2026 European Conference on Computing in Construction
Corfu, Greece
July 12–15, 2026

ALIGNING HETEROGENEOUS VIEWS: STAKEHOLDER TERMINOLOGY AND MULTI-LEVEL MAPPINGS IN THE AEC INDUSTRY

Jakob Martin[1], Cornelius Preidel[1], Simon Vilgertshofer[1], André Borrmann[2]
[1]University of Applied Sciences, Munich, Germany
[2]Technical University of Munich, Munich, Germany

Abstract

Construction stakeholders refer to the same asset using discipline-specific terms and abstraction levels, producing representations with inconsistent naming conventions and modeling granularity. Alignment across views often fails even with standards such as IFC unless semantic relations and transformations are made explicit. We contribute a typology of cross-view mapping requirements as a preceding decision layer for alignment. It determines the mapping type, intended semantic relation, and required transformation between views, guiding candidate generation and validation. Using representative examples from planning, construction, and operations, we show why cross-view alignment requires more than label matching alone.

Introduction

Semantic interoperability and semantic consistency remain challenges in the Architecture, Engineering, Construction, and Operations (AECO) domain, because project information is produced and consumed through many discipline-specific tools, standards, and organization-specific conventions (Ozturk, 2020; Pauwels et al., 2017). Beyond technical friction, poor data quality causes major economic losses, including rework triggered by data-related decision errors (FMI, 2021). While open standards such as Industry Foundation Classes (IFC) are well established in practice and widely adopted to support information exchange, they still fall short of enabling fully consistent and seamless integration. Persistent challenges arise from inconsistent naming conventions, incompatible property structures, and divergent semantics, particularly when project-specific extensions and local naming conventions are applied (Ozturk, 2020; Beck, 2023). As a result, project information ecosystems and exchange workflows frequently fail to provide consistent, linkable, and computationally interpretable data across stakeholder boundaries.

These recurring mismatches are closely related to ontology matching, which provides established primitives and workflows for relating heterogeneous schemas and knowledge representations, including correspondence types, confidence scoring, and hybrid matching and validation methods (Euzenat and Shvaiko, 2013).

A key reason is that stakeholders use different terminology and conceptualize and delineate assets differently, often referring to partially overlapping portions of the physical world. In this paper, *referent* denotes the underlying physical asset, space, or process, while modeled objects may differ slightly in scope across views (e.g., whether a door frame is considered part of the door) (Beck, 2023). Rather than enumerating all roles, we treat their information artifacts as view-based projections of this referent.

Figure 1 illustrates three recurring AECO views: a *design view* emphasizing geometry, types and instances, and design intent; a *checking view* restructuring information into rules, thresholds, and pass or fail outcomes; and an *operations view* focusing on maintainable assets, maintenance plans, and spare parts.

The central challenge is that these views can be internally consistent, yet difficult to link across contexts.

These view-specific representations differ in terminology, modeling intent, and information logic. Depending on the stakeholder task, the same referent may be represented as a geometric element, as the subject of rules and constraints, or as a maintainable asset with operational responsibilities. As a result, views systematically use different naming and classification conventions, delineate and group elements differently (in terms of granularity and decomposition), and encode information using different modeling constructs such as entities, attributes, relationships, or constraints, each optimized for a specific purpose rather than cross-view integration.

Across lifecycle phases, stakeholders need to reference and query the same assets consistently, for example, to ensure reliable handover, provide compliance evidence, and track the impact of design changes. Explicit mappings are therefore required to connect these heterogeneous representations. In practice, alignment is rarely just a matter of renaming labels or matching synonyms. Typical cases include entity-to-entity mappings, such as linking an IfcDoor element in a design model to the corresponding asset entry in a facility management system, such as a computer-aided facility management (CAFM) record. Another common case is an attribute-to-entity transformation, in which a fire rating expressed as a property is reinterpreted as an explicit requirement or compliance constraint. A further recurring case is many-to-one aggregation, in which several modeled doors are grouped into one operational set, such as a fire compartment boundary or a maintenance batch. These mapping types determine which semantic relations and transformations must be applied before data

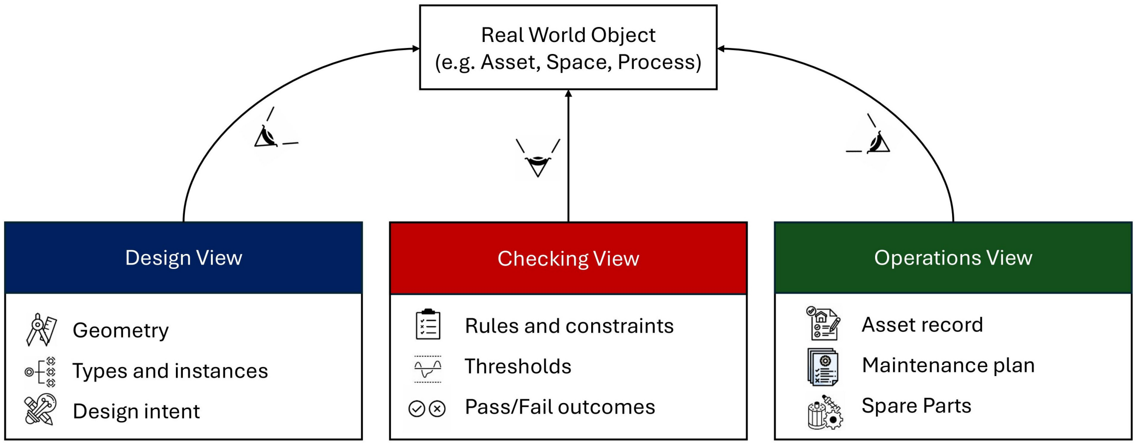

Figure 1: A shared physical referent is represented through different stakeholder views. The design view emphasizes geometry and design intent, the checking view encodes rules, thresholds, and pass/fail outcomes, and the operations view organizes information around maintainable assets. Meaning-preserving integration requires explicit semantic relations and transformations between views.

can be integrated or reasoned over across stakeholder systems.

Ontology matching approaches typically emphasize correspondences between entities or concepts. In AECO integration, link semantics are often context dependent and require interpreting modeling intent and scope differences between views; the core challenge is to determine the intended semantic relation and the required transformation for meaning-preserving integration (Beck, 2023).

This becomes evident when linking design models to downstream uses such as compliance checking and facility management handover. Although much of the required information may already be present, it often has to be restructured or made explicit to become usable in the target view. Even with IFC-based exchange, semantic ambiguity and alternative representation choices can create structural mismatches across federated models and stakeholder systems (Venugopal et al., 2015). More broadly, interoperability research in AECO/FM shows that data loss and communication problems persist across lifecycle exchanges despite the use of established standards (Ozturk, 2020).

This paper contributes a typology of cross-view mapping requirements as a preceding decision and structuring layer for cross-view alignment. It determines the mapping type, the intended semantic relation, and the required transformation between views, guiding candidate generation and validation. The goal is to reduce redundant information creation across disciplines by making explicit where information resides and how it can be referenced and transformed, so that it can be reused and delivered in the form each stakeholder expects. Rather than framing alignment as a uniform similarity-matching task, we derive mapping types that operate at different levels of granularity and semantic interpretation. Using representative examples from planning, construction, and operations, we characterize recurrent mismatch patterns and specify the semantic relations and transformations needed to resolve them, including direct entity correspondence, attribute-to-entity transformation, cluster-based mapping, granularity shift, and value or unit normalization. The contribution is twofold. First, the typology makes explicit which kind of mapping problem is present in a given integration scenario and, by doing so, clarifies the specific semantic relations and transformations that are required for meaning-preserving integration, rather than treating alignment as a uniform matching task. Second, the typology makes these distinctions actionable for AI-supported alignment workflows by specifying what evidence and mapping outputs are needed for candidate generation, validation, and reuse.

State of the Art

Prior work on ontology alignment has established how to define correspondences and transformations between heterogeneous representations (Portisch et al., 2022; Thiéblin et al., 2020). Ontology mapping and alignment research frames interoperability as the establishment of correspondences between entities in different models, enabling information to be interpreted, translated, and queried across sources (Abd Nikooie Pour et al., 2023). Such correspondences are not restricted to equivalence. They may encode subsumption or disjointness and often require transformations, for example, unit conversion, aggregation, or reinterpretation, to preserve semantics (Beck et al., 2021; Beck, 2023). In information exchange across AECO workflows, similar issues arise when stakeholders apply different boundaries and discipline-specific conventions, for example, gross versus net area definitions or properties expressed in different units, which cannot be resolved reliably through label similarity alone.

Matching approaches differ mainly in the information sources they use, for example, labels, structure, data instances, or formal semantics, and in how they combine them (Portisch et al., 2022; Abd Nikooie Pour et al., 2023). Across these approaches, candidate correspondences are typically treated as hypotheses whose usefulness depends on the application context, and many applications require user scrutiny before results are operationalized (Portisch et al., 2022; Beck et al., 2021).

Prior work on semantic interoperability in AECO indicates that lifecycle integration, meaning reuse, comparison, and translation of project information across design, construction, and operations views often require more than identifying similar terms (Beck et al., 2021). Ontology alignment research distinguishes between discovering candidate links and specifying mappings that preserve meaning under transformation, for example, when aggregation, decomposition, unit conversion, or reinterpretation are required (Thiéblin et al., 2020; Portisch et al., 2022).

In the AECO domain, semantic interoperability has been addressed through standards, ontologies, and linked data approaches, yet recurring gaps remain across lifecycle handovers. IFC provides a shared exchange schema, but IFC-based exchanges still exhibit variability in representation choices and a frequent reliance on property-set extensions, which can introduce ambiguity and structural mismatches across stakeholders and software toolchains (Venugopal et al., 2015). Semantic web research has therefore explored representing BIM data as RDF and OWL to enable explicit linking and cross-dataset querying, including ifcOWL as a representation that allows BIM graphs to be processed with ontology-based methods (Pauwels and Terkaj, 2016; Vilgertshofer et al., 2017). Lightweight ontologies, such as the Building Topology Ontology (BOT), provide shared structural primitives for linking elements, spaces, and zones to complementary sources, such as monitoring and operations data (Rasmussen et al., 2021). Complementary to such lightweight linking ontologies, modular ontology suites such as DiCon cover richer process and workflow semantics for digital construction and integration scenarios (Zheng et al., 2021). Linked data studies further show that interoperability can be achieved through explicit, maintained linking relations between datasets, provided that these relations are made machine-interpretable and that governance mechanisms are in place (Curry et al., 2013; Vilgertshofer et al., 2017).

Recent research operationalizes semantic enrichment and BIM knowledge graphs in concrete workflows, for example, for digital twin data integration and graph-based enrichment of BIM representations (Ramonell et al., 2023; Wang et al., 2024; Bloch et al., 2022). In parallel, machine-readable information requirements and automated checking approaches, such as the Information Delivery Specification (IDS) introduced and maintained by buildingSMART International, support systematic verification of required IFC information in practice (buildingSMART International, 2026; Kremer and Beetz, 2023; Fischer et al., 2024; Akbas et al., 2025). This line of work illustrates how view-specific requirements can be formalized, but it typically assumes that the required information is already structured in the target view and does not address how heterogeneous stakeholder representations should be related when transformations or grouping decisions are needed. However, recent reviews characterize this body of work as still fragmented and largely workflow- and method-driven, with limited consolidation of reusable patterns for relating heterogeneous stakeholder representations (Bloch et al., 2022). This motivates the development of an explicit mapping typology for recurrent cross-view mismatches in AECO.

The proposed typology complements rather than replaces ontology alignment and IDS-based checking. While ontology alignment provides general correspondence types and matching strategies, and IDS specifies target-side information requirements for validation or exchange, our contribution addresses the preceding AECO-specific task of determining what kind of cross-view mapping problem is present and which semantic relation and transformation are required for meaning-preserving integration. The typology therefore serves as a decision and structuring layer before candidate generation, rule execution, or downstream validation takes place. Based on this positioning, the following methodology derives the typology from recurrent cross-view scenarios and abstracts them into a set of mapping types.

Methodology

The State of the Art highlights that alignment problems often differ in the intended semantic relation and in the required transformation, and that these differences are frequently not captured when alignment is treated as a uniform matching task. Building on these observations, we systematize recurrent mismatch patterns into mapping types that make the required relation and transformation explicit.

The typology was derived through comparative analysis of recurrent cross-view integration scenarios involving design, checking, construction, and operations representations. In this paper, three lifecycle examples are discussed in detail, covering design-to-checking, design-to-construction, and design-to-operations transitions. These examples are complemented by additional recurrent cases reflected in the broader type definitions summarized in Table 1. The analyzed artifacts include BIM element representations, checking-oriented rule or constraint representations, construction-oriented grouping views, and operations-oriented asset records. Scenarios were selected deliberately to cover recurring handover and coordination situations, clearly different modeling intents across views, and distinct semantic relations or transformation requirements.

For each case, the source and target view were compared along five dimensions: information units, identifiers, boundaries, grouping, and representation form, meaning whether information is represented as explicit entities or encoded as attributes. A mapping type was assigned when a recurring combination of semantic relation and required transformation could be observed across the analyzed examples. For example, in the design-to-checking case, the door remains the same referent, but the checking view changes the representation context and reinterprets fire-rating information from an attribute into an explicit constraint object. This combination leads to direct entity correspondence for the door itself and attribute-to-entity transformation for the fire-rating requirement.

Accordingly, the typology is an analytical generalization derived from recurrent patterns in representative cases rather than from large-scale benchmarking, and it does not claim statistical completeness. In this paper, the most representative types are illustrated through concise lifecycle scenarios, while additional types are summarized in Table 1 to indicate the typology's broader applicability.

We refer to the source view as the representation from which information is taken, and to the target view as the representation to which data are mapped to satisfy the stated integration goal. These terms describe mapping direction only and do not assume a fixed lifecycle order.

Mapping Typology for Stakeholder Views

This section introduces a mapping typology that systematizes recurrent cross-view mismatches in AECO by the semantic relation and transformation required for meaning-preserving integration. To support practical use, Table 1 summarizes each mapping type as a recurring integration pattern. For each type, it identifies typical situations in stakeholder views that indicate the need for this mapping, and it specifies what kind of mapping result is required so that meaning is preserved in the target view.

Table 1 shows that resolving cross-view mismatches often requires structural rather than term-level alignment. Depending on the mapping type, integration requires identity linking, constraint lifting, operational grouping, granularity reconciliation, or value and unit normalization. The following scenarios illustrate three representative patterns, while granularity shift and value or unit normalization are summarized in Table 1.

Lifecycle Examples

This section shows how different stakeholder views lead to recurring mismatch patterns. It also shows how these patterns fit the typology categories defined in the Mapping Typology for Stakeholder Views section. Each example describes the source and target view, identifies the mismatch pattern, assigns a mapping type, and explains the semantic relation and transformation needed for actionable mapping across views.

Planning View

The planning view is contrasted with a compliance-based checking view to illustrate how a shared physical referent is conceptualized under different modeling intents. In the design view, the door is represented as a geometric building element, typically as an explicit object with shape, placement, and descriptive properties such as fire rating, acoustic rating, and clear opening width and height. In the compliance view, the focus is not only on the door itself but also on the rule context in which it participates. The door is considered an opening that must meet egress and fire safety standards. The checking workflow requires explicit constraints that are evaluated against the model. Figure 2 summarizes the mapping output structure required to connect these two representations.

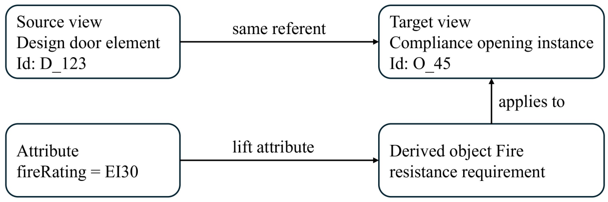

Figure 2: Planning pattern: identity link and attribute lifting.

There are two mismatch patterns. First, the door entity in the design view must be linked to the door or opening instance in the checking view. This is a direct entity correspondence, since both target the same referent but use different naming and referencing conventions and delineate the element differently due to modeling conventions. Second, the fire rating stored as a property in the design view must be an explicit constraint in the compliance view. This is an attribute-to-entity transformation. The target view requires a constraint object with explicit semantics, like a required fire resistance class and an explicit evaluation target.

A correct mapping output has two linked statements. The first links the design door entity to the checking door or opening instance using an equivalence relation. The second lifts the fire rating attribute into a compliance constraint object. It then links this constraint to the door instance as its evaluation target. The transformation requires normalizing the fire rating value and linking it with the correct rule context. Label similarity may help rank candidates. However, it is insufficient to infer the intended constraint semantics or the correct target mapping scope (which elements, at what level of granularity) without structural and contextual interpretation.

Construction View

To highlight phase-specific aggregation, the construction case relates a planning-oriented model view to a construction-oriented view centered on work packaging. The construction view organizes information by work packages, assemblies, and temporary states. In the construction view, the main units are often not single model elements. Instead, they are operational units for scheduling, procurement, and site work. Doors may be treated as installation packages, floor-based batches, or trade-specific scopes. These scopes combine multiple components and tasks. Figure 3 summarizes the cluster-based mapping relation needed to link element-level design representations to construction packages.

A common mismatch is a many-to-one operational grouping across views: multiple design elements correspond to a single construction concept, so a grouping transformation

Table 1: Mapping typology for stakeholder views and the semantic relations and transformations required for resolution.

Mapping type	**Definition**	**Typical trigger in stakeholder views**	**Required semantic relation**	**Required transformation**
Direct entity correspondence	A modeled entity in one view refers to the same real-world referent as an entity in another view, but identifiers, boundaries, or context differ. This type applies when the integration goal requires cross-view identity or traceability.	Cross-phase linking where the same object is represented in design and operations, for example, a design element versus an FM asset record for the same referent.	Equivalence, or a same-referent link, between entities.	Identifier linking and normalization, optional projection of relevant attributes, and provenance capture for traceability. Label similarity can support ranking, but it cannot establish identity when identifiers and scopes differ.
Attribute-to-entity transformation	Information encoded as an attribute in the source view must be lifted into an explicit object in the target view because the target view represents it as a requirement or constraint to be evaluated or reasoned over. This type applies when the target contains an explicit constraint object that is not present in the source and must be derived from values and context.	A property is descriptive in one view but normative in another, for example, a performance property in design that must become an explicit constraint in a checking view.	Constraint on, requirement for, or governs relation linking the derived constraint object to the relevant entity.	Attribute lifting into an explicit object, value parsing and normalization, interpretation of units and thresholds, and linkage to the checked entities. Label similarity alone cannot determine the target object type and its semantics.
Cluster-based mapping	A set of elements in one view corresponds to a single concept or unit in another view, where the target concept is defined operationally or functionally. Assign this type when the target concept is defined by membership across multiple source elements and the boundary is not explicitly defined in the source view.	Operational groupings are not explicit in the source view, for example, grouping elements into a maintenance bundle, an operational zone, or a compartment-related set.	Aggregation, part of, or member of a relation linking source elements to the target cluster concept.	Grouping and aggregation based on spatial, functional, or rule-based criteria, with optional creation of a derived target object representing the cluster. Label similarity alone cannot determine membership and boundary conditions.
Granularity-shift mapping	An entity in one view corresponds to multiple entities or a composed structure in another view because stakeholders model at different levels of granularity. Use this type when integration requires decomposition or composition across a part hierarchy, or alignment between type-level and instance-level representations.	Design uses type definitions or assemblies, while construction and operations use instances, components, or maintainable parts, for example, a type definition mapped to multiple installed assets, or an assembly mapped to its components.	Part of, composed of, or an instance of relations connecting coarse and fine-grained representations.	Decomposition or composition, propagation of attributes across levels, consistency checks to preserve totals, constraints, and traceability across levels. Label similarity alone cannot infer the correct hierarchy level.
Value and unit normalization mapping	The same intended property exists in both views, but its value semantics differ due to units, reference conditions, calculation rules, rounding conventions, or code lists. Use this type when the mapping challenge is primarily at the value level rather than the entity level.	Cross-domain exchanges of quantities and performance values, for example, gross versus net area conventions, thermal or acoustic ratings expressed under different standards, or values encoded as codes in one view and numeric values in another.	Same property relation with an explicit transformation function, or equivalence under conversion.	Unit conversion, normalization to reference conditions, code list translation, optional recomputation from available parameters, and validation against allowed ranges or constraints. Label similarity alone cannot guarantee comparable values.

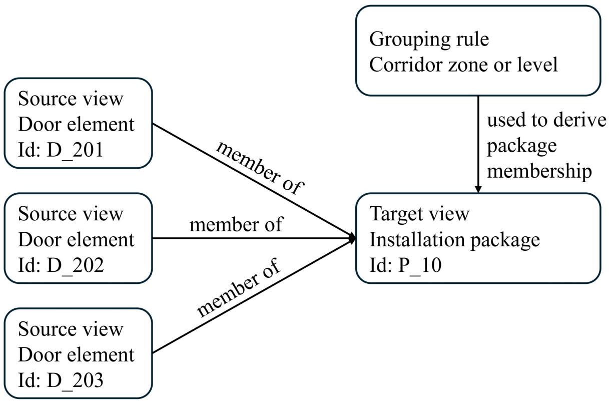

Figure 3: Construction pattern: clustering elements into an installation package.

is required. For instance, a set of doors in a corridor zone may be installed by a single subcontractor during a single scheduled activity. As a result, these doors form a single operational package in the construction view. The design view provides multiple distinct door entities, each with its own properties. In contrast, the construction view groups them as a package entity with references to its members and added attributes, such as planned start date, responsible trade, and installation status.

This mismatch is a cluster-based mapping. Here, the target concept is defined operationally, and its boundary is not clear in the design view. The transformation requires creating a group concept and assigning member relations from each door entity to the package. A correct mapping output includes a construction package object with a list of source doors and a grouping rule. The grouping rule can use spatial zone, level, or schedule logic. The mapping must keep the original door identifiers and introduce a stable identifier for the package. This allows construction reporting and later handover to reference both the group and the elements across phases.

Operations View

The operations case shows how a design representation is reinterpreted in facility management, where a door is managed as a maintainable asset linked to responsibilities and requirements such as maintenance and inspection. In the design view, the door exists as a building element instance with geometric and specification properties. In the operations view, the door becomes an asset record within an equipment register, linked to maintenance plans, spare parts, inspection intervals, warranty information, and potentially sensor or access control system identifiers. Figure 4 summarizes the mapping output structure needed to preserve traceability between the design element, the FM asset record, and operational groupings such as maintenance batches.

Two mismatch patterns often occur. First, the door entity must link to its asset record in the FM system. This is a direct entity correspondence. Both target the same physical door but use different naming and referencing conventions. They may also use different boundaries. For example, an asset record may include both the door leaf and hardware, while the design model separates these components. Second, FM workflows often create groupings that do not exist in the design. Doors may be managed as a maintenance batch, such as all doors of a given type on a floor, or as an operational set, such as a fire compartment boundary. This results in a cluster-based mapping.

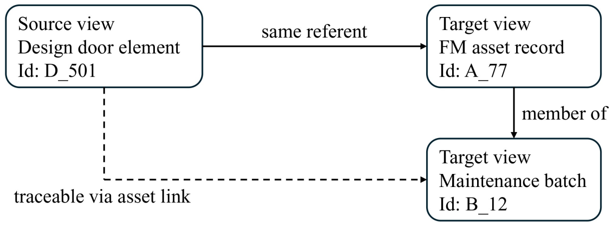

Figure 4: Operations pattern: identity link and operational grouping.

A correct mapping output must keep traceability between views. The output links each design door to its FM asset record and notes which identifiers each side uses. For maintenance batches, the output adds a grouping concept and assigns membership at the asset level. This lets FM tasks be managed at the batch level without losing links to the original elements. The transformation needs identifier reconciliation, optional aggregation rules, and preservation of essential attributes. Key attributes include location, type classification, fire rating, and manufacturer data if available.

Implications for AI-Supported Alignment

Our contribution lies in a mapping typology that provides a preceding decision and structuring layer for AI-assisted alignment workflows connecting different stakeholder views. By determining the mapping type, the intended semantic relation, and the required transformation, it constrains candidate generation and validation.

For direct entity correspondence, the main objective is to resolve identity across differing identifiers, modeling conventions, and element boundaries. For attribute-to-entity transformation, the workflow must detect when a descriptive property needs to be reinterpreted as an explicit requirement or constraint object in the target view. For cluster-based mapping, the workflow must recognize when multiple source elements collectively correspond to an operational concept and must make the grouping boundary and membership relations explicit. Granularity-shift mapping requires reconciling representations across levels of detail, and value and unit normalization mapping requires explicit conversion or normalization logic.

This process, therefore, requires type-specific training and reference data to support both automated candidate generation and subsequent validation. Lexical similarity and terminology resources are useful for candidate retrieval and ranking, but they are rarely sufficient when boundary decisions or semantic transformation is involved. For direct entity correspondence, robust evidence is typically provided by persistent identifiers where available, geometric and spatial consistency, containment relations, and stable type information. For attribute-to-entity transformation, relevant cues include requirement-oriented vocabulary, normative contexts, and value patterns that encode classes, thresholds, or compliance-relevant categories. Cluster-based mapping benefits from shared context and operational metadata, for example, co-location within a zone, shared classification, or explicit grouping rules derived from scheduling logic, inspection regimes, or maintenance strategies.

Human supervision remains essential because mapping correctness is context-dependent and directly affects lifecycle tasks. Validation effort should be prioritized for cases with ambiguous identity evidence, uncertain attribute interpretation, or unclear grouping boundaries. Once validated, these cases can be stored as reference mappings. Over time, curated reference sets enable calibration of type detection, support more reliable candidate ranking, and reduce recurring manual effort in comparable scenarios.

The typology also provides a concrete basis for specifying the minimum mapping outputs that an AI-assisted workflow must produce. Each mapping should record its mapping type, identify the linked source and target information units, and make the intended semantic relation explicit. When transformation is required, the aggregation rule or attribute-lifting logic should be represented in a reproducible form. Provenance metadata and a confidence estimate should be included to support auditability and to enable revision when models, standards, or organizational conventions evolve.

Discussion

Lessons Learned

The lifecycle examples show that semantic heterogeneity across stakeholders is driven not only by terminology, but also by representational differences. Even when referring to the same referent, views vary in what is modeled as an entity versus an attribute, and in how boundaries and groupings are defined. Therefore, interoperability often requires mappings beyond term equivalence, including transformation or aggregation logic.

Direct entity correspondence requires identity resolution across inconsistent identifiers and boundary conventions. Same-referent cases do not necessarily imply strict equivalence, because different views may adopt different boundaries or decomposition logics for the same physical asset. In such cases, the typology helps distinguish direct correspondence from granularity shift, for example when one view represents a door as a single object while another distinguishes frame, leaf, and hardware. To preserve traceability, such mappings should record provenance and, where ambiguity remains, an explicit boundary assumption and review status. Attribute-to-entity transformation requires normative interpretation, turning a descriptive value into an explicit requirement or constraint representation. Cluster-based mapping requires membership and boundary decisions that are often organization-specific and hard to infer reliably.

For practice, the typology suggests classifying the mapping need first, because evidence, validation effort, and maintenance burden differ by type. For standardization, it supports guidance on stable identifiers, operational groupings, and explicit requirement representations.

Limitations

The typology is scoped to recurrent cross-view mismatch patterns illustrated through representative examples from planning, construction, and operations and does not claim exhaustive coverage of AECO interoperability situations. Geometric conversion, coordinate reference transformations, and purely syntactic schema translation are outside scope.

Because mapping needs and boundaries depend on project context, organizational conventions, and metadata quality, some cases remain ambiguous, especially where identifiers are inconsistent or decomposition differs across views. The present paper establishes the typology through analytical comparison of representative cases rather than through a large annotated benchmark. Accordingly, it does not yet report inter-rater agreement, quantitative coverage, or downstream tool-performance evaluation. Its contribution is to provide a structured basis for distinguishing recurrent semantic relations and transformation requirements in AECO view alignment.

An important next validation step is therefore to test whether different analysts consistently assign the same mapping type, semantic relation, and transformation requirement to the same case, particularly in boundary-sensitive situations involving decomposition, grouping, or partial overlap of referents. This is especially relevant because the typology is intended to structure downstream AI-supported candidate generation, validation, and traceability.

Conclusion

AECO stakeholders represent the same real-world assets, spaces, and processes through different terminologies, conventions, and abstraction levels. As a result, structural mismatches persist even with IFC and cannot be resolved reliably through label matching alone.

This paper proposed a mapping typology that characterizes recurrent cross-view mismatch patterns across planning, construction, and operations and makes explicit the semantic relations and transformations required to resolve them. By turning alignment into an explicit decision layer, the typology supports more systematic interoperability practice and provides a foundation for AI-supported candidate generation and validation.

Acknowledgments

This research was supported by the Nemetschek Innovation Foundation and benefited from collaboration with industry partners within the framework of the openDBL project. The openDBL project is funded by the European Union's Horizon Europe research and innovation program under grant agreement No. 101092161, providing valuable test cases during development.

References

Abd Nikooie Pour, M. et al. (2023). Results of the ontology alignment evaluation initiative 2023. In Proceedings of the 18th International Workshop on Ontology Matching (OM 2023), volume 3591. CEUR Workshop Proceedings.

Akbas, B., Idour, A., Isikdag, U., Hammad, A., and Bonneville, C. (2025). A holistic approach to information requirements: Integration of level of information need and information delivery specification. Journal of Information Technology in Construction (ITcon), 30:731–744.

Beck, S. F. (2023). Context-sensitive linking of heterogeneous information models from the building and the urban domain. PhD thesis, Technical University of Munich.

Beck, S. F., Abualdenien, J., and Borrmann, A. (2021). An evaluation of the strict semantics of owl:sameas in the field of bim-gis integration. In Proceedings of the 9th Linked Data in Architecture and Construction Workshop (LDAC2021), volume 3081 of CEUR Workshop Proceedings.

Bloch, T., Sacks, R., Pollmann, C., and Borrmann, A. (2022). Connecting research on semantic enrichment of bim: review of approaches, process, main principles, goals, methods and associated concepts. Journal of Information Technology in Construction (ITcon), 27:416–440.

buildingSMART International (2026). Information delivery specification (ids) project page. Accessed 2026-01-14.

Curry, E., O'Donnell, J., Corry, E., Hasan, S., Keane, M., and O'Riain, S. (2013). Linking building data in the cloud: Integrating cross-domain building data using linked data. Advanced Engineering Informatics, 27(2):206–219.

Euzenat, J. and Shvaiko, P. (2013). Ontology matching: Second edition. Springer, Berlin, Heidelberg, 2 edition.

Fischer, S., Urban, H., Schranz, C., Loibl, P., and van Berlo, L. (2024). Extending information delivery specifications for digital building permit requirements. Developments in the Built Environment, 20:100560.

FMI (2021). Harnessing the data advantage in engineering and construction.

Kremer, N. and Beetz, J. (2023). Extending information delivery specification for linking distributed model checking services. In Proceedings of the 2023 European Conference on Computing in Construction (EC3 2023) and the 40th International CIB W78 Conference, Heraklion, Crete, Greece.

Ozturk, G. B. (2020). Interoperability in building information modeling for aeco/fm industry. Automation in Construction, 113:103122.

Pauwels, P. and Terkaj, W. (2016). Express to owl for construction industry: Towards a recommendable ifcowl ontology. Automation in Construction, 63:100–133.

Pauwels, P., Zhang, S., and Lee, Y.-C. (2017). Semantic web technologies in aec industry: A literature overview. Automation in Construction, 73:145–165.

Portisch, J., Hladik, M., and Paulheim, H. (2022). Background knowledge in ontology matching: A survey. Semantic Web.

Ramonell, C., Chacón, R., and Posada, H. (2023). Knowledge graph-based data integration system for digital twins of built assets. Automation in Construction, 156:105109.

Rasmussen, M. H., Lefrançois, M., Schneider, G. F., and Pauwels, P. (2021). Bot: The building topology ontology of the w3c linked building data group. Semantic Web, 12(1):143–161.

Thiéblin, É., Haemmerlé, O., Hernandez, N., and Trojahn, C. (2020). Survey on complex ontology matching. Semantic Web, 11(4):689–727.

Venugopal, M., Eastman, C., and Teizer, J. (2015). Ontology-based analysis of the industry foundation class schema for building information model exchanges. Advanced Engineering Informatics, 29(4):940–957.

Vilgertshofer, S., Amann, J., Willenborg, B., Borrmann, A., and Kolbe, T. H. (2017). Linking bim and gis models in infrastructure by example of ifc and citygml. In Computing in Civil Engineering. TUM CMS related work, includes linked data based combined querying of IFC and CityGML.

Wang, Z., Sacks, R., Ouyang, B., Ying, H., and Borrmann, A. (2024). A framework for generic semantic enrichment of bim models. Journal of Computing in Civil Engineering, 38(1):04023038.

Zheng, Y., Törmä, S., and Seppänen, O. (2021). A shared ontology suite for digital construction workflow. Automation in Construction, 132:103930.