state verdict model (OK, EMPTY, MISSING, SUSPECT, SBIM_UNSET, N/A, SPEC_DEFECT) that distinguishes genuine model defects from non-applicable requirements, project-rule gaps, and defects of the requirement table itself;
3. Closed-set coverage accounting, which requires the sum of processed, failed, unsupported, and excluded objects to equal the eligible population before any result may be reported as complete;
4. Native-first, provenance-aware geometric abstraction, in which every point–line–surface element carries an explicit origin label that propagates into a downstream data-readiness assessment.

The framework is implemented as a local web application on an IfcOpenShell-based Python server [8]. Verdicts are rendered as a color overlay with per-element evidence tracing. Each judgment is linked back to the requirement sheet, row, and raw model value that produced it, and the IFC file itself is never modified. Deflection and equivalent-static seismic modules are included but deliberately reported as data-readiness demonstrations, with boundary conditions, member-level loads, material strengths, and seismic parameters absent from the submitted models. Their role is to diagnose which information a structural screening would additionally require, not to assess code compliance. A formal structural safety evaluation was outside the scope of this study.

Four research questions guided this evaluation. RQ1: To what extent does separating schema validity from project requirements, together with multi-state verdicts, reduce misclassification relative to a naive application of rules? RQ2: With what accuracy can a native-first method recover member centerlines when the authored axes are absent? RQ3: How do model gaps, rule gaps, specification defects, and geometric provenance affect the readiness for downstream analysis? RQ4: What processing coverage and computational performance does the system achieve across the IFC models of different sizes and origins? These questions were answered using an audit-first experimental design on eight production IFC models: a metric-reproducibility audit, verdict conformance against a deterministic oracle grounded in the IFC4 schema and requirement tables, defect injection with constructed ground truth, a native-axis holdout test, and cold/warm performance measurement. The remainder of this paper reviews the related work (Section 2), describes the framework and experimental design (Section 3), reports the results (Section 4), discusses the implications and limitations (Section 5), and concludes the paper (Section 6).

2. Related Work

2.1. IFC Validation and Automated Rule Checking

The automated checking of building models has been studied for over three decades, although its adoption in practice still lags behind its promise [9]. Eastman et al. organized the field into four stages—rule interpretation, model preparation, rule execution, and reporting—and observed that most systems hardened normative text into black-box code [10]. Subsequent work formalized what rules may legitimately demand of a model: Solihin and Eastman classified rules according to the complexity of the model information they required [11], and the RASE methodology translated normative clauses into machine-interpretable requirement units [12]. Visual programming languages render checking logic inspectable rather than hard-coded [13], whereas natural language processing automates the extraction of rules from the regulatory texts [14]. Most recently, large language models have been applied to both interpret regulations and drive BIM-based compliance checks directly [15,16], and rule execution has been extended further to include checks of models against ontology-encoded quality standards [17] and the combination of semantic-web reasoning with IFC data for automated compliance [18].

Across this line of work, attention has been concentrated on the execution stage—how to represent and fire rules—while the reporting stage has remained comparatively primitive: an element either passes or fails, and objects that the pipeline cannot process rarely appear in the summary. The present study is complementary to rule-execution research; it takes the verdict itself and the accounting of what was and was not checked as its focus.

2.2. Project-Specific Information Requirements

Standards have long distinguished between what the IFC schema permits and what a particular exchange requires. ISO 19650 [19] frames information requirements organizationally; the Information Delivery Manual specifies exchange processes [20]; and Model

View Definitions bind subsets of the schema to exchange scenarios, with Venugopal et al. providing their formal semantics [21] and later work formalizing the logic for validating exchanged models against them [22]. The Information Delivery Specification (IDS), approved as a buildingSMART Final Standard in 2024, now standardizes machine-readable project-specific requirements and automated checking [23]. Recent studies extend IDS toward digital building permits and, notably, generate IDS files from the tabular requirement formats practitioners maintain [24,25]—the same starting point as the Korean sBIM information-requirement system used here [7], in which requirements are authored as spreadsheet tables and mechanically converted into checking rules. Natural language processing has likewise been applied to turn unstructured design briefs into machine-readable parameters [26], a reminder that requirement capture is itself an automation problem. Comparable openBIM efforts now define IDS-based minimum data requirements for specific downstream uses such as operational-carbon assessment [27].

The two-axis separation proposed in this study is conceptually aligned with the IDS; both distinguish schema validity from project conformity. It extends the IDS result model in two directions that its pass/fail vocabulary does not cover: (a) verdict states for requirement-side conditions/rules that the project has not specified (SBIM_UNSET) or that are defective in the specification itself (SPEC_DEFECT); and (b) mandatory coverage accounting for the checked population. Appendix B demonstrates on an excerpt how sBIM v8.3 rules are re-expressed as IDS 1.0 specifications and which of the seven states survive the translation; a full conversion would make both extensions applicable beyond the Korean context.

2.3. BIM-to-Analysis Transformation and Lightweight Abstraction

A separate line of research has focused on transforming physical BIM into an analysis-ready form. Ramaji and Memari systematized the interpretation of engineering analysis information from IFC models [28]. Later work generated finite-element models from BIM with ontology support and, conversely, wrote structural analysis results back into IFC for code-compliance use [29,30]. Lightweighting research simplifies IFC geometry for web-scale visualization [31], IFC-centric pipelines now drive performance-based evacuation evaluation directly from the exchanged model [32], and automated architectural-to-structural model generation now derives analysis models directly from IFC [33], although interoperability assessments show how much still depends on the completeness of the exchanged model [34], and recent reviews question how ready IFC data actually are for such downstream AI and analysis tasks [35]. These pipelines share the assumption that what they need—axes, sections, and connectivity—is present and trustworthy; otherwise, elements are silently skipped or approximated. We treat this assumption as the object of validation: geometry is abstracted native-first, every derived element is labeled as such, and elements that cannot be abstracted are accounted for rather than dropped.

2.4. Provenance, Semantic Enrichment, and Trustworthy Automation

In general data management practice, recording where a value comes from is a solved design problem; the W3C PROV data model standardizes lineage across entities, activities, and agents [36], and, as digital twins come to integrate heterogeneous and continuously updated data [37], provenance mechanisms are beginning to appear in their construction [38]. Semantic-enrichment research is relevant in a complementary way: Belsky et al. established rule-based inference of implicit semantics from model geometry and topology [39], Bloch and Sacks showed that missing semantics can be inferred from geometry with useful accuracy [40], machine-learning methods have since detected misclassified IFC elements [41] and recovered missing BIM semantics from imagery [42], and the ifcOWL

ontology and the broader semantic-web literature establish how IFC data can be represented as linked data to which such inferred knowledge is attached [43,44], all of which make it more important that inferred and authored information remain distinguishable downstream. IFC checking and abstraction pipelines have thus far adopted little of this discipline; verdicts are typically reported without evidence, and the derived geometry is indistinguishable from the authored geometry once it enters an analysis.

2.5. Research Gap

Prior work has addressed each of these concerns in isolation: rule-checking formalisms classify and execute requirements, IDS standardizes their machine-readable exchange, BIM-to-analysis studies transform physical models into analytical ones, and provenance models describe the data lineage in general terms. What is missing is a validation framework in which model defects, rule gaps, specification defects, non-applicability, geometric inference, and processing coverage are first-class, mutually exclusive outcomes of a single audit, a compliance rate, abstraction, or readiness claim that can be traced to what was actually checked, on what evidence, and with what omissions. This study aims to address this gap.

3. Materials and Methods

3.1. Framework Overview and Nondestructive Architecture

Figure 1 illustrates this framework. It is implemented as a local web application. A Python 3.12.10 server based on IfcOpenShell 0.8.4 [8] with NumPy 2.4.3 for array processing and openpyxl 3.1.5 for workbook export, parses and judges the model, and a browser viewer built on the Three.js library renders the geometry with verdicts as a color overlay. The input IFC is opened read-only and never rewritten; every verdict, abstraction, and readiness flag is stored in an overlay keyed by GlobalId so that disabling the overlay returns a plain viewer. Each run is pinned by a manifest recording the SHA-256 digests of the IFC file, the sBIM v8.3 requirement workbook, and its machine-readable derivation, together with code and library versions, hardware, and cache state. Caches are keyed by these digests, so a change to the rulebook or the code invalidates prior results automatically.

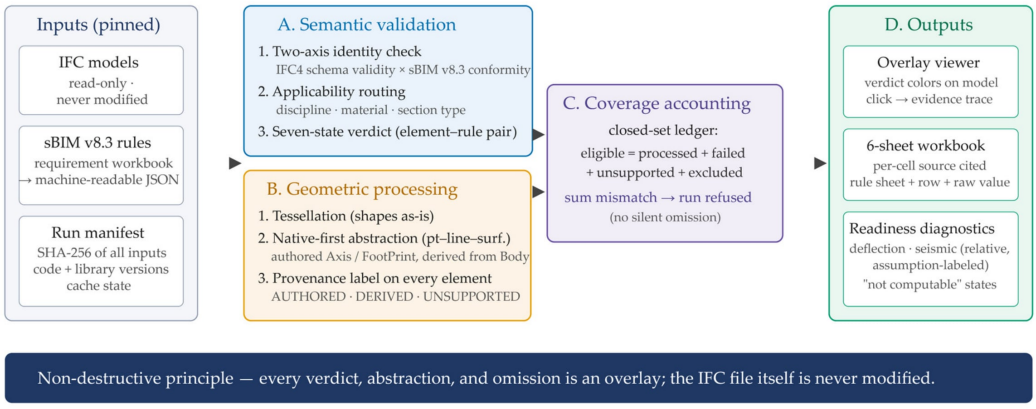


Figure 1. Framework overview. An IfcOpenShell-based Python server parses the read-only Industry Foundation Classes (IFC) model, evaluates the two verdict axes and the coverage ledger, and abstracts lightweight geometry with provenance labels; a browser viewer renders verdicts as a color overlay. Every run is pinned by a manifest of input and code digests.

3.2. Dataset and Requirement Rules

The evaluation uses eight production IFC4 models from a municipal digital safety management program: architecture, structure, and fire-protection models of four buildings, produced by two independent authoring/export pipelines (anonymized as A and B). Model sizes range from 1.97 to 16.09 MB, and eligible-element counts range from 364 to

1627 (5994 elements and 88,499 element–rule pairs in total; Table 1). Because no single building was processed by both pipelines, the dataset supports observations of exporter behavior but not vendor rankings (Section 5). The requirements come from version 8.3 of the sBIM information-requirement system [7]: a workbook whose sheets define, per entity class, the allowed PredefinedType classifications and the required properties by discipline, including conditional requirements. The workbook was mechanically converted into a JSON rule set, and both artifacts were hashed into the run manifest. Throughout this paper, importing a file successfully and processing all of its objects are reported as separate facts: the first is a property of the parser, and the second is a property of the coverage ledger.

Table 1. Characteristics of the eight IFC models (anonymized labels; A/B denote the two authoring pipelines).

Model	Pipeline	Discipline	Size (MB)	Elements	C_geo (%)	Element–Rule Pairs	P_native (%)
A1 Arch/Fire	A	Architecture/Fire	16.09	1627	96.3	14,033	93.1
A1 Struct	A	Structural	2.84	671	99.9	14,952	92.3
A2 Arch/Fire	A	Architecture/Fire	4.14	1108	97.8	8470	81.4
A2 Struct	A	Structural	3.64	567	100.0	14,484	83.5
B1 Arch/Fire	B	Architecture/Fire	13.58	389	98.5	5319	0.0
B1 Struct	B	Structural	3.17	633	99.8	17,235	0.0
B2 Arch/Fire	B	Architecture/Fire	6.53	635	89.3	5472	0.0
B2 Struct	B	Structural	1.97	364	100.0	8534	0.0

3.3. Two-Axis Identity Validation

For each element i, identity is judged as a pair $V_i = (V_{IFC,i}, V_{sBIM,i})$: validity with respect to the IFC4 schema and conformity with respect to sBIM v8.3 (Figure 2a). The judgment proceeds in that order. If the sBIM defines no rules for the entity class, the element is N/A. If PredefinedType carries a value outside the IFC4 enumeration for that class, the element is MISSING on the schema axis, a genuine schema violation. If PredefinedType is absent, the element is also MISSING but on different grounds: the attribute is optional in IFC4, so its absence is schema-valid, and the verdict records that the sBIM requirement for a classification is unmet; both cases are charged to the model author and are treated identically in the one-axis/two-axis comparison of Section 4.2 and by the oracle. In the eight models, all 85 MISSING identity verdicts were absences, and no out-of-enumeration value occurred. The value is then compared with the sBIM-allowed list for the class: a schema-valid value that the list does not contain is SBIM_UNSET, a gap in the project-rule system, not a violation by the model; a value that the list contains is OK, unless it is NOTDEFINED or USERDEFINED, which a list may admit but which carry no classification content and are therefore SUSPECT. The order matters in one case: where a project sheet does not admit NOTDEFINED—the space sheet of sBIM v8.3 is the only such sheet in this dataset—a NOTDEFINED element is recorded as SBIM_UNSET rather than SUSPECT. This is the precedence that the audited checker implements and that every reported count follows; Section 4.2 quantifies the alternative in which NOTDEFINED and USERDEFINED are flagged before the list is consulted. The checker evaluates the attribute of the occurrence only; the IFC4 convention by which a type object's PredefinedType applies when the occurrence's is absent was not evaluated, which Section 6 lists as a limitation. This last distinction matters in practice: values such as STANDARD or LANDING are perfectly legal in IFC4 and must not be reported as model errors merely because a project list omits them.

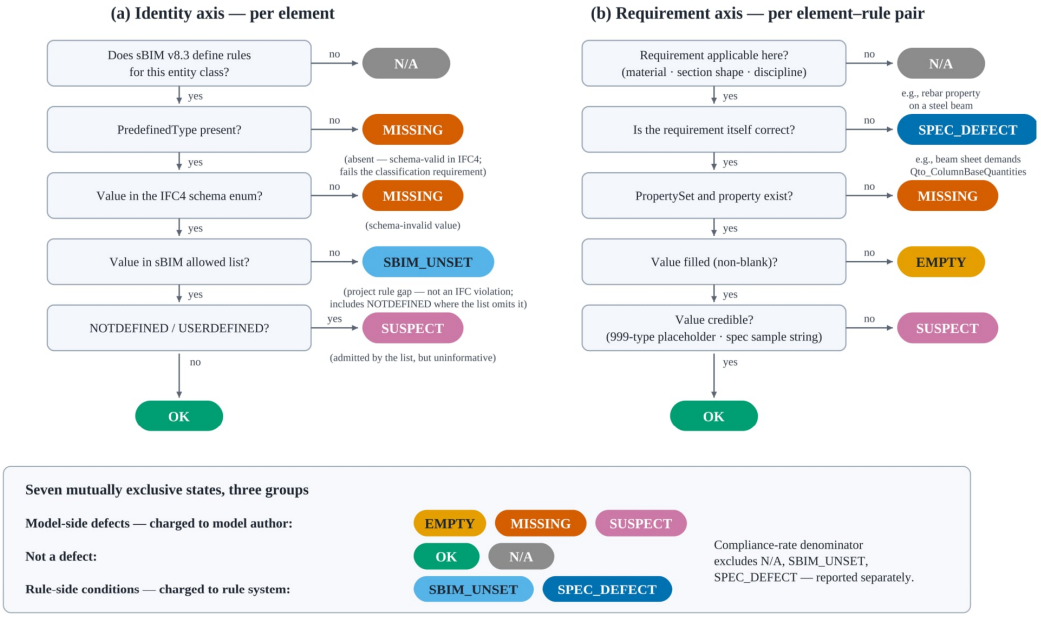


Figure 2. Verdict decision flow: (**a**) the identity axis, judging IFC4 schema validity and sBIM v8.3 conformity in precedence order; (**b**) the property axis, routing each element–rule pair by applicability before assigning one of the seven states. The flow shows the precedence implemented by the audited checker: NOTDEFINED/USERDEFINED is flagged SUSPECT only where the project list admits it, and the credibility tests are the placeholder and sample-string tests of Section 3.5.

3.4. Applicability-Aware Property Checking

The required properties are checked per element–rule pair, where the rule set for an element is selected by discipline and each rule is then routed by applicability (Figure 2b). Concrete reinforcement properties are not demanded of steel members, mutually exclusive section parameters are routed by profile shape (a rectangular profile parameter is not required for a radius-defined profile), and the conditional requirement expressions in the workbook are normalized before checking. Two requirement-side conditions are recognized explicitly: a rule that does not apply to the element at hand yields N/A, and a rule that is itself wrong—for example, a beam sheet demanding Qto_ColumnBaseQuantities, or a requirement referencing a property set that does not exist in the schema—yields SPEC_DEFECT and is charged to the rule system, not to the model author.

3.5. Seven-State Verdict Model

Each element–rule pair receives exactly one of the seven mutually exclusive verdict states (Table 2). The states are partitioned into model-side defects (EMPTY, MISSING, SUSPECT), non-defects (OK, N/A), and rule-side conditions (SBIM_UNSET, SPEC_DEFECT); only the first group is charged to the model author. SUSPECT belongs to that group in the accounting: it enters the denominator of the compliance rate and not the numerator, exactly the same as EMPTY and MISSING, and it differs from them only in that its evidence is circumstantial, so the verdict is flagged for human review rather than treated as final (Section 5.3). SUSPECT captures values that are present but not credible. The audited checker applies exactly three tests: (i) a numeric value equal to one of the placeholder magnitudes {999, 999.0, 9999, 99,999, −999, 0.999}, compared after numeric conversion and irrespective of unit; (ii) a string value containing both "/" and "$", the signature of the sample strings of the requirement tables pasted as values; and (iii) on the identity axis, NOTDEFINED or USERDEFINED where the project list admits them (Section 3.3). Two further credibility checks that the framework provides were not verdict tests in the audited run and therefore enter no reported rate: the population-constant test (a required property whose

value is identical across all members of a class with at least five members) is implemented in the audit module but is not invoked by the verdict path, and the section name–shape comparison (a profile name denoting an H-section on a member whose body polygon has at most five vertices) is raised as a warning by the structural module. Appendix D reports which test produced each SUSPECT verdict and where the verdicts concentrate. The compliance rate is computed over applicable pairs only—N/A, SBIM_UNSET, and SPEC_DEFECT are excluded from the denominator and reported as separate shares; thus, a model is neither punished for requirements that could not be applied to it nor flattened by counting placeholders as content.

Table 2. The seven verdict states.

State	Definition	Model Defect?	Charged to
OK	Requirement present, filled, and plausible	No	—
EMPTY	Field exists but the value is blank	Yes	Model author
MISSING	Required property, PropertySet, or classification absent (schema-valid absence that fails the requirement); or a value outside the IFC4 schema	Yes	Model author
SUSPECT	Value present but not credible: a 999-type placeholder magnitude, a specification sample string, or NOTDEFINED/USERDEFINED where the project list admits them	Yes (flagged for review)	Model author
SBIM_UNSET	Valid in IFC but no allowed value/rule specified by sBIM v8.3	No	Rule system (gap)
N/A	Requirement does not apply to this element, material, or condition	No	—
SPEC_DEFECT	Requirement table or its machine derivation is itself wrong	No	Rule system (defect)

Semantic checking is defined independently of geometry generation: an element whose shape fails to tessellate still receives full identity and property verdicts, and geometry failure affects only the geometry-coverage ledger.

3.6. Closed-Set Coverage Accounting

For every processing stage, the eligible population must be conserved as follows: $N_{\text{eligible}} = N_{\text{processed}} + N_{\text{failed}} + N_{\text{unsupported}} + N_{\text{excluded}}$ (Figure 3). The ledger is implemented as a first-class object; every input element must be placed into exactly one bucket, and if the buckets do not sum to the input population, the run is refused with an accounting error rather than being reported as a smaller success. The same ledger is maintained in five layers: semantic eligibility, rule mapping, geometry processing, lightweight abstraction, and structural readiness. The coverage line is displayed with every result. Declared exclusions are part of the ledger; for example, section properties are integrated for extruded profiles only, and mesh-defined sections are counted as excluded rather than being silently skipped.

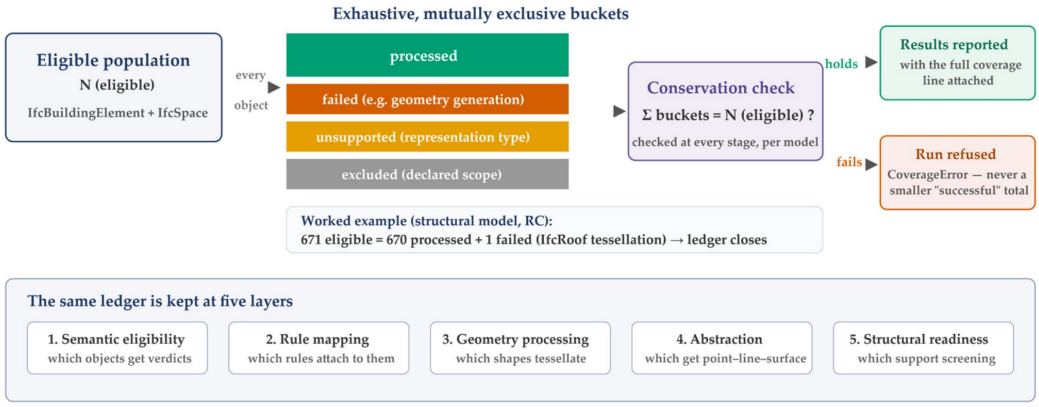


Figure 3. Closed-set coverage accounting. At every stage, the eligible population must partition exactly into processed, failed, unsupported, and excluded; a ledger that does not sum to the input population aborts the run rather than reporting a smaller success.

3.7. Provenance-Aware Lightweight Abstraction

Lightweight point–line–surface geometry is extracted native-first (Figure 4a). An authored structural analysis representation, where present, takes precedence; otherwise, the element's own axis (members) or footprint (slabs, walls) representation is used verbatim. Only when no authored representation exists is a centerline derived; for extruded body solids, the profile's area centroid is computed and swept along the extrusion direction. Elements whose representations support neither route are recorded as UNSUPPORTED by the abstraction ledger. Each abstracted element is a tuple $G_i = \{g_i, p_i, c_i, e_i\}$ of geometry, provenance (NATIVE_AXIS, NATIVE_FOOTPRINT, DERIVED_BODY, UNSUPPORTED), inferred confidence, and error versus reference; the provenance label follows the geometry into every downstream view and export, and authored and derived elements are never rendered identically (Figure 4b). The confidence and error components are quantified using the native-axis holdout experiment described in Section 3.9.

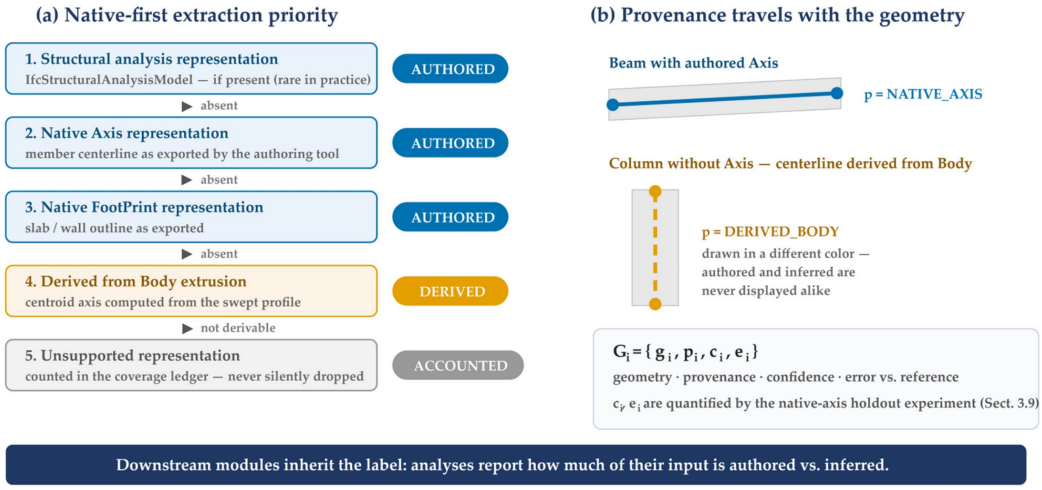


Figure 4. Provenance-aware abstraction. Lightweight geometry is extracted native-first, and every element is labeled NATIVE_AXIS, NATIVE_FOOTPRINT, DERIVED_BODY, or UNSUPPORTED; (**a**) the derivation schematic and (**b**) a viewer capture in which authored and derived elements are never rendered identically.

3.8. Downstream Data-Readiness Modules

The two structural modules demonstrate what the audited provenance-labeled abstraction supports and, deliberately, what it does not. The deflection module computes single-member deflections for horizontal beams under a simply supported, uniformly dis-

tributed load assumption, with dead and live loads attributed to IfcSpace load properties through geometric containment, tributary widths derived from space areas (capped, with capped values flagged), and spans subdivided at the detected intermediate supports. Material categories are inferred from IFC material names, after which predefined elastic moduli are assigned; placeholder loads (999) are excluded as SUSPECT, and section names that contradict the modeled shape are flagged. The equivalent-static seismic module applies a shear-building approximation with an assumed seismic coefficient and distributes story forces from the geometry-derived story weights. Both modules operate behind a readiness gate: where lateral-stiffness information, story definition, or load data are insufficient, the system reports "Not computable" instead of fabricating a value. All outputs are labeled as relative, assumption-listed diagnostics, and none constitutes a code-compliant structural assessment.

3.9. Experimental Design

Six experiments (E0–E5) evaluated the framework on the eight-model dataset, and Table 3 defines all the metrics.

Table 3. Metric definitions.

Metric	Definition	Note
C_geo	N_processed/N_eligible	geometry tessellation coverage
C_sem	N_semantically-checked/N_eligible	=100% after semantic–geometry decoupling
R_app	OK/(OK + EMPTY + MISSING + SUSPECT)	applicability-aware; N/A, SBIM_UNSET, SPEC_DEFECT excluded and reported separately
G_unset	SBIM_UNSET/N_IFC-valid	project-rule-gap rate (identity axis)
D_spec	SPEC_DEFECT/N_element–rule pairs	requirement-table defect rate
P_native	authored/(authored + derived)	share of lightweight geometry that is authored
Conformance	agreement(system, oracle) on decidable pairs	exact-match rate + confusion matrix (E1)
Detection	recall/false-alarm per injected-defect type	external ground truth (E3)

(E0) Metric reproducibility audit. All models were re-run from cold (no cache) under a pinned manifest, and 112 count cells (8 models × 2 axes × 7 states) plus the two headline rates were compared cell-by-cell against the result workbooks delivered during development. On the first pass, 9 cells differed, all explained by the single code change between the two runs—the semantic–geometry decoupling, which newly judged the 160 elements whose shapes had failed to tessellate, including the one B1 Struct element whose 24 rules account for the difference between the development total of 88,475 pairs and the audited 88,499; after the baseline was regenerated, all 112 cells matched. One rate claimed during development (an identity classification rate of 99.7% for A1 Struct) did not reproduce (99.5%) and was replaced by the reproduced value; two "before improvement" figures from the development log could not be re-run because the pre-change code had not been retained and were excluded for that reason, with the one-axis/two-axis comparison of Section 4.2 being computed from retained artifacts instead. No reproducible result was excluded.

(E1) Verdict conformance against an independent oracle. The verdict logic was verified against, rather than through polling of a human panel, a deterministic oracle re-implemented independently of the system and grounded in objective sources: the published IFC4 schema enumerations and the frozen machine-readable rule set derived from the sBIM v8.3 tables—the same rule set that the validator consumes, so the table-to-rule conversion

itself lies outside the oracle's scope—for the identity axis, and the model value itself for the value-decidable property verdicts (placeholder magnitudes → SUSPECT; the beam-sheet Qto_ColumnBaseQuantities defect → SPEC_DEFECT; a present, plausible value → OK). The oracle assigns a verdict wherever an objective verdict can be defined and abstains otherwise; system verdicts are compared over that decidable subset. Verdicts whose ground truth required model geometry or material—the MISSING/EMPTY/N/A distinctions on the property axis—were validated by defect injection (E3), whose ground truth was created by construction.

(E2) Native-axis holdout. For members that carry an authored axis, the axis is hidden and rederived from the body representation alone using the derivation path of the system. Because a single endpoint statistic conflates unrelated phenomena, the derived centerlines were compared on three separate axes: orientation (angular error), length (relative error), and position (endpoint distance), together with recovery and abstention rates and connectivity preservation.

(E3) Defect injection. Ten defect types were injected into working copies of the models (original hash verified as unchanged). Each injection was self-verified before judging, and detection and collateral effects were reported per type. Injections were confined to the two pipeline-A structural models: A1 Struct hosted eight types, and A2 Struct hosted the two that require steel members with authored axes and material associations (axis and material removal). For each type, the ten target elements were drawn with a fixed pseudo-random seed (20260901) from that type's candidate population in the host model—elements carrying a PredefinedType, elements carrying the targeted property set, beams with an authored axis representation, box-shaped members whose declared profile name is not an H-section, or members with a material association—without stratification by entity class. The sample is summarized in Appendix C (Table A2) and listed in full in Supplementary Table S1.

(E4) Ablations. Two components, SPEC_DEFECT separation and the body-derivation fallback, were evaluated by analytical ablation: rather than re-running a modified build, the contribution of each component was quantified from the ledgers of the audited run, which is possible because both components are deterministic relabelings (Section 4.8).

(E5) Performance. Cold-parse and warm-cache times, end-to-end processing, and peak memory were measured over 15 repetitions per condition (median, interquartile range), with scaling reported against element count and cache effects reported as an engineering property distinct from the parsing cost.

4. Results

4.1. Dataset and Coverage Characteristics

All eight files were imported without any errors. Geometry coverage C_geo ranged from 89.3% to 100.0%, and after semantic–geometry decoupling, semantic coverage C_sem reached 100.0% for every model; all eligible elements received identity and property verdicts regardless of whether their shape tessellated. The decoupling did more than adjust a metric: in one architecture/fire model (model A1 Arch/Fire), the 48 IfcCurtainWall elements had all failed tessellation and had therefore been omitted from judgment entirely, so an entire class went unreported under the coupled code. Once these elements were judged separately and received a SUSPECT verdict (NOTDEFINED), the model's identity compliance fell from an artifactual 100.0% to 91.6% (Appendix A, Figure A1). The coupling's effect went beyond distorting the rate—it hid the real defect population (Section 5.1). Provenance splits sharply by pipeline: pipeline-A models retain 81.4–93.1% authored geometry, whereas all four pipeline-B models had a P_{native} value of 0.0%—every lightweight element was derived because the delivered files carried no authored axes or footprints.

Whether these representations were never modeled or were discarded at export cannot be separated in this dataset, because no building passed through both pipelines (Section 5.3).

4.2. Verdict Distributions and Rule-Model Comparison

The eight models were evaluated along two verdict axes (Figure 5). On the identity axis, applicable compliance (R_{app}) among pipeline-A models ranged from 64.4% to 99.5% (model A1 Arch/Fire 91.6%, A1 Struct 99.5%, A2 Arch/Fire 64.4%, A2 Struct 79.1%), whereas all four pipeline-B models scored 0.0%, not because their elements were schema-invalid but because none carried a specific entity classification that the sBIM-allowed lists recognize. The required-property compliance spanned 0.0–67.2% across the dataset. The requirement-side states were substantial and, crucially, were reported separately rather than folded into the model's score; the project-rule-gap rate (G_unset) on the identity axis reached 47.7%, 41.0%, 28.2%, and 7.0% for the four pipeline-A models—schema-valid classifications for which sBIM v8.3 simply specifies no allowed value, while the requirement-table-defect rate (D_{spec}) ranged from 0.0% to 4.5%. These distributions were computed for all 88,499 element–rule pairs; the 24-pair difference from the development total of 88,475 is the one B1 Struct element whose shape failed to tessellate and whose 24 rules the geometry-coupled code had never judged (Section 4.1).

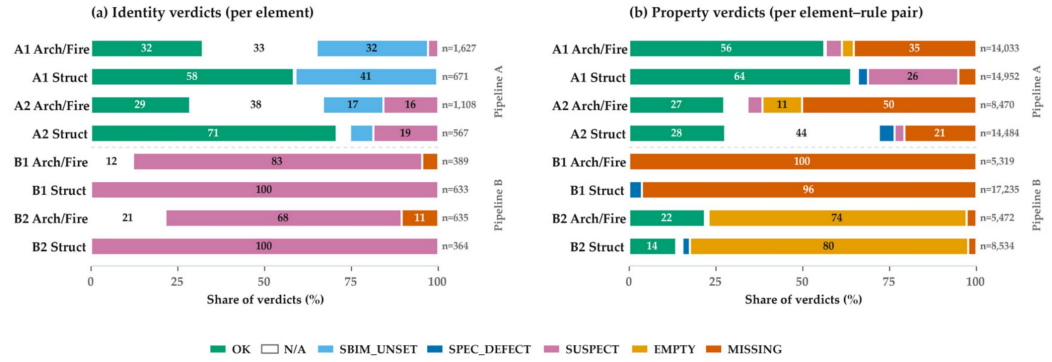


Figure 5. Per-model distribution of the seven verdict states across the eight IFC models. Pipeline-B models (right) receive no OK identity verdict because none of their elements carry a specific entity classification.

One-Axis Versus Two-Axis Verdict Rules

The effect of the two-axis separation is a key component of the identity result. Table 4 traces this effect on one structural model (671 elements), showing how a one-axis rule would classify these elements very differently from the two-axis rule, with no change to the underlying model.

Table 4. One-axis versus two-axis identity verdicts on model A1 Struct (671 elements). Only the rule differs; the model is identical.

A1 Struct Identity (671 Elements)	One-Axis Rule	Two-Axis Rule	Interpretation
OK	392	392	unchanged
MISSING → SBIM_UNSET	274 (reported as violations)	274 (reclassified as rule gap)	IfcWall = STANDARD (222) + IfcSpace = NOTDEFINED (52): valid IFC4 values
SUSPECT	2	2	unchanged
N/A	3	3	unchanged
Applicable compliance rate	58.7%	99.5%	model unchanged—rule only

Under the one-axis rule, 274 of the 671 elements of this model—222 walls typed as STANDARD and 52 spaces typed as NOTDEFINED—were reported as violations, which were reclassified as SBIM_UNSET, increasing the applicable compliance rate from 58.7% to 99.5% without any change to the model (Appendix A, Figure A2). The separation is not a blanket relabeling; in the two pipeline-B architectural models, 18 and 67 elements whose PredefinedType is genuinely absent—a schema-valid absence that fails the classification requirement—remain MISSING under both rules (Table 5). Because the historical outputs were produced by an earlier stand-alone implementation, we report this as a comparison of two verdict-rule models on identical inputs and not as a longitudinal improvement in a single pipeline.

The identity results are sensitive to one precedence choice. The 274 reclassified elements comprise 222 walls typed STANDARD and 52 spaces typed NOTDEFINED; under the implemented precedence (Section 3.3), the spaces are SBIM_UNSET because the sBIM space sheet does not admit NOTDEFINED, whereas the alternative precedence that flags NOTDEFINED and USERDEFINED before consulting the list would record them as SUSPECT. The alternative affects exactly the 172 NOTDEFINED spaces of the four pipeline-A models (52, 52, 34, and 34) and no pipeline-B element: identity SUSPECT would rise from 2084 to 2256, SBIM_UNSET would fall from 1027 to 855, and the identity compliance of the pipeline-A models would be 84.0%, 87.9%, 60.2%, and 74.1% (A1 Arch/Fire, A1 Struct, A2 Arch/Fire, A2 Struct) instead of 91.6%, 99.5%, 64.4%, and 79.1%. Because the one-axis rule charges the same 172 spaces as MISSING under either precedence, the reclassification counts of Tables 4 and 5 are unchanged; only their destination state differs (222 to SBIM_UNSET and 52 to SUSPECT for A1 Struct). All counts reported in this paper follow the implemented precedence.

Table 5. One-axis versus two-axis identity rates across all eight models. Reclassified counts (MISSING → SBIM_UNSET) are IFC4-valid values previously reported as violations; persistent MISSING counts are absent classifications that the two-axis rule does not excuse. Pipeline-B models expose no valid specific classification, so an applicable rate is undefined (—).

Model	1-Axis Rate	2-Axis Rate	Reclassified (MISS→UNSET)	Persistent MISSING	Pipeline
A1 Arch/Fire	47.9%	91.6%	522	0	A
A1 Struct	58.7%	99.5%	274	0	A
A2 Arch/Fire	46.2%	64.4%	193	0	A
A2 Struct	73.6%	79.1%	38	0	A
B1 Arch/Fire	—	—	0	18	B
B1 Struct	—	—	0	0	B
B2 Arch/Fire	—	—	0	67	B
B2 Struct	—	—	0	0	B

The property axis exhibited the same denominator effect. Applicability-aware recalculation corrected the apparent compliance rate by excluding non-applicable requirements and specification defects and reported the excluded shares separately (Figure 6b). For one structural model (model A2 Struct), 14,484 property pairs partitioned into OK 4000, model-side defects 3416, N/A 6414, and SPEC_DEFECT 654, giving an applicable rate of 53.9% versus a naive all-pairs rate of 27.6% on the same data—the largest such gap in the dataset, driven by that model's high count of non-applicable and defective requirements; for the other models, the correction was 0.2–3.3 points. The gap was a matter of the denominator and not the model quality (Figure 6).

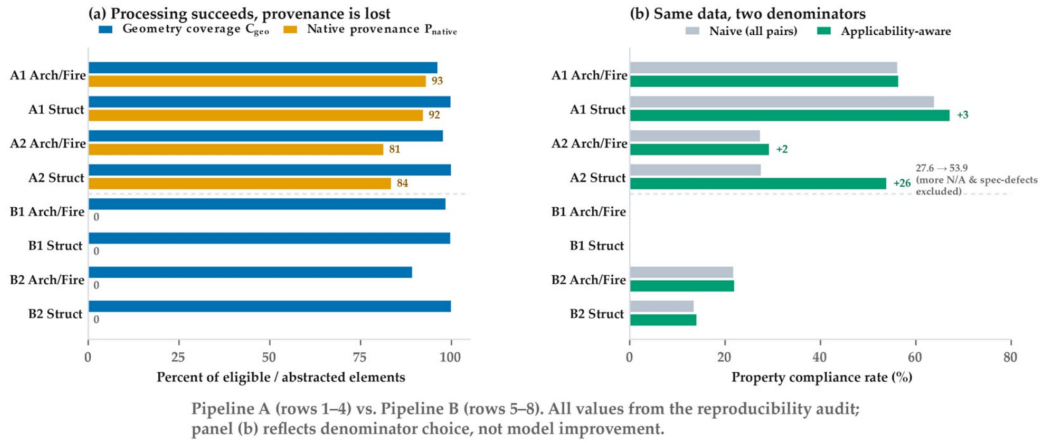


Figure 6. Confirmed per-model metrics: (**a**) geometry coverage and authored-versus-derived provenance shares; (**b**) the effect of applicability-aware denominators on the reported compliance rate.

4.3. Verdict Conformance Against the Oracle

Verdict correctness was established against a deterministic oracle re-implemented independently of the system, rather than by polling a human panel. Table 6 records the oracle's objective ground truth basis and the subset of verdicts it can decide on. Verdicts whose ground truth required model geometry or material were deferred to the defect-injection study of Section 4.5, whose ground truth was created by construction.

Table 6. Oracle ground truth basis and decidable scope.

Axis	Objective Ground Truth	States Decided	Out of Scope (→ Section 4.5)
Identity	IFC4 schema enumerations (published) + sBIM v8.3 allowed list (frozen machine-readable rule set); absence judged against the sBIM classification requirement	OK · MISSING · SUSPECT · SBIM_UNSET · N/A (all)	—
Property	the model value itself (placeholder/spec-defect/value present)	SUSPECT · SPEC_DEFECT · OK (value-decidable)	MISSING · EMPTY · N/A

The oracle decides a verdict only where an objective ground truth exists without interpreting how the requirement tables route a rule: the identity axis in full (5994 pairs) and the value-decidable property verdicts (33,340 pairs), together accounting for 39,334 pairs, or 41.6% of the 94,493 identity and property pairs (37.7% of the property axis alone). The remaining 55,159 property pairs—MISSING, EMPTY, and N/A, whose correctness depends on how the tables are interpreted and routed by discipline, material, and profile—lie outside the oracle and were exercised by defect injection (Section 4.5), with the limits stated there and in Section 6. The oracle is a separate implementation, written under the direction of J.L., who did not develop the validator, directly from the IFC4 schema enumerations and the frozen machine-readable rule set derived from the sBIM v8.3 tables; it shares no code with the validator, but it consumes the same rule set, so the table-to-rule conversion is common to both and is not verified by their agreement. Both implementations also embody the authors' reading of the tables; their agreement therefore establishes that the validator realizes the specification, not that the specification itself is correct.

System verdicts conformed exactly to the independent oracle over every decidable pair: all 5994 identity pairs across the eight models (now including the previously skipped elements: OK 1633; MISSING 85; SUSPECT 2084; SBIM_UNSET 1027; N/A 1165) and

all 33,340 value-decidable property pairs (OK 26,112; SUSPECT 5324; SPEC_DEFECT 1904)—totaling 39,334 decidable pairs—with zero discrepancies (Table 7).

Table 7. Verdict conformance (oracle versus system) for all eight models. Off-diagonal entries are zero.

Axis/Scope	Decidable States (Count)	Pairs	Agreement	Discrepancies
Identity (all)	OK 1633 · MISSING 85 · SUSPECT 2084 · SBIM_UNSET 1027 · N/A 1165	5994	100.0%	0
Property (value-decidable)	OK 26,112 · SUSPECT 5324 · SPEC_DEFECT 1904	33,340	100.0%	0
Combined decidable	Eight states	39,334	100.0%	0
Property (structure-dep.)	MISSING 35,266 · EMPTY 12,355 · N/A 7538 → Section 4.5	55,159	(oracle abstains)	—

The conformance was exact but not tautological: the oracle rederived each verdict from the IFC4 schema enumerations and the machine-readable rule set rather than from the validator's code; thus, the agreement established that the implementation realized the specification without deviation on the decidable subset and, in particular, that the non-trivial distinctions were correct, including 1027 schema-valid-but-unspecified classifications (SBIM_UNSET), 5324 non-credible values flagged as SUSPECT, and 1904 rulebook defects routed to SPEC_DEFECT rather than charged to the model. This is a conformance verification and not an external ground truth accuracy benchmark; the latter was supplied by the defect injection below. Structure-dependent verdicts outside the oracle's scope—35,266 MISSING, 12,355 EMPTY, and 7538 N/A property pairs requiring a model geometry or material to decide—were evaluated separately via the defect-injection study (Section 4.5).

4.4. Geometric Abstraction Accuracy

The native-axis holdout was run on the two structural models that carried authored beam axes: 545 eligible beams (218 in A1 Struct, 327 in A2 Struct), of which 432 were recovered and 113 abstained; the architectural and pipeline-B models exposed no beam/column axis representations and were therefore out of scope. Table 8 reports every metric against the full eligible population, so that the recovery and abstention rates are not separated from the accuracy of the recovered members. A single endpoint statistic conflated three distinct phenomena; therefore, we decomposed the error into orientation, length, and position (Table 8, Figure 7). In the reinforced-concrete model, 163 of 218 eligible beams (74.8%) were recovered and 55 (25.2%) abstained because their body was not an extrusion; in the steel model, 269 of 327 (82.3%) were recovered and 58 (17.7%) abstained (46 degenerate sections, 12 non-extruded bodies). Orientation was recovered exactly, with a median angular error of 0.00° in both models, except for 13 reinforced-concrete beams—6.0% of the eligible population and 8.0% of the recovered members—whose extrusion axis was selected orthogonally; these were reported as failure cases rather than folded into the median. Among the correctly oriented members, the length was recovered exactly in the steel model (median 0.00%) and with a systematic 8.5% shortening in the reinforced-concrete model, which also reduced its connectivity-preservation rate: 35 of 218 eligible beams (16.1%; 21.5% of those recovered) kept their connectivity, against 269 of 327 (82.3%; 100% of those recovered) in the steel model. Position, measured as endpoint distance, was dominated not by recovery error but by a datum-convention difference. For the steel model, where orientation and length were exact, endpoint distances fell into discrete clusters (300, 350, 150, 900, 925 mm) equal to roughly half the section depth—with the authored axis and the body-centroid axis referenc-

ing different planes. Therefore, we reported orientation and length as recovery-accuracy metrics and treated the position as a documented convention offset.

Table 8. Native-axis holdout on the full eligible population: recovery and abstention rates, orientation failures, and the length, position, and connectivity metrics of the recovered members.

Model	Eligible Beams	Recovered	Abstained (Reason)	Orientation Failures	Length Error, Median (Aligned)	Endpoint Distance, Median (Aligned)	Connectivity Kept
A1 Struct (RC)	218	163 (74.8%)	55 (25.2%; non-extruded body)	13 (6.0% of eligible; 8.0% of recovered)	−8.5% (shorter)	425 mm	35 (16.1% of eligible; 21.5% of recovered)
A2 Struct (steel)	327	269 (82.3%)	58 (17.7%; 46 degenerate section, 12 non-extruded body)	0	0.00%	300 mm (datum offset)	269 (82.3% of eligible; 100% of recovered)
Both models	545	432 (79.3%)	113 (20.7%)	13 (2.4% of eligible)	—	—	304 (55.8% of eligible)

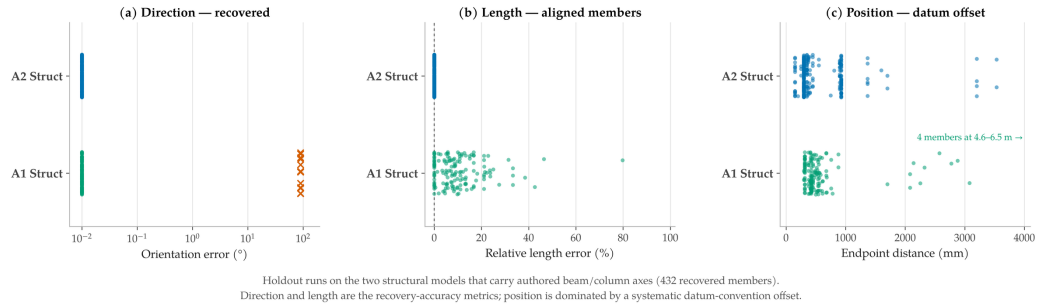


Figure 7. Native-axis holdout error decomposed into orientation, length, and position for the two structural models. The position component is dominated by a datum-convention offset rather than by recovery error.

4.5. Robustness: Defect Injection

Ten defect types were injected into the working copies (original SHA-256 digests were unchanged), and each injection was self-verified by re-reading the copy before judging (Table 9). Eight of the ten types produced the expected verdict in every case (10/10): seven model-side defects—deleted PredefinedType, PredefinedType set to NOTDEFINED, deleted PropertySet, blanked value, inserted 999 placeholder, renamed PropertySet, and name–shape mismatch—and one rule-side defect, an injected rulebook error that was correctly routed to SPEC_DEFECT rather than charged to the model (Figure 8). The remaining two types are reported differently by design: authored axis removal as a provenance outcome and material removal as a routing observation. Removing an authored axis was detected as a provenance downgrade for seven of the ten members; the remaining three had body representations from which no centerline could be derived, so the system recorded an abstention in the abstraction ledger—graceful degradation at the boundary of support, not a false negative. The name–shape mismatches—an H-section name written into the profile-name property of ten box-shaped reinforced-concrete beams of A1 Struct, the only host with such candidates (144 members)—were flagged by the reconstruction cross-check in all ten cases, which is why Figure 8 draws them as a credibility warning rather than as a verdict-change bar. Material removal did not behave as anticipated: removing one material association from each of ten steel members of A2 Struct changed no N/A routing in any case, evidently because the steel classification is also available from other material information in the file, so Table 9 records it as an observation without a detection rate, and the dependency it was designed to expose is discussed in Section 4.7. The blanked values were reported as EMPTY, not MISSING, confirming that the two states were distinct. Collateral

verdict changes were confined to injections on PropertySet instances that multiple elements shared (blank value 291, placeholder 304), where one edit legitimately altered every sharing element, and renaming and structural injections produced zero collateral. Ten injections per type are a functional test of deterministic logic against constructed ground truth rather than a statistical sample: each injection exercises a distinct element, property set, and code path, and a single miss would have exposed a logic error, so the counts are reported exactly rather than as rates with confidence intervals. They do not bound the frequency of structure-dependent misjudgments in the unperturbed population, and because all 100 injections were made in two structural models of pipeline-A, they verify the logic on reinforced-concrete and steel structural elements only; architectural and fire-protection classes and the pipeline-B files were not perturbed (Section 6).

Table 9. Injected-defect detection by type (10 injections per type). In the "Expected effect" column, the arrow (→) denotes the verdict or outcome that the injection was expected to produce.

Injected Defect	Expected Effect	Detected	Note
Delete PredefinedType	→ MISSING (schema-invalid)	10/10	identity axis
Set PredefinedType = NOTDEFINED	→ SUSPECT	10/10	no classification content
Delete PropertySet	→ MISSING	10/10	property axis
Blank a value	→ EMPTY (not MISSING)	10/10	EMPTY/MISSING kept distinct
Insert 999 placeholder	→ SUSPECT	10/10	placeholder magnitude
Rename PropertySet	→ MISSING	10/10	zero collateral
Inject rulebook defect	→ SPEC_DEFECT	10/10	charged to rule system
Remove authored axis	→ provenance downgrade	7/10 (+3 abstain)	graceful degradation
Name–shape mismatch	→ SUSPECT (credibility)	10/10	reconstruction cross-check
Remove material	→ routing change (anticipated)	0/10 changed	N/A routing unchanged (Section 4.7)

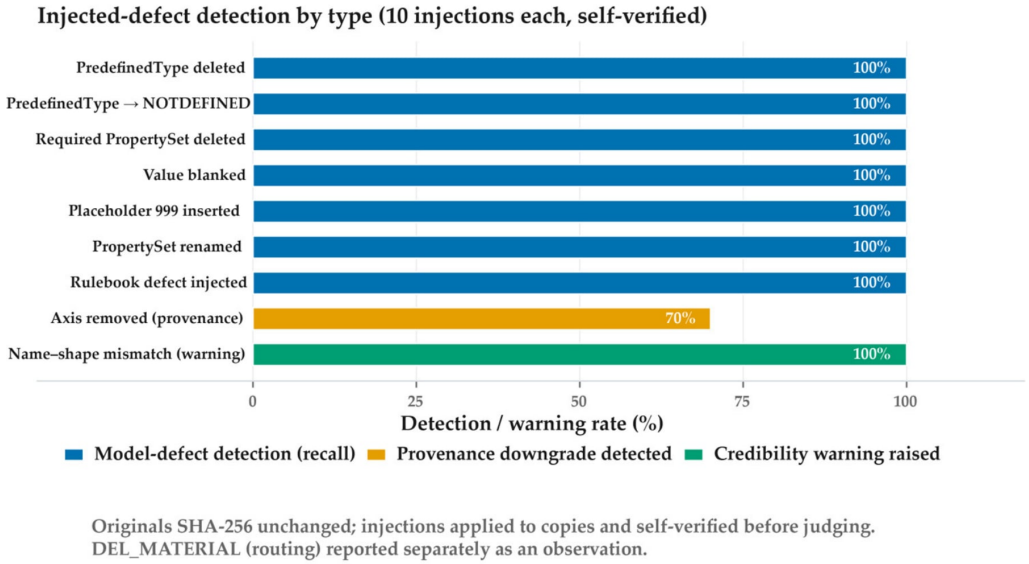


Figure 8. Detection rate for each injected-defect type. Blue bars: the seven types scored as verdict changes (six model-side defects and the rule-side SPEC_DEFECT routing), all detected at 10/10; orange: authored axis removal detected as a provenance downgrade (7/10, three recorded abstentions

at the boundary of derivation support); green: name–shape mismatch raised as a credibility warning (10/10). Material removal is reported qualitatively (Table 9) and is not plotted.

4.6. Computational Performance

Performance was measured over 15 repetitions per condition on a Windows 11 desktop workstation (Intel Core processor with 28 logical processors, 64 GB RAM; Python 3.12.10, IfcOpenShell 0.8.4), and all eight models were deterministic (a single-result fingerprint across repetitions). Cold end-to-end processing ranged from a median of 1.20 s (364 elements) to 18.61 s (1627 elements), scaling with element and triangle counts, and was dominated by geometry tessellation (Figure 9). Warm loading from the on-disk cache ran in 0.007–0.060 s and from the in-memory cache in less than 1 ms. Cold and warm were reported as separate conditions rather than as a single speed-up ratio because the cache was an engineering convenience distinct from the parsing cost. The peak resident memory was 233–553 MB. Axis-aligned bounding-box (AABB) broad-phase collision-candidate detection over same-building structural/architectural pairs was completed in 0.025–0.255 s with stable results across repetitions. Excel exports of the six-sheet workbook took 0.92–2.85 s (Table 10).

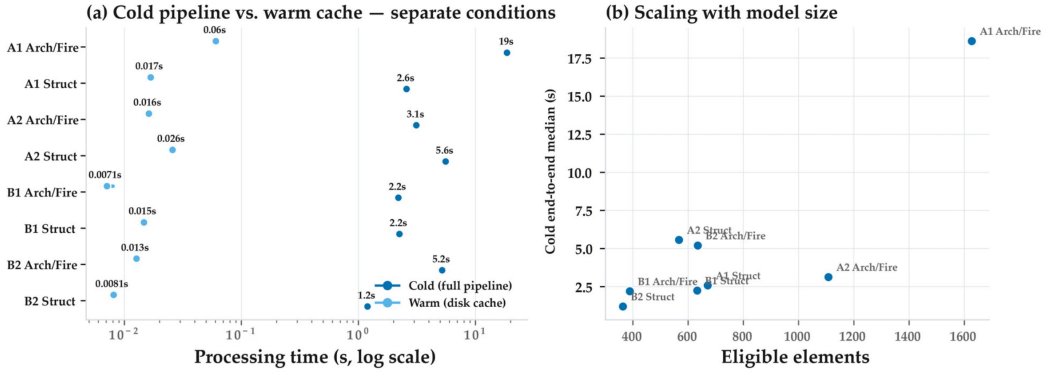


Figure 9. Computational performance: cold end-to-end processing time scales with model size, while warm-cache loading is orders of magnitude faster and is reported as a separate condition.

Table 10. Computational performance (median ranges over 15 repetitions).

Condition	Median Range	Unit	Repetitions	Note
Cold end-to-end	1.20–18.61	s	15	scales with element/triangle count
Warm (disk cache)	0.007–0.060	s	15	engineering convenience, not parse speed
Warm (memory cache)	<0.001	s	15	LRU hit
Peak resident memory	233–553	MB	15	sampled
AABB candidates	0.025–0.255	s	15	broad-phase; result-stable

4.7. Structural Data-Readiness Demonstration

The structural readiness gate returned the same verdict across all eight models; none carried the information required for a code-level structural evaluation. Material strengths, site and seismic class, response modification and importance factors, boundary conditions, and member-level loads were absent without exception, so both the deflection and the equivalent-static seismic modules reported “Not computable” wherever those inputs were required, rather than substituting default values. Geometry-derived story weights and an equivalent-static force distribution were computed as relative quantities for every model but never as code-compliant values. A single simply supported deflection comparison was performed on a beam whose span, section, and containment-attributed load could

all be extracted. The defect-injection study was designed to expose one further readiness dependency—that material completeness gates the applicability logic as well as the geometry—but the observation was the opposite of the anticipated one: removing a single material association from each of ten steel members of A2 Struct left every N/A routing unchanged, evidently because the steel classification is also carried by other material information in the file. Material completeness therefore gates routing only when no other material signal remains, a condition that none of the eight models exhibited; the dependency stands as a property of the routing logic rather than as an observed failure mode. These are concrete gaps that must be closed before structural screening can be performed.

4.8. Ablation of SPEC_DEFECT Separation and the Derivation Fallback

Both ablated components are deterministic relabelings, so their contribution can be read from the ledgers of the audited run without re-executing a modified build (E4, Section 3.9). Disabling SPEC_DEFECT separation restores the one-axis behavior in which a defective requirement is scored against the model: the 1904 SPEC_DEFECT property pairs (Table 7) are all consequences of the beam sheet's demand for Qto_ColumnBaseQuantities (436, 654, 632, and 182 pairs in the four structural models; none in the architecture/fire models) and are charged to the model authors as MISSING, because no beam carries that quantity set. Per model, the applicable property compliance falls from 67.2% to 65.2% (A1 Struct), from 53.9% to 49.6% (A2 Struct), and from 14.1% to 13.8% (B2 Struct) and stays at 0.0% for B1 Struct (Table 11); on the dataset as a whole, the applicable population grows from 79,057 to 80,961 pairs while the OK count is unchanged at 26,112, so aggregate compliance falls from 33.0% to 32.3%, and 1904 verdicts—2.4% of the applicable pairs—are misattributed to model authors.

Table 11. Analytical ablation from the audited ledgers: SPEC_DEFECT separation (property axis) and the body-derivation fallback (abstraction ledger). R_app is the applicable property compliance with and without the SPEC_DEFECT channel; abstracted share = (authored + derived)/N_eligible with the fallback and authored/N_eligible without it.

Model	SPEC_DEFECT Pairs	R_app, Props: With/Without Separation (%)	N_eligible	Authored	Derived (Body)	Abstained	Abstracted Share: With/Without Fallback (%)
A1 Arch/Fire	0	56.4/56.4	1627	644	48	935	42.5/39.6
A1 Struct	436	67.2/65.2	671	551	46	74	89.0/82.1
A2 Arch/Fire	0	29.3/29.3	1108	499	114	495	55.3/45.0
A2 Struct	654	53.9/49.6	567	444	88	35	93.8/78.3
B1 Arch/Fire	0	0.0/0.0	389	0	122	267	31.4/0.0
B1 Struct	632	0.0/0.0	633	0	413	220	65.2/0.0
B2 Arch/Fire	0	22.0/22.0	635	0	51	584	8.0/0.0
B2 Struct	182	14.1/13.8	364	0	145	219	39.8/0.0
All models	1904	33.0/32.3	5994	2138	1027	2829	52.8/35.7

Disabling the body-derivation fallback removes every derived lightweight element from the abstraction output instead of labeling it. The abstraction ledger (Table 11) places each eligible element on exactly one of three routes—authored (axis or footprint representation), derived (body), or abstained (no supported route, which includes classes outside the abstraction scope such as spaces and mesh-defined elements)—and conserves N_eligible. With the fallback, 2138 authored and 1027 derived elements are abstracted across the eight models (3165 of 5994 eligible elements, 52.8%); without it, the 1027 derived elements join

the abstentions, the four pipeline-B models lose their entire lightweight geometry (122, 413, 51, and 145 elements), and each pipeline-A model loses between 46 and 114 elements (6.9–18.6% of its abstracted members), all of which would then disappear from the structural readiness modules rather than appear as labeled derivations. The ablation therefore quantifies two things the main results imply: SPEC_DEFECT separation moves a small but systematic share of blame from authors to the rulebook, and the fallback is what makes the pipeline-B models analyzable at all—at the expense of their entire lightweight geometry being inferred, which the provenance label keeps visible.

5. Discussion

5.1. What Makes Automated Validation Trustworthy

The framework's claim of trustworthiness rests on separation rather than on the score. A high compliance rate is easy to manufacture—count placeholder values as content, demand nothing from objects that failed to process, or shrink the denominator quietly; each of these was observed during the development of the system itself. The most instructive instance appeared when we decoupled semantic checking from geometry generation: one architecture/fire model had reported 100.0% identity compliance only because its 48 curtain-wall elements, all of which failed tessellation, had been dropped from judgment altogether; once judged, they proved to be unclassified (NOTDEFINED), and the rate fell to 91.6%. Coupling verdict coverage with geometry does not merely perturb a number; it conceals the entire defective class. This is exactly the silent omission failure that closed-set coverage accounting prevents, which is why semantic and geometric coverage are now conserved on independent ledgers. The seven-state model forces every element–rule pair to answer two questions rather than one: Is something wrong, and whose fault is it? The reproducibility audit adds the corresponding operational discipline: because every run is pinned to the digests of its inputs and code, the delivered workbooks, the interactive overlay, and the numbers in this study constitute a single artifact rather than three renditions, and the audit rerun reproduces every verdict count exactly. Closed-set coverage accounting completes the argument by making omissions structurally impossible rather than procedurally discouraging them; an accounting failure aborts the run instead of shrinking it. We regard these three properties—two-sided verdicts, pinned provenance of results, and closed-set accounting—as a minimum standard for automated checking whose outputs are acted upon administratively.

5.2. Agreement Is Not Correctness

Two values that descend from the same source agree even when both are incorrect. Therefore, a shape-versus-attribute cross-check—a natural design for IFC validation—certifies consistency, not truth: for example, where a modeled shape and its stated dimension are both derived from the same bounding-box approximation, they match—while both misstate the real section. Pairwise cross-checking is structurally blind to these co-derived errors. This is why the SUSPECT state in our framework is not triggered by pairwise disagreement alone but by independent evidence that shares no derivation path with either channel: reconstruction from standardized section designations, placeholder detection, and population-level constancy tests. We suggest this as a general design rule for validation systems: any two channels that may share an origin must be audited against a third channel that does not share an origin.

5.3. Exporter-Side Information Loss and Public BIM Procurement

The starkest patterns in Table 1 and Figure 5 are not a single model's defect count but the difference between the authoring pipelines. Every pipeline-B model arrived with zero

usable entity classifications—each element typed NOTDEFINED or left untyped—and zero authored axes or footprints, so 100% of its lightweight geometry had to be derived; pipeline-A models retained 81–93% authored geometry. Because these were different buildings, no vendor comparison was conducted. What follows is therefore an observation rather than a demonstrated cause: in the delivered files of one pipeline, classifications and analytical representations were absent wholesale, a pattern consistent with export settings discarding them rather than with uneven modeling effort, although the two cannot be separated here because no building was processed by both pipelines. If export configuration is the cause, submission guides could address it by specifying fixed export settings across entire portfolios; establishing this would require the same building exported through both pipelines, which we identify as follow-up work. The rule-model comparison in Section 4.2 carries a complementary lesson for the rule side: under a one-axis regime, 274 schema-valid elements in one model were charged to the modeler, while genuinely defective elements elsewhere would have drowned among them. Requirement tables are also software—they age and contain errors, and the SPEC_DEFECT channel routes those errors to the maintainers of the rulebook instead of the authors of the model. Therefore, we argue that procurement-side checking requires three artifacts beyond the compliance rate: the coverage line, provenance shares, and requirement-side verdict counts.

Who acts on a verdict is part of the same procurement question, and it is where the SUSPECT state becomes operable. In the workflow for which the framework was built, the exported workbook lists every SUSPECT pair with the evidence that triggered it—the placeholder value, the population-wide constant, or the mismatched name–shape pair—and the reviewing authority's BIM manager records an accept-or-reject decision against the manifest digest of that run; SPEC_DEFECT rows are routed instead to the maintainers of the requirement tables. Because every run is pinned to its input digests, a resubmitted model is re-audited in full rather than patched, and a review decision attaches to the digest it was made against and expires with it. The recheck thus has a defined owner and a defined record, which is what distinguishes a reviewable verdict from an unresolved one.

5.4. From Readiness Diagnosis to Structural Screening

The deflection and seismic modules were deliberately positioned as diagnostics of what the submitted models could and could not support. Across all eight models, the information required for a code-level structural evaluation—material strengths, site class and spectral parameters, response modification and importance factors, boundary conditions, and analytical models—was absent in every case, indicating gaps that are systematic across both pipelines rather than isolated modeling lapses, whether they arise in modeling or at export. Readiness gates make this finding actionable by refusing to fabricate results—a story without extractable lateral stiffness is reported as not computable rather than assigned a median value. The provenance labels propagate into every structural view, so a screening result declares how much of its input geometry was authored and how much was derived. Because the verdict states, coverage ledger, and provenance labels are independent of the sBIM rule content, the same machinery applies wherever project requirements are machine-readable. Appendix B shows on an excerpt how the sBIM v8.3 rules are re-expressed as IDS 1.0 specifications and which of the seven states survive the translation; a full conversion would make the two requirement-side states applicable beyond the Korean context. Quantifying how geometric, material, load, and boundary-condition gaps propagate into screening uncertainty is the subject of a planned follow-up study.

The same readiness logic extends beyond structural screening to data-driven building operation. Operational analytics such as the automated fault detection and diagnosis (AFDD) of air-handling units are typically trained on sensor time series alone; the model

side can supply the context those series lack—equipment identities, system membership, design capacities, and spatial location—but only when the IFC model actually carries and validates that information, and integrating BIM with building automation and IoT data remains an active research problem whose main obstacle is inconsistent or missing metadata about equipment and systems [45,46]. Recent AFDD studies on real operational data from office buildings [47,48] illustrate the demand: equipment must be mapped, sensor variables contextualized, and system-specific features defined before a model can be trained. A coverage-accounted, provenance-aware validation report answers the preparatory question for such work—which equipment and system attributes are present, authored rather than inferred, and conformant to the requirement tables, and which are missing—so that gaps are identified before operational data are joined to the model. We regard this BIM–operational-data integration as a natural extension of the readiness modules and leave its demonstration on live AHU data to future work.

6. Conclusions, Limitations, and Future Work

Automated IFC validation earns trust not by assigning every object a verdict but by explicitly recording what is authored and what is derived, what applies and what does not, what is processed, and what is not, and, no less importantly, when the requirement itself rather than the model is at fault. We operationalized these separations as a two-axis, seven-state validation framework with closed-set coverage accounting and provenance-aware geometric abstraction and evaluated it on eight production IFC models. Verdict counts were reproducible under pinned inputs and conformed to an independent oracle over the 39,334 decidable element–rule pairs (41.6% of all pairs) without a single discrepancy. Of ten injected defect types, eight were detected in every case, and authored axis removal degraded gracefully with recorded abstentions; the framework surfaced defects that a geometry-coupled checker had hidden, quantified the wholesale absence of authored classifications and axes in every model delivered through one pipeline, and processed the largest model end-to-end in less than 20 s. Because every verdict, abstraction, and omission is exported with its evidence, the framework supports procurement-side model quality assurance today and provides the audited inputs required for uncertainty-aware structural screening of aging public buildings.

This study has nine limitations. First, the dataset comprised eight models from one national program and two authoring pipelines, with at most 1627 elements per model; scalability beyond this size range is not established. Second, no single building was processed by both pipelines, so exporter observations could not be separated from building effects. Third, the centerline derivation covered extruded solids with polygonal profiles, and the native-axis holdout could only run on the two structural models that carried authored beam/column axes; its position metric was further dominated by a datum-convention offset that we document rather than correct, and 8% of the reinforced-concrete members failed with an orthogonal axis flip. Fourth, the one-axis/two-axis comparison contrasted two implementations on identical inputs and not one implementation before and after a change. Fifth, verdict accuracy was established by conformance to a deterministic oracle over the objectively decidable subset (39,334 pairs, 41.6% of all pairs; the oracle consumes the same machine-readable rule set as the validator, so the table-to-rule conversion is not independently verified) and controlled defect injection with constructed ground truth, which verifies the logic on ten constructed cases per type, all injected into the two pipeline-A structural models, but does not bound the frequency of structure-dependent misjudgments in the unperturbed population or cover the architectural, fire-protection, and pipeline-B files; however, we did not conduct large-scale expert adjudication, which would test the requirement tables' own judgments rather than the system's faithfulness to them; this,

together with extending the oracle to the structure-dependent states and enlarging the injection study across disciplines and pipelines, remains a valuable avenue for future work. Sixth, no field measurements backed the models; therefore, validation concerned internal consistency and requirement conformity, not as-built fidelity. Seventh, the structural modules embedded the stated assumptions (simply supported members, uniform loads, and an assumed seismic coefficient) and inherited every data gap noted above; their outputs were only relative diagnostics. Eighth, the checker evaluates the PredefinedType of the occurrence only; the IFC4 convention by which a type object's PredefinedType applies when the occurrence's is absent was not evaluated, so a classification carried only by the type object is counted here as absent or NOTDEFINED. Ninth, the credibility tests that produce SUSPECT verdicts are the three stated in Section 3.5; the population-constant and name–shape checks were not verdict tests in the audited run, and no independent adjudication of naturally occurring SUSPECT, MISSING, EMPTY, or N/A verdicts was performed, so the accuracy of those verdicts is established only by the deterministic tests described and by the injection study. The evaluated rule content was specific to sBIM v8.3, although the framework itself was rule-agnostic.

Future work follows directly from these limitations. The IDS re-expression of Appendix B should be completed for the full workbook and executed through an IDS checker on the eight models. A stratified sample of naturally occurring structure-dependent verdicts should be adjudicated by independent experts; a 257-pair blind adjudication protocol has been prepared for this purpose. The injection study should be enlarged across disciplines and pipelines, and the same building should be exported through both pipelines under controlled settings to separate exporter from modeling effects. Type-object inheritance and the population-constant test should be evaluated as verdict inputs, and the NOTDEFINED precedence should be settled on the basis of the sensitivity reported in Section 4.2. Finally, validated model information should be joined to operational data for applications such as the fault detection of air-handling units discussed in Section 5.4.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/buildings16214263/s1: Table S1: Defect-injection targets (100 rows: injection type, host model, entity class, GlobalId, targeted property set, original value, change applied, expected and observed verdict, outcome).

Author Contributions: Conceptualization, J.L. and H.S.C.; methodology, J.L. and C.P.; software, W.C.; validation, W.C., J.L. and C.P.; formal analysis, J.L.; investigation, W.C. and C.P.; data curation, W.C.; writing—original draft preparation, J.L.; writing—review and editing, J.L., C.P. and H.S.C.; visualization, J.L.; supervision, H.S.C.; project administration, H.S.C. and C.P.; funding acquisition, H.S.C. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) grant funded by the Ministry of Land, Infrastructure and Transport (Grant RS-2022-00143584, Development of Safety Information Digitization and Integrated Safety Management System for Decrepit Buildings in Local Government).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The checker source code, anonymized IFC test models, run manifests, and reproducibility scripts (including the deterministic oracle) supporting the findings of this study are available from the corresponding author upon reasonable request. The sBIM v8.3 requirement tables are the property of KICT and cannot be redistributed. The machine-readable rule set used by the checker is a derivative of those tables and is subject to the same restriction; its structure and aggregate statistics are described in the text, and KICT provides the tables to researchers on request. Independent re-execution of the main experiment therefore requires that access, whereas

every reported count can be verified without re-execution from the per-element verdict workbooks, the run manifests (SHA-256 digests of the code, the rule set, and the models), and the oracle outputs, all of which are available from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest. The funders had no role in the study design; collection, analyses, or interpretation of data; writing of the manuscript; or decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

Abbreviation	Definition
AABB	Axis-Aligned Bounding Box
AFDD	Automated Fault Detection and Diagnosis
BIM	Building Information Modeling
IDS	Information Delivery Specification
IFC	Industry Foundation Classes
ISO	International Organization for Standardization
IoT	Internet of Things
RC	Reinforced Concrete
sBIM	Korean standardized BIM information-requirement system

Appendix A. Representative Validator Captures

The following figures capture the interactive validator operating on a representative production model (Model A1, pipeline-A), illustrating how the verdicts reported in Section 4 appear. Each capture pairs the read-only 3D model, colored by verdict, with the file provenance and coverage panel (left) and the verdict tally and systematic findings panel (right). The interface labels were translated into English, and the authoring tool and file identifiers were masked to preserve pipeline anonymity.

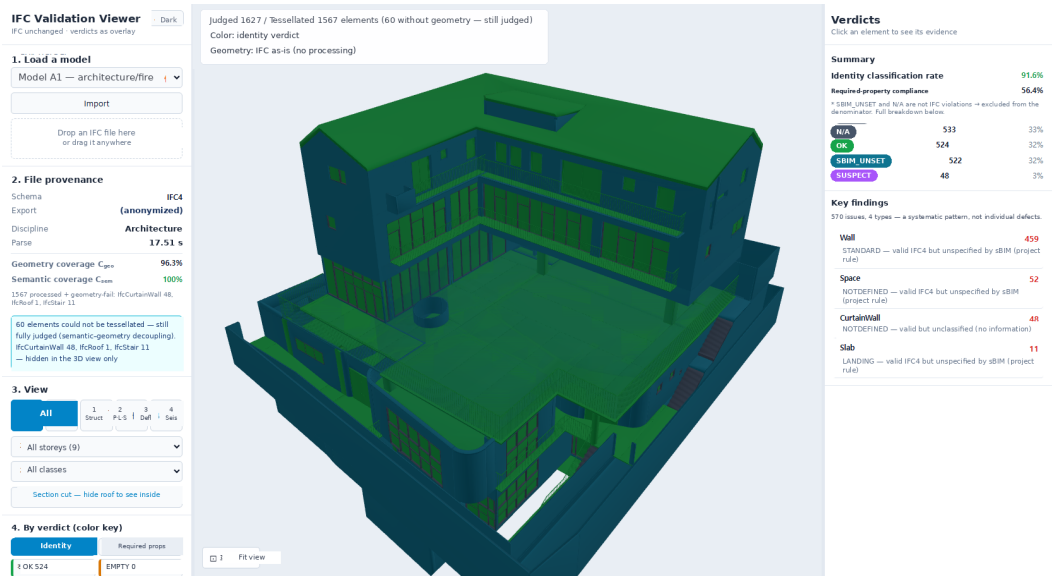


Figure A1. Validator capture for the architecture/fire model of building A1 (pipeline-A). Left: file provenance and coverage—IFC4 schema, C_geo = 96.3%, C_sem = 100%—with the note that the 60 elements whose geometry failed to tessellate (48 curtain walls, 11 stairs, 1 roof) are still fully judged after semantic–geometry decoupling. Center: the model colored by identity verdict (green = OK, blue = other states). Right: the verdict tally (identity classification rate 91.6%; OK 524, SBIM_UNSET 522, N/A 533, SUSPECT 48) and the systematic findings—459 walls typed STANDARD, 52 spaces and 48 curtain walls typed NOTDEFINED, 11 slabs typed LANDING. The 48 curtain walls constitute the entire class that a geometry-coupled checker had omitted from judgment (Section 4.1). The viewer

header reports both ledgers—1627 elements judged and 1567 tessellated—so the element count of Table 1 and the rendered count of the capture are reconciled.

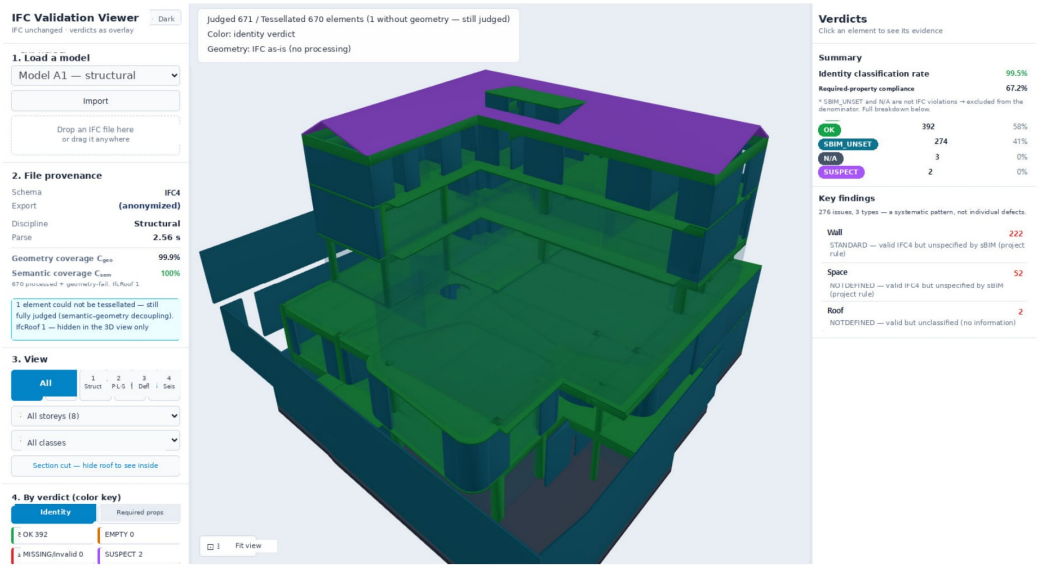


Figure A2. Validator capture for the structural model of the same building (Model A1, pipeline-A). Coverage C_geo = 99.9%, C_sem = 100%; the header reports both ledgers (671 elements judged, 670 tessellated). The identity classification rate is 99.5%: of the 671 elements, 392 were OK, and 274 were SBIM_UNSET—222 walls typed STANDARD and 52 spaces typed NOTDEFINED (the walls are shown in dark teal; spaces are drawn almost transparent), all IFC4-valid values for which sBIM v8.3 specifies no allowed entry; the purple surfaces are the two roof elements typed NOTDEFINED, a value the project list admits, which are therefore recorded as SUSPECT. This is the 274-element reclassification analyzed in Section 4.2 (Table 4). Other colors, the chips, the green headline rate, and the red counts in the right-hand panel are as in Figure A1.

Appendix B. Proof-of-Concept Mapping of sBIM v8.3 Rules to IDS 1.0

This appendix tests the claim of Sections 2.2 and 5.4 that the framework is rule-agnostic by re-expressing the rule shapes used in this study as Information Delivery Specification (IDS) 1.0 specifications [22] and recording which of the seven verdict states survive the translation. The excerpt uses the IFC4 standard names and the rule shapes already described in Section 3 (an allowed classification list, a discipline-routed property requirement, a conditional requirement, and a defective requirement); the project-specific allowed values of the sBIM v8.3 workbook are shown as placeholders because the workbook is not redistributable. The identity axis of Section 3.3 becomes three layered specifications over the same applicability (B-1a, B-1b, B-1c—the latter two differ from B-1a only in their enumeration and are given as comments), whose pass/fail pattern is decoded into OK, MISSING, SUSPECT, and SBIM_UNSET (Table A1); the property axis of Section 3.4 becomes property facets whose routing conditions are carried by applicability facets (B-2, B-3); and a requirement-table defect is included deliberately (B-4) to show that it is valid IDS. The listing validates against the IDS 1.0 XML schema (checked with ifctester 0.9.0).

```
<?xml version="1.0" encoding="UTF-8"?>
<ids xmlns="http://standards.buildingsmart.org/IDS"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://standards.buildingsmart.org/IDS
http://standards.buildingsmart.org/IDS/1.0/ids.xsd">
<info><title>sBIM v8.3 excerpt as IDS 1.0 (proof of concept)</title>
```

```
<version>0.1</version></info>
<specifications>
<!-- B-1a identity axis, schema layer: PredefinedType is an IfcWallTypeEnum value -->
<specification name="B-1a Wall PredefinedType is schema-valid" ifcVersion="IFC4">
<applicability minOccurs="1" maxOccurs="unbounded">
<entity><name><simpleValue>IFCWALL</simpleValue></name></entity>
</applicability>
<requirements>
<entity><name><simpleValue>IFCWALL</simpleValue></name>
<predefinedType><xs:restriction base="xs:string">
<xs:enumeration value="MOVABLE"/>      <xs:enumeration value="PARAPET"/>
<xs:enumeration value="PARTITIONING"/>  <xs:enumeration value="PLUMBINGWALL"/>
<xs:enumeration value="SHEAR"/>        <xs:enumeration value="SOLIDWALL"/>
<xs:enumeration value="STANDARD"/>     <xs:enumeration value="POLYGONAL"/>
<xs:enumeration value="ELEMENTEDWALL"/> <xs:enumeration value="USERDEFINED"/>
<xs:enumeration value="NOTDEFINED"/>
</xs:restriction></predefinedType></entity>
</requirements>
</specification>
<!-- B-1b identity axis, content layer: same applicability and facet as B-1a with
USERDEFINED and NOTDEFINED removed from the enumeration (fail = SUSPECT) -->
<!-- B-1c identity axis, project layer: same facet, enumeration = the allowed
values of the sBIM v8.3 wall sheet, which omit STANDARD (fail = SBIM_UNSET) -->
<!-- B-2 property axis, discipline-routed requirement (fail = MISSING or EMPTY) -->
<specification name="B-2 Structural beams declare LoadBearing" ifcVersion="IFC4">
<applicability minOccurs="1" maxOccurs="unbounded">
<entity><name><simpleValue>IFCBEAM</simpleValue></name></entity>
</applicability>
<requirements>
<property dataType="IFCBOOLEAN" cardinality="required">
<propertySet><simpleValue>Pset_BeamCommon</simpleValue></propertySet>
<baseName><simpleValue>LoadBearing</simpleValue></baseName>
</property>
</requirements>
</specification>
<!-- B-3 conditional requirement carried by applicability: concrete beams only;
a steel beam falls outside the applicability (= N/A) -->
<specification name="B-3 Concrete beams declare a strength class" ifcVersion="IFC4">
<applicability minOccurs="1" maxOccurs="unbounded">
<entity><name><simpleValue>IFCBEAM</simpleValue></name></entity>
<material><value><xs:restriction base="xs:string">
<xs:pattern value=".*([Cc]oncrete|CONCRETE).*"/></xs:restriction></value></material>
</applicability>
<requirements>
<property dataType="IFCLABEL" cardinality="required">
<propertySet><simpleValue>Pset_ConcreteElementGeneral</simpleValue></propertySet>
<baseName><simpleValue>StrengthClass</simpleValue></baseName>
</property>
</requirements>
</specification>
<!-- B-4 a requirement-table defect (Section 3.4): valid IDS that fails every beam -->
<specification name="B-4 Beams carry column base quantities" ifcVersion="IFC4">
<applicability minOccurs="1" maxOccurs="unbounded">
<entity><name><simpleValue>IFCBEAM</simpleValue></name></entity>
</applicability>
```

```
<requirements>
<property dataType="IFCLENGTHMEASURE" cardinality="required">
<propertySet><simpleValue>Qto_ColumnBaseQuantities</simpleValue></propertySet>
<baseName><simpleValue>Length</simpleValue></baseName>
</property>
</requirements>
</specification>
</specifications>
</ids>
```

Table A1 summarizes the correspondence. Four of the seven states are obtained by decoding the pass/fail pattern of the layered specifications, and N/A is carried by applicability; the two requirement-side states have no IDS counterpart. An IDS checker reports B-4 as a failure of every beam—the defect is charged to the model—and reports nothing about elements it could not load or that were excluded upstream, so SPEC_DEFECT routing and closed-set coverage accounting must be implemented by the tool that wraps the checker, which is what the framework does for the sBIM tables. Converting the full workbook and running it through an IDS checker on the eight models is identified as follow-up work.

Table A1. Correspondence between the seven verdict states and IDS 1.0 (decoding of the specifications in the listing above).

Verdict State	IDS 1.0 Expression	Decoded from	What the Translation Loses
OK	Every applicable specification passes	B-1a ∧ B-1b ∧ B-1c pass; B-2/B-3 pass	—
MISSING	Fail of the schema-layer identity specification (B-1a) or of a property facet (B-2)	B-1a fail; B-2 fail	An empty value counts as absent in IDS, so EMPTY and MISSING are reported identically
EMPTY	Fail of a property facet	B-2 fail	Collapses into MISSING
SUSPECT	Partly: the content layer B-1b (NOTDE-FINED/USERDEFINED); a placeholder magnitude only where a value restriction can exclude it	B-1a pass ∧ B-1b fail	Population-wide constants and name–shape mismatches require cross-element and geometric reasoning that IDS does not express
SBIM_UNSET	Fail of the project layer B-1c while B-1a and B-1b pass	B-1a ∧ B-1b pass ∧ B-1c fail	IDS reports only a failure; the schema-valid-but-unspecified distinction exists only in the decoding table—that table is the two-axis rule itself
N/A	Element outside the applicability of the specification (entity, material, property facets)	Applicability of B-3	Equivalent where the routing condition is expressible as a facet; profile-shape routing needs a property facet on the profile type
SPEC_DEFECT	Not expressible: B-4 is valid IDS 1.0 and fails every beam	—	The rulebook defect is charged to the model; an authoring-time audit of the IDS file can catch a non-existent property set but not a wrong-but-existing one
Coverage line	Not expressible: IDS reports applicable/pass/fail counts per specification	—	Elements the checker could not load or that failed upstream are outside its report; closed-set accounting must wrap the checker

Appendix C. Sampled Elements of the Defect-Injection Study

Table A2 summarizes, for each of the ten injection types of Table 9, the host model, the candidate population from which the ten targets were drawn with the fixed seed

20260901, the entity classes of the ten targets, and the outcome recorded in the injection log. All 100 injections were made in the two pipeline-A structural models (80 in A1 Struct, reinforced concrete; 20 in A2 Struct, steel); the originals were untouched (identical SHA-256 digests before and after), every injection was self-verified by re-reading the working copy, and the ledger of outcomes is 77 detected, 3 abstentions, and 20 recorded observations. The full list—GlobalId, targeted property set, original value, change applied, expected and observed verdict—is provided as Supplementary Table S1.

Table A2. Sampling frame and outcome of the defect-injection study (seed 20260901; ten targets per type).

Injection Type	Host	Candidate Population	Entity Classes of the Ten Targets	Outcome
Delete PredefinedType	A1 Struct	Elements carrying a PredefinedType	IfcSlab 5, IfcBeam 3, IfcColumn 2	10/10 MISSING
Set PredefinedType = NOTDEFINED	A1 Struct	Elements carrying a PredefinedType	IfcBeam 6, IfcSlab 3, IfcColumn 1	10/10 SUSPECT
Delete PropertySet	A1 Struct	Elements carrying a required property set	IfcSlab 3, IfcBeam 3, IfcWall 2, IfcColumn 2	10/10 MISSING (shared instances propagate)
Blank a value	A1 Struct	Required properties with a value	IfcWall 3, IfcBeam 3, IfcColumn 2, IfcSlab 1, IfcSpace 1	10/10 EMPTY
Insert 999 placeholder	A1 Struct	Numeric required properties	IfcBeam 5, IfcWall 4, IfcSlab 1	10/10 SUSPECT
Rename PropertySet	A1 Struct	Required property sets (unshared instances)	IfcWall 4, IfcBeam 3, IfcSlab 2, IfcColumn 1	10/10 MISSING, zero collateral
Inject rulebook defect	A1 Struct	Beams judged against a working copy of the beam sheet	IfcBeam 10	10/10 SPEC_DEFECT
Remove authored axis	A2 Struct	Beams with an authored axis representation	IfcBeam 10	7 derived, 3 abstentions
Name–shape mismatch	A1 Struct	Box-shaped members whose declared profile name is not an H-section (144)	IfcBeam 10	10/10 credibility warning
Remove material	A2 Struct	Members with a material association	IfcBeam 8, IfcColumn 2	0/10 routing change (N/A unchanged)

Appendix D. Where the SUSPECT Verdicts Come From

Table A3 attributes every SUSPECT verdict of the audited run to the test that produced it. All 5324 property-axis SUSPECT verdicts were produced by the placeholder test (values of 999 or 999.0); the sample-string test fired in no model. The verdicts concentrate in a few custom reinforcement and label properties of the pipeline-A models, a pattern consistent with placeholders written by an authoring template rather than by individual error, which is what the review workflow of Section 5.3 is designed to resolve. The two length values of 999 in B2 Struct are the only cases in which the placeholder test could plausibly have flagged a genuine value (a 999 mm length); they are retained as SUSPECT because the test is deterministic, and they illustrate its false-positive mode. On the identity axis, all 2084 SUSPECT verdicts are NOTDEFINED or USERDEFINED values admitted by the project list of the class concerned.

Table A3. SUSPECT verdicts of the audited run by model, test, and property (property axis: placeholder test only; identity axis: NOTDEFINED/USERDEFINED admitted by the project list).

Model	Property SUSPECT Pairs	Elements Affected	Distinct Properties	Most Frequent Properties (Pairs)	Identity SUSPECT (Value, Class)
A1 Arch/Fire	654	221	9	Effective_Depth 198, Concrete_Cover 198, Reinforcement_Area_Ratio 198	48 (NOTDEFINED, IfcCurtainWall)
A1 Struct	3889	558	33	Effective_Depth 281, Concrete_Cover 281, Reinforcement_Area_Ratio 281	2 (NOTDEFINED, IfcRoof)
A2 Arch/Fire	365	291	11	Reference 242 (two property sets), Cross_Section_Area 29	175 (USERDEFINED 156, NOTDEFINED 19; IfcCurtainWall)
A2 Struct	414	111	10	Rebar-label fields X2/X3/X4 92 each	106 (USERDEFINED, IfcBeam)
B1 Arch/Fire	0	0	0	—	324 (NOTDEFINED; eight classes)
B1 Struct	0	0	0	—	633 (NOTDEFINED; IfcBeam 316, IfcSlab 220, IfcColumn 97)
B2 Arch/Fire	0	0	0	—	432 (NOTDEFINED; seven classes)
B2 Struct	2	2	1	Length 2 (value 999; possibly a genuine 999 mm)	364 (NOTDEFINED; four classes)
All models	5324	1183	—	placeholder test 5324; sample-string test 0	2084

References

1. Nam, S.; Kim, J.; Lee, D. Current status of aged public buildings and effect analysis prediction of green remodeling in South Korea. *Sustainability* **2021**, *13*, 6649. [CrossRef]
2. Cho, K.; Mun, J.; Park, H.; Kim, T. Preliminary assessment method for structural performance (PAMSP) of RC Rahmen structure building during Scan-to-BIM procedure. *Sustainability* **2024**, *16*, 4663. [CrossRef]
3. Kang, S.; Kim, S.; Kim, S. Improvement of the defect inspection process of deteriorated buildings with scan-to-BIM and image-based automatic defect classification. *J. Build. Eng.* **2025**, *99*, 111601. [CrossRef]
4. Bartels, N.; Eilers, M.; Pütz, C.; Meins-Becker, A. IFC-based linking of the risk management process using a building data model. *IOP Conf. Ser. Earth Environ. Sci.* **2022**, *1101*, 092001. [CrossRef]
5. *ISO 16739-1:2024*; Industry Foundation Classes (IFC) for Data Sharing in the Construction and Facility Management Industries—Part 1: Data Schema. International Organization for Standardization: Geneva, Switzerland, 2024.
6. Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds.) *Building Information Modeling: Technology Foundations and Industry Practice*; Springer: Cham, Switzerland, 2018.
7. Korea Institute of Civil Engineering and Building Technology. *SBIM Information-Requirement System, Version 8.3*; KICT: Goyang, Republic of Korea, 2026.
8. IfcOpenShell: The Open-Source IFC Toolkit and Geometry Engine. Available online: https://ifcopenshell.org (accessed on 18 September 2026).
9. Amor, R.; Dimyadi, J. The promise of automated compliance checking. *Dev. Built Environ.* **2021**, *5*, 100039. [CrossRef]
10. Eastman, C.; Lee, J.-M.; Jeong, Y.-S.; Lee, J.-K. Automatic rule-based checking of building designs. *Autom. Constr.* **2009**, *18*, 1011–1033. [CrossRef]
11. Solihin, W.; Eastman, C. Classification of rules for automated BIM rule checking development. *Autom. Constr.* **2015**, *53*, 69–82. [CrossRef]

12. Hjelseth, E.; Nisbet, N. Capturing normative constraints by use of the semantic mark-up RASE methodology. In Proceedings of the CIB W78-W102 2011 International Conference, Sophia Antipolis, France, 26–28 October 2011.
13. Preidel, C.; Borrmann, A. Towards code compliance checking on the basis of a visual programming language. *J. Inf. Technol. Constr. (ITcon)* **2016**, *21*, 402–421.
14. Zhang, J.; El-Gohary, N.M. Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking. *Autom. Constr.* **2017**, *73*, 45–57. [CrossRef]
15. Iversen, O.; Huang, L. Leveraging large language models for BIM-based automated compliance checking. *Autom. Constr.* **2026**, *182*, 106707. [CrossRef]
16. Shi, J.W.L.; Solihin, W.; Yeoh, J.K.W. Fine-tuning a large language model for automated code compliance of building regulations. *Adv. Eng. Inform.* **2025**, *68*, 103676. [CrossRef]
17. Ma, Z.; Zhu, H.; Xiang, X.; Turk, Ž.; Klinc, R. Automatic compliance checking of BIM models against quality standards based on ontology technology. *Autom. Constr.* **2024**, *166*, 105656. [CrossRef]
18. Jia, L.; Chen, M.; Chen, C.; Jin, Y. A Semantic Web and IFC-based framework for automated BIM compliance checking. *Buildings* **2025**, *15*, 2633. [CrossRef]
19. *ISO 19650-1:2018*; Organization and Digitization of Information about Buildings and Civil Engineering Works, Including Building Information Modelling (BIM)—Information Management Using Building Information Modelling—Part 1: Concepts and Principles. International Organization for Standardization: Geneva, Switzerland, 2018.
20. *ISO 29481-1:2016*; Building Information Models—Information Delivery Manual—Part 1: Methodology and Format. International Organization for Standardization: Geneva, Switzerland, 2016.
21. Venugopal, M.; Eastman, C.M.; Sacks, R.; Teizer, J. Semantics of model views for information exchanges using the industry foundation class schema. *Adv. Eng. Inform.* **2012**, *26*, 411–428. [CrossRef]
22. Lee, Y.-C.; Eastman, C.M.; Solihin, W. Logic for ensuring the data exchange integrity of building information models. *Autom. Constr.* **2018**, *85*, 249–262. [CrossRef]
23. buildingSMART International. *Information Delivery Specification (IDS), Version 1.0*; buildingSMART International: Zurich, Switzerland, 2024. Available online: https://www.buildingsmart.org (accessed on 18 September 2026).
24. Fischer, S.; Urban, S.; Schranz, H.; Zucker, C. Bridging the gap between tabular information requirements and the Information Delivery Specification (IDS). *Buildings* **2025**, *15*, 1017. [CrossRef]
25. Fischer, S.; Urban, H.; Schranz, C.; Loibl, P.; van Berlo, L. Extending Information Delivery Specifications for digital building permit requirements. *Dev. Built Environ.* **2024**, *20*, 100560. [CrossRef]
26. Okonta, E.D.; Okeke, F.O.; Mgbemena, E.E.; Nnaemeka-Okeke, R.C.; Guo, S.; Awe, F.C.; Eke, C. An intelligent natural language processing (NLP) workflow for automated smart building design. *Buildings* **2025**, *15*, 2413. [CrossRef]
27. Hosseini Gourabpasi, A.; Jalaei, F.; Ghobadi, M. Developing an openBIM information delivery specifications framework for operational carbon impact assessment of building projects. *Sustainability* **2025**, *17*, 673. [CrossRef]
28. Ramaji, I.J.; Memari, A.M. Interpreted information exchange: Systematic approach for BIM to engineering analysis information transformations. *J. Comput. Civ. Eng.* **2016**, *30*, 04016028. [CrossRef]
29. Jia, J.; Gao, J.; Wang, W.; Ma, L.; Li, J.; Zhang, Z. An automatic generation method of finite element model based on BIM and ontology. *Buildings* **2022**, *12*, 1949. [CrossRef]
30. Lee, W.; Jeong, Y.; Jeong, W.; Yu, Y.; Park, S.I.; Koo, B. An IFC-based framework for automated integration of structural analysis results to support BIM-based code compliance. *Buildings* **2026**, *16*, 746. [CrossRef]
31. Sun, Z.; Wang, C.; Wu, J. Industry Foundation Class-based building information modeling lightweight visualization method for steel structures. *Appl. Sci.* **2024**, *14*, 5507. [CrossRef]
32. Mirahadi, F.; McCabe, B.; Shahi, A. IFC-centric performance-based evaluation of building evacuations using fire dynamics simulation and agent-based modeling. *Autom. Constr.* **2019**, *101*, 1–16. [CrossRef]
33. Singh, T.; Mahmoodian, M.; Wang, S. Enhancing open BIM interoperability: Automated generation of a structural model from an architectural model. *Buildings* **2024**, *14*, 2475. [CrossRef]
34. Gerbino, S.; Cieri, L.; Rainieri, C.; Fabbrocino, G. On BIM interoperability *via* the IFC standard: An assessment from the structural engineering and design viewpoint. *Appl. Sci.* **2021**, *11*, 11430. [CrossRef]
35. Du, S.; Hou, L.; Zhang, G.; Tan, Y.; Mao, P. BIM and IFC data readiness for AI integration in the construction industry: A review approach. *Buildings* **2024**, *14*, 3305. [CrossRef]
36. W3C. *PROV-DM: The PROV Data Model*; W3C Recommendation; World Wide Web Consortium: Wakefield, MA, USA, 2013. Available online: https://www.w3.org/TR/prov-dm/ (accessed on 18 September 2026).
37. Boje, C.; Guerriero, A.; Kubicki, S.; Rezgui, Y. Towards a semantic Construction Digital Twin: Directions for future research. *Autom. Constr.* **2020**, *114*, 103179. [CrossRef]
38. Tavakoli, P.; Yitmen, I.; Sadri, H.; Taheri, A. Blockchain-based digital twin data provenance for predictive asset management in building facilities. *Smart Sustain. Built Environ.* **2024**, *13*, 4–21. [CrossRef]

39. Belsky, M.; Sacks, R.; Brilakis, I. Semantic enrichment for building information modeling. *Comput.-Aided Civ. Infrastruct. Eng.* **2016**, *31*, 261–274. [CrossRef]
40. Bloch, T.; Sacks, R. Comparing machine learning and rule-based inferencing for semantic enrichment of BIM models. *Autom. Constr.* **2018**, *91*, 256–272. [CrossRef]
41. Koo, B.; Shin, B. Applying novelty detection to identify model element to IFC class misclassifications on architectural and infrastructure building information models. *J. Comput. Des. Eng.* **2018**, *5*, 391–400. [CrossRef]
42. Mirarchi, C.; Gholamzadehmir, M.; Daniotti, B.; Pavan, A. Semantic enrichment of BIM: The role of machine learning-based image recognition. *Buildings* **2024**, *14*, 1122. [CrossRef]
43. Pauwels, P.; Terkaj, W. EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Autom. Constr.* **2016**, *63*, 100–133. [CrossRef]
44. Pauwels, P.; Zhang, S.; Lee, Y.-C. Semantic web technologies in AEC industry: A literature overview. *Autom. Constr.* **2017**, *73*, 145–165. [CrossRef]
45. Tang, S.; Shelden, D.R.; Eastman, C.M.; Pishdad-Bozorgi, P.; Gao, X. A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends. *Autom. Constr.* **2019**, *101*, 127–139. [CrossRef]
46. Balaji, B.; Bhattacharya, A.; Fierro, G.; Gao, J.; Gluck, J.; Hong, D.; Johansen, A.; Koh, J.; Ploennigs, J.; Agarwal, Y.; et al. Brick: Metadata schema for portable smart building applications. *Appl. Energy* **2018**, *226*, 1273–1292. [CrossRef]
47. Wang, S. Automated fault detection and diagnosis of AHUs via tabular-based methods using operational data from a large office building. *J. Comput. Civ. Eng.* **2026**, *40*, 04026018. [CrossRef]
48. Wang, S. Empirical comparison of sequence-aware augmentation strategies for real operational AHU AFDD under data scarcity and class imbalance. *Expert Syst. Appl.* **2026**, *326*, 132818. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.