




Article

Temporally Qualified Building Elements: A DOLCE-Based Ontology for Phase-Dependent Identity and Change Tracking in BIM Models

Andrzej Szymon Borkowski ^{1,*} , Paulina Jarema ¹ , Magdalena Kładz ¹  and Anatolii Smoliar ²

¹ Faculty of Geodesy and Cartography, Warsaw University of Technology, Politechniki Square 1, 00-661 Warsaw, Poland; paulina.jarema.stud@pw.edu.pl (P.J.); magdalena.kladz.dokt@pw.edu.pl (M.K.)

² Department of Industrial and Civil Engineering, Cherkasy State Technological University, 18006 Cherkasy, Ukraine; a.smoliar@chdtu.edu.ua

* Correspondence: andrzej.borkowski@pw.edu.pl

Abstract

Building Information Modeling (BIM) usually represents a building as a static snapshot of the model's state. Dynamic extensions, such as Internet of Things (IoT)-enabled sensing or immersive visualization, already exist, but the underlying data model remains state-based. The Industry Foundation Classes (IFC) standard does not define a formal mechanism that would link the same physical element across successive phases of a building's life cycle. Design, construction, and operation are recorded in separate IFC files, and the same element is assigned different Globally Unique Identifiers (GUIDs) in each. The result is fragmentation of the element's identity, loss of the history of property changes, and the inability to formulate cross-phase queries. This paper proposes the BIM-Phase ontology based on the fundamental Descriptive Ontology for Linguistic and Cognitive Engineering (DOLCE) ontology, which solves this problem by introducing a distinction between a building element as an enduring and its life cycle phases as perdurants. The ontology comprises nine classes, six object relations, and six axioms expressed in Web Ontology Language 2 Description Logic (OWL 2 DL). Phase properties and relations are represented using a reification pattern, which maintains full compatibility with the expressiveness of OWL 2 DL. The ontology was validated using an example of a single-family residential building developed in Autodesk Revit. Three structural elements (external wall, floor slab, and column) were tracked across three phases of the life cycle. Eight competency questions covering scalar, constitutional, and mereological changes were defined and mapped to ontology constructs, confirming that the BIM-Phase enables the recording of changes and the formulation of cross-phase queries that are impossible in classic IFC. All eight questions were answered correctly on the published knowledge graph, and the Hermit reasoner confirmed the logical consistency of the model. The findings show that preserving element identity across phases requires only a minimal ontological layer on top of existing standards. We recommend introducing persistent, phase-independent identifiers of building elements alongside IFC GUIDs, as this single change enables full lifecycle change tracking.



Academic Editor: Ioannis K. Brilakis

Received: 18 May 2026

Revised: 1 July 2026

Accepted: 3 July 2026

Published: 6 July 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](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: BIM (Building Information Modeling); IFC (Industry Foundation Classes); ontology; DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering); building lifecycle; temporal modeling; phase-based modeling; semantic interoperability; change tracking; identity preservation; reification; event-based modeling; digital twin

1. Introduction

Building Information Modeling (BIM) assumes that a digital model of a building provides a reliable representation of reality at every stage of its life cycle—from design concept through construction to operation and demolition [1]. In practice, the IFC (Industry Foundation Classes) data schema, which serves as the fundamental open standard for information exchange in the openBIM approach, treats a building element as a static snapshot of a given state. An entity (physical or conceptual object) such as `IfcWall` stores a set of properties, geometry, material, and classification, all referring to a single, unspecified moment in time, without a formal mechanism to express that the same wall existed previously in a different configuration and will exist in yet another in the future [2]. The consequences of this limitation may be significant for design and construction practice. First, at the level of object identity, since, in a typical BIM project separate, IFC files are created for individual phases (design model, as-built model, and operational model), in which the same physical element may have different GUIDs (Globally Unique Identifiers) and differ in geometry or attributes (Figure 1), there is no formal relationship establishing that they refer to the same entity. Second, at the level of change tracking, i.e., when an element’s property is modified (e.g., the thickness of thermal insulation is changed during construction), the new value overwrites the previous one, and the change history is lost or, at best, stored in external document management systems (e.g., in an Electronic Document Management System—EDMS), outside the semantic scope of the model. Third, at the level of cross-phase inference, it is not possible to formulate a semantic query such as, for example, “which structural elements changed their constituent material between the design and as-built phases?” This is because the IFC schema lacks a conceptual framework linking the states of the same object at different points in time.

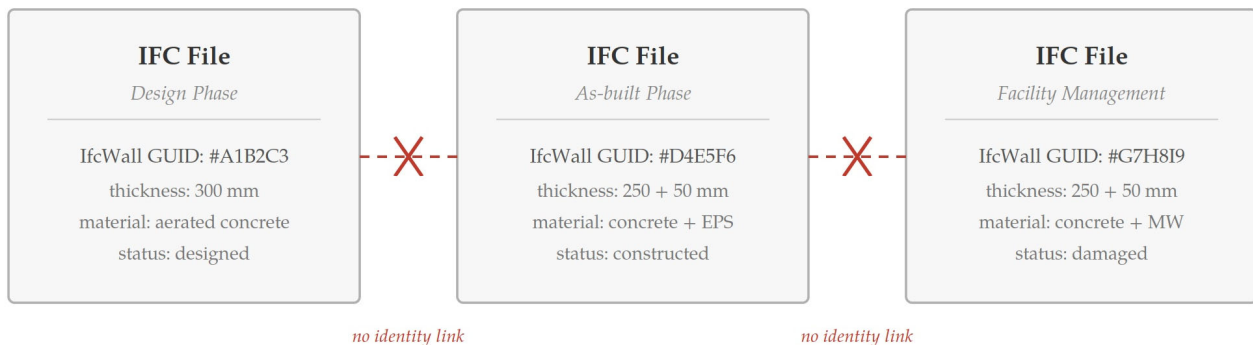


Figure 1. Current IFC-based approach to building element lifecycle representation: identity fragmentation across project phases. Source: own elaboration.

This problem is not merely a technical inconvenience in data exchange. It reflects a deeper ontological deficiency, as the IFC does not formally distinguish between an object and its state at a given phase, between the duration of an entity and the sequence of its configurations, or between identity and change [3]. In the terminology of formal ontology, this means that the IFC schema treats building elements as so-called *endurants* (entities fully present at every moment of their existence) but offers no axiomatics allowing for the modeling of their temporal phases, time-varying relations, or temporally qualified part-whole relations. The DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) ontology formalizes these distinctions through the apparatus of temporary parthood (a temporally indexed part-whole relation) and constitution as a relation between endurants that are spatially and temporally co-located (Figure 2) but distinguishable in terms of the conditions of their duration [3]. Existing semantic extensions of IFC, such as `ifcOWL` or the ontologies of the Linked Building Data stack (in particular, the Building

Topology Ontology), inherit the temporal limitation of the source schema, focusing on a faithful translation of the EXPRESS structure into Web Ontology Language (OWL) without introducing an additional layer of change modeling [2,4].

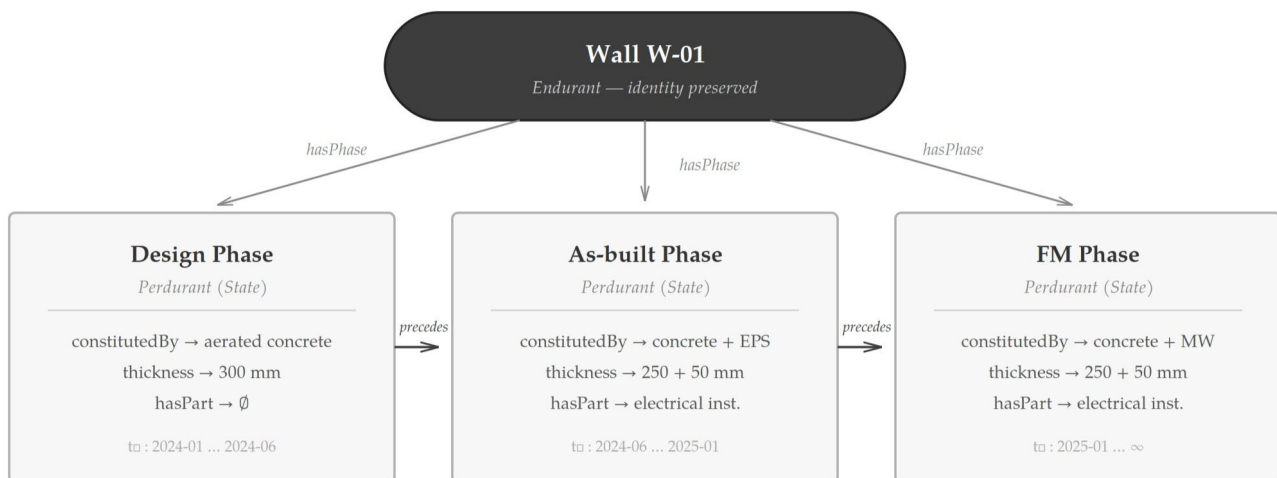


Figure 2. Proposed DOLCE-based ontological approach. Source: own elaboration.

Thus, this article posits that effective management of information about a building object throughout its full life cycle requires a formal theory of change—one that allows a building element to be treated as an entity that remains identical to itself over time while simultaneously passing through distinct, ontologically characterized phases, each of which may be associated with different properties, a different material structure, and different meronomic relations.

This postulate must be positioned against the current state of research. The integration of the IFC schema with formal ontologies has been the subject of intensive research for at least a decade. In a bibliometric review covering 122 publications from 2011 to 2023, Jia et al. identified four mature application areas for the integration of IFC with ontologies: (i) regulatory compliance, (ii) design knowledge management, (iii) interoperability support, and (iv) semantic enrichment of models [5]. At the same time, the authors noted that the vast majority of studies focus on individual life cycle phases, most often the design phase, while only a few address the issue of semantic continuity between phases. None of the analyzed publications proposed a formal axiomatics of a building object’s identity over time based on a fundamental ontology. Concurrently, research is being conducted on the semantic enrichment of BIM models [6], aimed at supplementing missing information in IFC models using rule-based or machine learning methods. In a systematic review of this research trend, Bloch pointed out that existing approaches to semantic enrichment operate exclusively within a single state of the model, classify objects, supplement attributes, and detect topological relationships, but they do not account for the temporal dimension—that is, they do not model the fact that the same object passes through successive states in a building’s life cycle [7]. In other words, semantic enrichment enriches a snapshot of a given state, but it does not link snapshots into a coherent narrative of existence unfolding over time.

An analysis of the current state of knowledge and technology thus allows us to identify a research gap at the intersection of three areas. First, fundamental ontologies, particularly DOLCE, offer a mature framework for formalizing change, temporary parthood relations, temporally indexed constitution, and the phasicity of endurants; however, this framework has not yet been applied in the BIM domain. Second, existing construction ontologies (ifcOWL, BOT, and PROPS) faithfully map the structure of the IFC schema, but they inherit

its phase limitation by treating relationships between objects as atemporal. Third, work on semantic enrichment and on the integration of IFC with ontologies does not address the problem of object identity over time, focusing instead on enriching individual model states without a mechanism for linking them.

Thus, this work attempts to fill the identified research gap by proposing a so-called phase ontology (BIM-Phase), which formalizes temporally qualified relationships between building elements and their states in specific phases of the life cycle, embedding them within the conceptual framework of the DOLCE ontology.

Accordingly, the main objective of this work is to develop an ontological framework that enables a coherent representation of building elements throughout the full life cycle of a structure. This representation should preserve the identity of the element while explicitly modeling changes in its properties over time.

To achieve this goal, the work adopts the conceptual foundation of the DOLCE ontology and proposes a phase-based approach. In this approach, building elements are treated as durants associated with life cycle phases. These phases are temporal states. This approach allows for separating the object's identity from its phase-dependent properties and relationships. Thus, this work addresses the limitations of the IFC schema regarding temporal continuity, which were described earlier in this section.

The proposed approach is operationalized through the following research questions:

RQ1. How can the identity of a building element persisting over time be formally represented in such a way as to distinguish the object itself from its state in a given phase of the life cycle?

RQ2. How can scalar properties, material composition, and meronomic relations be modeled in Web Ontology Language 2 Description Logic (OWL 2 DL) as phase-dependent without violating logical decidability?

RQ3. How can transitions between life cycle phases be represented to enable reasoning about change as a process rather than merely as a difference between states?

In response to these questions, this article makes the following five original contributions:

(1) Formalization of Phase-Dependent Identity: Building elements are modeled as durants, and the phases of their life cycle are modeled as state-based perdurants. This allows for a clear separation of an object's identity from its temporal states. This contribution answers RQ1 and addresses the first component of the research gap identified in this section.

(2) Reification of Phase-Qualified Relations: The BIM-Phase ontology uses a reification pattern to represent properties, material constitution, and meronomic relations as phase-dependent constructs. This solution is compliant with OWL 2 DL and enables semantic querying using SPARQL. This contribution answers RQ2 and addresses the second component of the research gap.

(3) A Mechanism for Change Tracking and Cross-Phase Reasoning: Linking properties to specific life cycle phases enables explicit recording of changes. It also allows for queries comparing the states of the same object across different phases. This contribution addresses the third component of the research gap, namely the lack of a mechanism for combining model states into a coherent representation of a time-extending entity.

(4) Event-Based Extension of the Phase Model: This work proposes an event layer in which transitions between life cycle phases are modeled as perdurants (PhaseTransition-Event). This layer is a first-class, modular extension of the proposed framework rather than a peripheral add-on, and it is kept separate from the base ontology to preserve applicability to scenarios that require only phase-state representations (see Section 3.4). It allows change to be interpreted not only as a difference between states but as a process with a specific causal structure. This contribution addresses RQ3.

(5) Validation via a Case Study: The framework was verified on a model of a single-family residential building developed in Autodesk Revit 2026.2. This study covers three representative types of phase changes: changes in scalar properties, changes in material composition, and changes in meronomic relations. This contribution provides proof of the feasibility of the proposed approach and highlights its applications in areas such as digital twins and life cycle analytics.

The remainder of this article is organized as follows. Section 2 presents the theoretical background, with particular emphasis on the conceptual framework of the DOLCE ontology. Section 3 describes the structure of the BIM-Phase ontology and the design decisions made. Section 4 discusses the implementation, a case study, and competency questions. Section 5 contains an evaluation, a discussion of limitations, and directions for further work.

2. Theoretical Background

2.1. Formal Ontology: Endurants, Perdurants, and Temporal Qualification in DOLCE

DOLCE (Descriptive Ontology for Linguistic and Cognitive Engineering) is a fundamental ontology for which its central distinction is the division of entities into endurants and perdurants [3]. Endurants are physical objects or persons that are fully present at every moment of their existence; for example, a wall exists in its entirety both during the design phase and the operational phase, regardless of changes over time (e.g., changes in insulation). Perdurants, on the other hand, are events, processes, or states that extend over time and possess temporal parts: The design phase of a wall is a temporal part of its history, not a part of the wall itself. This distinction has fundamental consequences for modeling changes: An endurant can change its properties over time (changes in thickness, material, and meronomic relations) while remaining the same entity, whereas each configuration of properties at a given moment constitutes a separate perdurant, i.e., a state (DOLCE:State) of that endurant. DOLCE formalizes the relations between endurants and their states using three mechanisms. First, the constitution relation links the endurant to the matter of which it is composed at a given moment, since the same wall may be constituted by aerated concrete at time t_1 and by insulated concrete at time t_2 , without losing its identity. Second, temporal qualification assigns properties not directly to the endurant but to the pair “⟨endurant, time⟩”, which eliminates contradictions arising from changes in the value of the same property. Third, temporal parthood allows for the modeling of variable meronomic composition, meaning that an element can gain or lose parts over time, and each configuration is temporally indexed [3,8].

The choice of a foundational ontology is not a neutral decision. The three most widely adopted top-level ontologies are BFO (Basic Formal Ontology) [9], DOLCE [3], and UFO (Unified Foundational Ontology) [10]. BFO is built on the distinction between continuants and occurrents. It is standardized and dominates in biomedical applications. However, BFO is intentionally minimalistic. It does not provide a built-in mechanism for the phasicity of continuants. It also lacks a constitution relation between spatially co-located objects. Modeling phase-dependent identity in BFO would therefore require substantial domain-level extensions. UFO offers a rich apparatus of phases, roles, and relators. In this respect, it is conceptually well suited to lifecycle modeling. Its main limitation is practical. The reference implementation of UFO in OWL, known as gUFO, is less mature and less extensively axiomatized than DOLCEbasicOWL. DOLCE occupies a middle position. It provides explicit theories of constitution, temporary parthood, and temporal qualification. These theories directly address the problem of change in BIM, as described in Section 1. Moreover, the OWL 2 serialization of DOLCE published in 2024 provides verified reification patterns that can be reused directly [8]. For these reasons, DOLCE was selected as the foundation of the proposed ontology. This choice involves a trade-off. DOLCE is descriptive

and cognitively oriented, and its full axiomatization exceeds the expressiveness of OWL 2 DL. Section 3.1 explains how this trade-off was managed in the design of BIM-Phase.

Porello et al. [8] proposed a serialization of the DOLCE core in OWL 2 (DOLCEbasicOWL) using a reification pattern—that is, temporally qualified relations are represented as instances of intermediate classes, linked by binary relations to the enduring, perdurant, and value. This pattern, consistent with the W3C recommendation on n-ary relations, forms the direct basis for the BIM-Phase architecture presented in Section 3.

The DOLCE distinctions described above have direct practical implications for BIM implementation. First, the concept of an enduring maps onto a persistent identifier of a physical building element. Such an identifier can remain stable across IFC files exported in different phases. Second, the concept of a state maps onto a lifecycle phase of an element. Each phase can be linked to a specific IFC file or model snapshot. Third, the constitution relation provides a formal way to record the replacement of materials during construction or renovation. Fourth, temporal qualification allows different values of the same property to coexist in one knowledge graph without logical contradiction. For example, a wall can have a thickness of 40 cm in the design phase and 42 cm in the as-built phase. Both values remain queryable. In practical terms, these mechanisms enable change auditing, as-built verification, and the maintenance of element history in digital twin environments. Section 3 shows how these concepts are operationalized in OWL 2 DL.

DOLCE offers constitution and temporary parthood as native primitives, so phase-dependent identity is modeled without custom domain axioms, at the cost of the reification verbosity discussed in Section 5.

2.2. BIM Data Ecosystem: IFC, ifcOWL, and Linked Building Data

The independent IFC format is an open data exchange standard for the construction industry, defining an object schema in the EXPRESS language. The IFC4x3 schema describes nearly nine hundred entity classes, ranging from structural elements (IfcWall, IfcSlab, and IfcColumn) through spatial and functional relationships to property sets (PropertySets), but treats each IFC file as an independent snapshot of a given model state. There is no formal mechanism linking the same physical element represented in two IFC files created at different phases of the life cycle, as each instance receives its own GUID, and the specification does not define cross-file identity relationships [1].

The transfer of the IFC schema to the Semantic Web environment has resulted in several complementary ontologies. ifcOWL [2] is a faithful translation of the EXPRESS schema into OWL, preserving the full structure of IFC classes and relations and thus inheriting its temporal limitation. In response to the complexity of ifcOWL (tens of thousands of axioms), the W3C Linked Building Data Community developed lightweight modular ontologies: BOT (Building Topology Ontology) [4] models building topology (zones, spaces, and elements), and PROPS (Properties Ontology) provides representation patterns for properties. These ontologies form the Linked Building Data ecosystem, in which the building knowledge graph is created by integrating modules using OWL imports and shared URIs (Uniform Resource Identifiers) [11].

The interoperability challenges in this ecosystem are not limited to schema translation. They have at least four technical sources. First, the EXPRESS language and OWL differ in their semantic assumptions. EXPRESS uses a closed-world assumption and procedural validation rules. OWL uses an open-world assumption and description logic. A faithful translation such as ifcOWL therefore produces tens of thousands of axioms and ordered list constructs that are computationally expensive for reasoners [2]. Second, IFC property sets are late-bound. Properties are attached to objects through IfcRelDefinesByProperties relations and string-based names. This design prevents static typing of properties at the

schema level and complicates query formulation. Third, geometry in IFC is encoded procedurally. Boundary representations and swept solids do not map naturally onto RDF triples. Most Linked Building Data pipelines therefore separate the geometric layer from the semantic layer. Fourth, element identity is unstable. GUIDs are guaranteed to persist only within a single authoring environment. Export, import, and re-export operations frequently regenerate identifiers. This last problem is the direct concern of the present work.

A more detailed comparison of existing semantic BIM frameworks reveals differences in scope and in temporal capabilities. The closest existing approach to the problem addressed in this paper is OPM (Ontology for Property Management) [12]. OPM extends the SEAS ontology and represents property values as property states. Each state can carry provenance metadata and a timestamp. OPM therefore supports the versioning of individual property values. However, OPM operates at the level of single-property assignments. It does not model the identity of an element across separate IFC files. It does not treat a lifecycle phase as a first-class entity. It also does not cover changes in material constitution or in mereological structure. Moreover, OPM is not grounded in a foundational ontology. The ontological status of a property state therefore remains informal. Table 1 summarizes the comparison of the discussed frameworks.

Table 1. Comparison of semantic BIM frameworks with respect to temporal and identity-related capabilities. Source: own elaboration.

Framework	Scope	Temporal Support	Identity Across Phases	Foundational Grounding
ifcOWL [2]	Full translation of the IFC schema	None, single snapshot	No	No
BOT [4]	Building topology	None	No	Declared alignment with DOLCE Ultralite
PROPS	Property representation	None, new value overwrites old	No	No
OPM [12]	Property states with provenance	Versioning of property values	No	No
BIM-Phase (this work)	Element identity and phase qualified properties and relations	Discrete lifecycle phases	Yes	DOLCE

A key observation is that none of the ontologies mentioned—neither ifcOWL, BOT, nor PROPS—introduces a temporal dimension at the element identity level. BOT declares compatibility with DOLCE Ultralite but does not implement a mechanism for phases or constitutions. PROPS allows property values to be assigned to an element but without temporal indexing; thus, a new value replaces the old one, and the history is lost. The Linked Building Data ecosystem therefore provides a rich infrastructure for representing a building at a single point in time, but it does not offer a formal mechanism for tracking the same element across multiple phases of its lifecycle. Recent developments confirm the relevance of this gap. Research on federated common data environments, such as the ConSolid ecosystem, shows that multi-stakeholder projects require stable and dereference-

able identities of building objects across heterogeneous data sources [13]. Research on construction digital twins points in the same direction. A digital twin must maintain a persistent link between a physical asset and its evolving digital representation throughout the whole lifecycle [14]. Operational applications already combine building knowledge graphs with live data streams, for example, in the semantic control of building systems [15]. All these directions presuppose a capability that current Linked Building Data ontologies do not provide. They require a formal account of element identity over time. Filling this gap is the goal of the BIM-Phase ontology.

2.3. State of the Art: Temporal and Lifecycle Modeling in AEC Ontologies

Existing approaches to modeling the temporal dimension in AEC ontologies can be divided into three groups. The first consists of IFC extensions with a 4D (time) dimension, focusing on construction scheduling, which link model elements to tasks in the schedule and model execution time but do not track changes in an element's identity or properties between phases. The second group consists of BIM model versioning systems that record successive states of the IFC file at the documentation management level, without ontological formalization—that is, differences between versions are detected algorithmically but are not represented as entities in a knowledge graph. The third group consists of work on product lifecycle ontologies in mechanical engineering and product lifecycle management (PLM), which apply temporal formalisms to track product configurations but have not been adapted to the specific nature of construction—namely, the multi-phase nature of the investment process, distributed model authorship, and the lack of continuous digital updates [16].

Each of these three groups has a specific structural limitation. The 4D extensions attach time to activities, not to elements. Time enters the model as an attribute of a scheduled task. The building element itself remains a static object that is merely referenced by the task. As a result, 4D models can answer when an element will be built. They cannot answer how the element differed between the design and the execution. Versioning systems operate at the level of files or objects, not at the level of formal semantics. A difference between two versions is computed algorithmically. It exists only as the output of a comparison tool. It is not a queryable entity in a knowledge graph, and it carries no ontological interpretation. Identity between versions is typically established by heuristic GUID matching. This method fails exactly in the cross-phase scenarios described in Section 1. PLM ontologies come closest in spirit. They track product configurations over time. However, they assume a single authoritative configuration management system and a continuous digital thread. Construction projects violate both assumptions. Model authorship is distributed across many parties, and information is exchanged in discrete file-based handovers.

A common limitation of all three groups is the lack of a formal mechanism linking the identity of a building element to its changing properties across successive life cycle phases at the ontology level. None of the existing proposals utilizes the apparatus of fundamental ontology (endurant/perdurant, constitution, and temporal qualification) to address this problem in the context of Linked Building Data.

Against this background, the position of the proposed ontology can be stated precisely. BIM-Phase differs from OPM [12] in the unit of analysis. OPM versions individual property values. BIM-Phase models the element itself as an entity persisting through phases. It treats properties, material constitution, and parthood uniformly as phase-qualified relations. BIM-Phase differs from OWL-Time [17] in temporal granularity. OWL-Time provides instants and intervals on a continuous timeline. BIM-Phase deliberately adopts discrete and ontologically characterized phases. This choice matches the milestone structure of construction practice. The two approaches are complementary. Phases could be aligned

with OWL-Time intervals in future work. Finally, BIM-Phase differs from generic temporal RDF mechanisms such as named graphs or statement-level annotation. These mechanisms record when a triple was asserted. They do not state what kind of entity a phase is. They also do not define how identity, constitution, and parthood behave across phases. BIM-Phase encodes these commitments as explicit axioms grounded in DOLCE. The cost of this choice is the verbosity of the reification pattern. This trade-off is discussed in Section 5. To our knowledge, BIM-Phase is the first construction ontology that grounds element identity across lifecycle phases in a foundational ontology, which is the feature that distinguishes it from OPM, OWL-Time, and named graph approaches.

2.4. Temporal Qualification of Building Elements

Temporal qualification is the principle that governs how building elements are represented in the BIM-Phase ontology. It works such that an element's properties, material structure, and relationships are not assigned directly to the object itself. Instead, they are assigned to a pair consisting of the object and a life cycle phase. Each property therefore has two dimensions: the object dimension and the phase dimension.

In the BIM-Phase ontology, temporal qualification is implemented through a relation reification pattern. For each scalar property, for each material constitution relation, and for each meronomic relation, a reifying class is created. This class links the enduring representing the building element to the perdurant representing the life cycle phase and to a specific property value or a specific relation partner. This pattern preserves the logical decidability required in OWL 2 DL and allows for the flexible extension of the phase qualifier with additional dimensions. The detailed structure of these classes and their axiomatization are presented in Section 3.2.

The operation of this pattern can be illustrated with a minimal formal example. Consider the wall W-01 with a thickness of 40 cm in the design phase and 42 cm in the as-built phase. In the knowledge graph, the wall is a single instance of the class *BuildingElement*. It is linked by the *hasPhase* relation to two phase instances: one of the class *DesignPhase* and one of the class *AsBuiltPhase*. The thickness values are not asserted directly on the wall. Instead, each value is carried by a separate instance of a reifying class. Each of these instances points to the wall through the relation *appliesToElement* and to exactly one phase through the relation *holdsAtPhase*. The property name is stored as a literal of the data property *hasPropertyName* and the numeric value as a literal of *hasValue*. The whole construction uses only binary properties. It therefore stays within OWL 2 DL and requires no n-ary constructs.

Both thickness values coexist in the same graph without logical contradiction, because neither of them is attached directly to the element. In an SPARQL environment, the full history of the property is retrieved by a simple query pattern. The query selects all instances of the reifying class that apply to the given element and carry the property name thickness, and it returns the phase and the value from each of them. For wall W-01, such a query returns two bindings, namely the design phase with a value of 0.40 and the as-built phase with a value of 0.42. The same query template works for material constitution and mereological relations. Only the property name and reifying class change. The example can be reproduced directly with the ontology and the instance files provided as the open research data, and the exact names of the classes and relations are defined in Section 3.2.

Thus, temporal qualification is not merely a technical representational procedure. It constitutes an ontological design commitment, according to which a building element is treated as an entity existing over time, and its states in individual phases are derivatives of this identity. This approach is consistent with the conceptual framework of the DOLCE

ontology discussed in Section 2.1 and distinguishes BIM-Phase from existing construction ontologies described in Section 2.2.

3. Proposed Ontology: BIM-Phase

3.1. *Ontological Commitments and Design Decisions*

The first design decision was the selection of DOLCE as the foundational ontology. This selection was made against four explicit criteria. The first criterion is native support for the phasicity of endurants and for material constitution. The second criterion is the availability of a verified implementation in OWL 2 DL. The third criterion is compatibility with the existing Linked Building Data stack. The fourth criterion is the maturity of documentation and of the supporting community. Among the three most widely used top-level ontologies—namely BFO (Basic Formal Ontology), DOLCE, and UFO (Unified Foundational Ontology)—DOLCE offers the most mature and best-documented apparatus for modeling the temporality of endurants, particularly the constitution relation and the phasicity mechanism—that is, the ability to treat temporal segments of an endurant’s life as distinct ontological entities [3]. Importantly, DOLCE was implemented in OWL 2 (DOLCEbasicOWL) in 2024, enabling direct integration with building ontologies operating within the same formalism [8]. Furthermore, the Building Topology Ontology (BOT) explicitly declares compatibility with DOLCE Ultralite, which facilitates alignment at the class and relation levels [4]. BFO, although widely used in bioinformatics, formalizes temporality in a less flexible manner but is based on the continuant/occurrent distinction without a mechanism for the phasicity of endurants. UFO offers a comparable apparatus (phases, roles, and relators), but its implementation in OWL is less mature than DOLCEbasicOWL. In terms of the four criteria, DOLCE satisfies all of them. BFO fails the first criterion because phase-dependent identity would have to be reconstructed through domain-level extensions. UFO satisfies the first criterion but fails the second one, because gUFO is less mature and less extensively axiomatized than DOLCEbasicOWL. A detailed comparison of the three candidates is presented in Section 2.1. The selection of DOLCE is therefore not a matter of convention but the result of an explicit evaluation.

The second decision was to adopt a simple, three-level ontological model. At the highest level is the endurant, i.e., a building element identical to itself over time, corresponding to a physical object in a building. At the middle level are phases, i.e., perdurants representing the states of this endurant at various stages of its life cycle (design, as-built, FM—facility management). At the lowest level are phase properties, i.e., reified attributes and relations assigned to a specific phase rather than directly to the endurant. This model deliberately forgoes the full DOLCE axiomatics (qualities, quality spaces, and qualia) in favor of a lightweight structure that can be implemented in OWL 2 DL without the need for higher-order reasoners. The choice of exactly three levels deserves a separate justification. A two-level structure, consisting of elements and time-stamped properties, was rejected for two reasons. In such a structure, the phase is not a first-class entity. It cannot carry its own metadata, such as the source IFC file or the date of handover. It also cannot be ordered by an explicit succession relation. A richer structure based on the full DOLCE apparatus of qualities and quality spaces was rejected as well. None of the competency questions defined in Section 4.3 requires this level of detail. The additional layers would multiply the number of instances without improving the answers to the queries. The three-level structure is therefore the minimal structure in which the element, the phase, and the phase-dependent property are all first-class entities. Each level also has a direct counterpart in BIM practice. The endurant corresponds to the physical asset. The phase corresponds to a milestone model. The reified property corresponds to an entry in a property set.

The third decision was to reify phase relations rather than use temporal indexing. In the full DOLCE formalization, relations such as parthood or constitution are indexed by a temporal argument, which requires n -ary relations ($n > 2$) and consequently exceeds the expressive power of OWL 2 DL. Instead, BIM-Phase employs a reification pattern, meaning that each phase relation (e.g., material constitution in the as-built phase) is represented as a separate instance of an intermediary class, linked to the endurant, the phase, and the relation value via simple binary relations. This approach is consistent with the W3C recommendation on modeling n -ary relations in OWL and with the Temporally Qualified Continuants pattern used in DOLCEbasicOWL.

The fourth decision was to limit the temporal model to discrete phases. The BIM-Phase ontology does not model continuous time or changes occurring within a phase. Each phase is treated as an indivisible state, and the transition between phases is represented by a succession relation. This limitation corresponds to BIM practice, in which information is organized around life cycle milestones (handover of design documentation, as-built acceptance, and operational review) rather than around a continuous stream of changes. An extension to a continuous model using OWL-Time [17] is possible but is beyond the scope of this work. The discrete model has consequences that should be stated openly. First, changes occurring within a phase are invisible. If an element is modified twice between two milestones, only the final state is recorded, and the intermediate state is lost. Second, the model assumes that a meaningful linear order of phases exists. Parallel design variants or branching renovation scenarios cannot be represented without extending the phase ordering to a partial order. Third, the granularity of phases is inherited from the file exchange practice. If a project delivers models rarely, the temporal resolution of the knowledge graph is correspondingly coarse. Fourth, phase boundaries are fixed by convention. The model does not represent the duration of a phase or the exact moment of a transition. These limitations are the price of decidability and simplicity. They also define the scope of applicability of BIM-Phase. The ontology is adequate for milestone-based change tracking. It is not adequate for continuous monitoring scenarios, such as streaming sensor data in digital twins. Such scenarios require an alignment with OWL-Time [17] or an event sourcing architecture, which is indicated as a direction for future work in Section 5.3.

In summary, the BIM-Phase ontology was designed as a lightweight, modular extension of the Linked Building Data stack embedded within the DOLCE conceptual framework but reduced to a minimal set of constructs necessary to preserve the identity of a building element over time, track changes in its properties and relationships, and enable cross-phase queries in SPARQL.

3.2. Classes, Relations, and Axioms

The BIM-Phase ontology consists of nine classes, six object properties, and one data property. The specification of each element (Figure 3) is presented below, along with a justification of their role in the model.

Before the formal definitions are given, the structure of the ontology can be summarized intuitively. The reader may think of a single load-bearing wall. The wall as a physical object is one individual, and it never changes its identity. This individual is the endurant. The wall appears in the design model, in the as-built model, and in the operational model. Each of these appearances is one phase individual. Everything that can differ between these appearances, such as the thickness, the material, or the membership in a storey, is not attached to the wall directly. It is attached to a small intermediary object that binds together the wall, one phase, one property name, and one value. These intermediary objects are the reified phase relations. The following subsections define this structure formally.

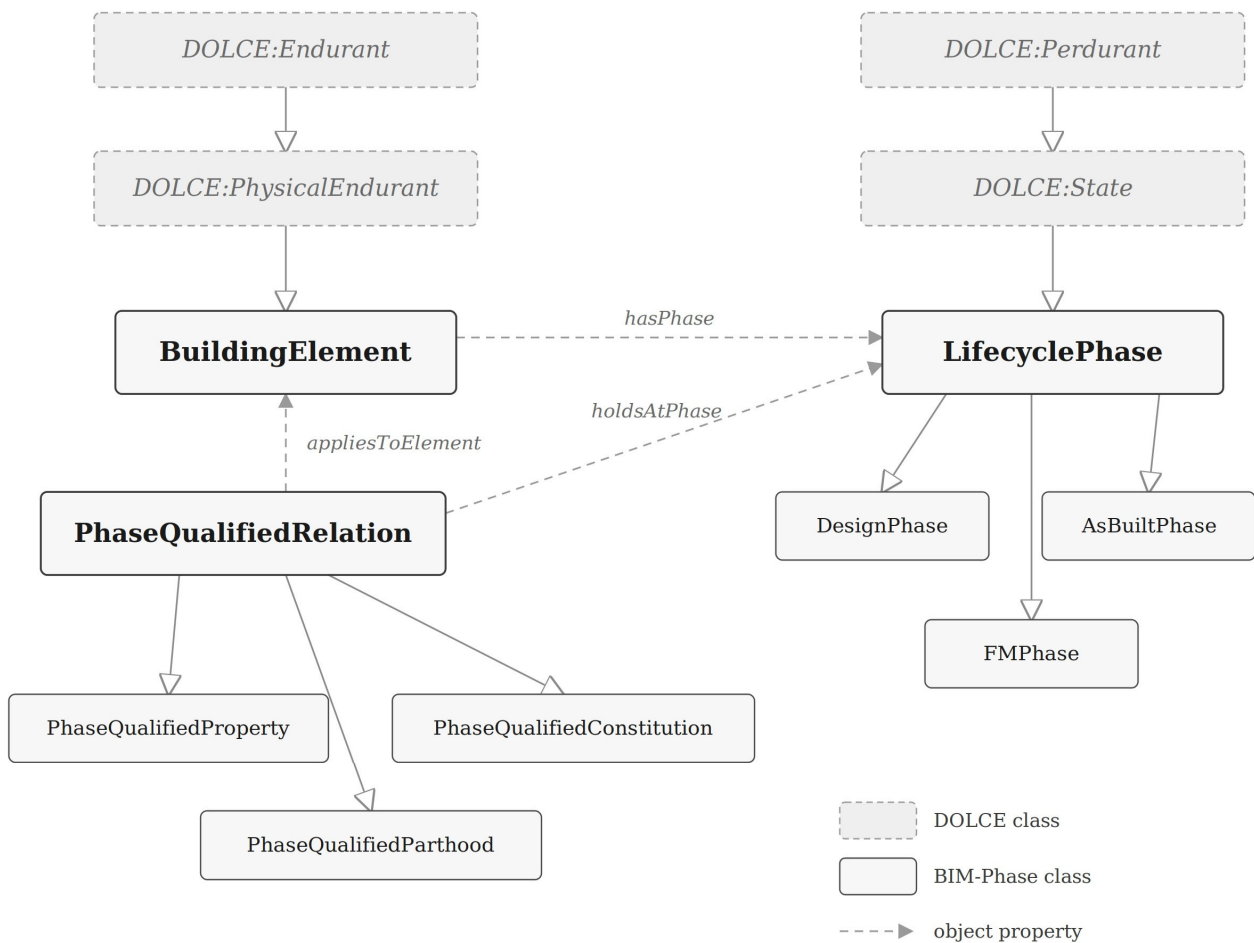


Figure 3. Class hierarchy of the BIM-Phase ontology. Dashed boxes denote DOLCE superclasses; solid boxes denote BIM-Phase classes. Open-headed arrows represent subsumption (*rdfs:subClassOf*); dashed arrows represent object properties linking the three class groups. Source: own elaboration.

3.2.1. Classes

The central class of the ontology is *BuildingElement*, which is a subclass of *DOLCE:PhysicalEndurant*. It represents a physical building element—a wall, floor, column, or window—understood as an entity that remains identical throughout the building’s entire life cycle. Each instance of this class corresponds to a single physical object regardless of how many times it has changed its properties, material, or relationships with other elements. In the IFC context, an instance of the *BuildingElement* refers to a single physical element that may be represented in multiple IFC files under different GUIDs.

The *LifecyclePhase* class denotes a subclass of *DOLCE:State*, i.e., a stative perdurant that is a state. It represents a temporal segment of an endurant’s life corresponding to a single phase of the life cycle. The ontology predefines three subclasses: *DesignPhase*, *AsBuiltPhase*, and *FacilityManagementPhase*, but this taxonomy is extensible—the user can add phases such as *DemolitionPhase* or *RenovationPhase* without modifying the core axioms.

3.2.2. Object Relations

The *hasPhase* relation links a *BuildingElement* to a *LifecyclePhase* and forms the core of the ontology. It is a one-to-many relation—a single endurant can have multiple phases (Figure 4). The inverse relation *isPhaseOf* allows navigation from a phase to an element.

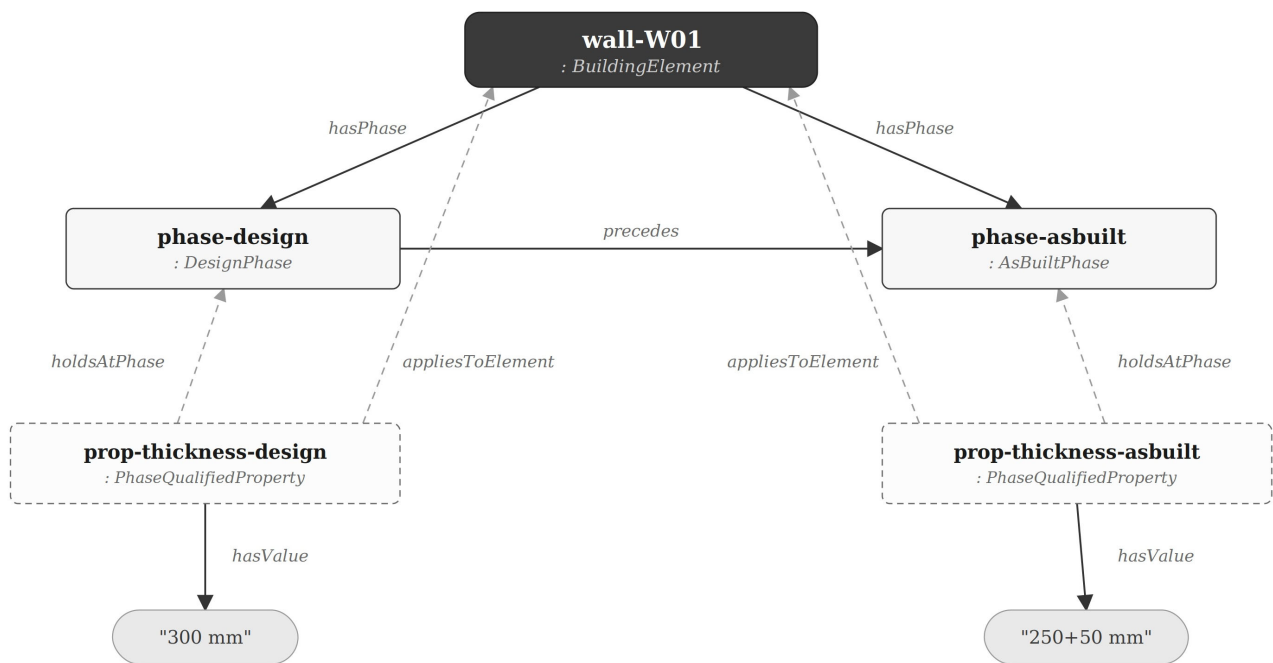


Figure 4. Instance-level example of the reification pattern. The enduring wall-W01 has two lifecycle phases linked by precedes. Source: own elaboration.

The precedes relation links two instances of LifecyclePhase and expresses temporal succession. It is transitive (if phase A precedes B and B precedes C, then A precedes C). This relation allows for the reconstruction of the complete chronology of phases for a given element.

There are three relations linking reification classes to their context: appliesToElement (indicates which enduring a given phase relation applies to), holdsAtPhase (indicates in which phase a given relation applies), and hasValue (indicates the value of the relation—material, dimension, or component). Each instance of a reification class is associated with exactly one appliesToElement relation, exactly one holdsAtPhase relation, and at least one hasValue relation.

Additionally, the ontology defines the hasPropertyName relation as a data property of type xsd:string, assigned to an instance of PhaseQualifiedRelation, which identifies the name of a property or relation (e.g., “thickness”, “constitutiveMaterial”, “hasPart”). This solution avoids the proliferation of classes for each property type—instead of creating separate classes such as ThicknessAtPhase, MaterialAtPhase, etc., we use a single reification class with a name attribute.

3.2.3. Axioms

The axiomatics of the BIM-Phase ontology are intentionally minimal and limited to six axioms expressible in OWL 2 DL.

Axiom 1: Every instance of BuildingElement must have at least one hasPhase relation (minimum cardinality constraint: “BuildingElement $\sqsubseteq \exists$ hasPhase.LifecyclePhase”). A building element without any phase is ontologically empty.

Axiom 2: Every instance of PhaseQualifiedRelation must be associated with exactly one element (“PhaseQualifiedRelation $\sqsubseteq =1$ appliesToElement.BuildingElement”) and with exactly one phase (“PhaseQualifiedRelation $\sqsubseteq =1$ holdsAtPhase.LifecyclePhase”). A phase relation without a context is undefined.

Axiom 3: The classes BuildingElement and LifecyclePhase are disjoint (“BuildingElement \sqcap LifecyclePhase $\sqsubseteq \perp$ ”). An enduring cannot simultaneously be its own phase.

Axiom 4: The precedes relation is transitive. This allows us to infer a complete chronological closure of phases, including pairs of phases not directly connected. Transitivity alone does not imply acyclicity of the phase order. Acyclicity is guaranteed at the IFC translation layer described in Section 3.3, where phases are ordered according to the chronology of the source files.

Axiom 5: The subclasses of LifecyclePhase—DesignPhase, AsBuiltPhase, and FacilityManagementPhase—are disjoint pairs. An element may have at most one instance of each phase type.

Axiom 6: The hasPhase relation is the inverse of the isPhaseOf relation (“hasPhase \equiv isPhaseOf⁻”).

3.2.4. Complete Schema

The schema comprises nine classes connected by six object relations and one data relation (Figure 5). BuildingElement and LifecyclePhase are linked by the pair of inverse relations hasPhase/isPhaseOf. The precedes relation orders the phases temporally and is declared transitive. PhaseQualifiedRelation serves as a reification node: each instance points to exactly one element (appliesToElement), exactly one phase (holdsAtPhase), and at least one value (hasValue), while hasPropertyName identifies the relation type as xsd:string. Three specific subclasses—PhaseQualifiedProperty, PhaseQualifiedConstitution, and PhaseQualifiedParthood—specialize the reification pattern for attributes, material constitution, and meronomic composition, respectively. DesignPhase, AsBuiltPhase, and FacilityManagementPhase are mutually exclusive subclasses of LifecyclePhase, and the axiom “BuildingElement \sqcap LifecyclePhase $\sqsubseteq \perp$ ” guarantees that no individual can be both an enduring and its temporal phase (in other words, simply put, nothing exists that is simultaneously a building element and a life cycle phase).

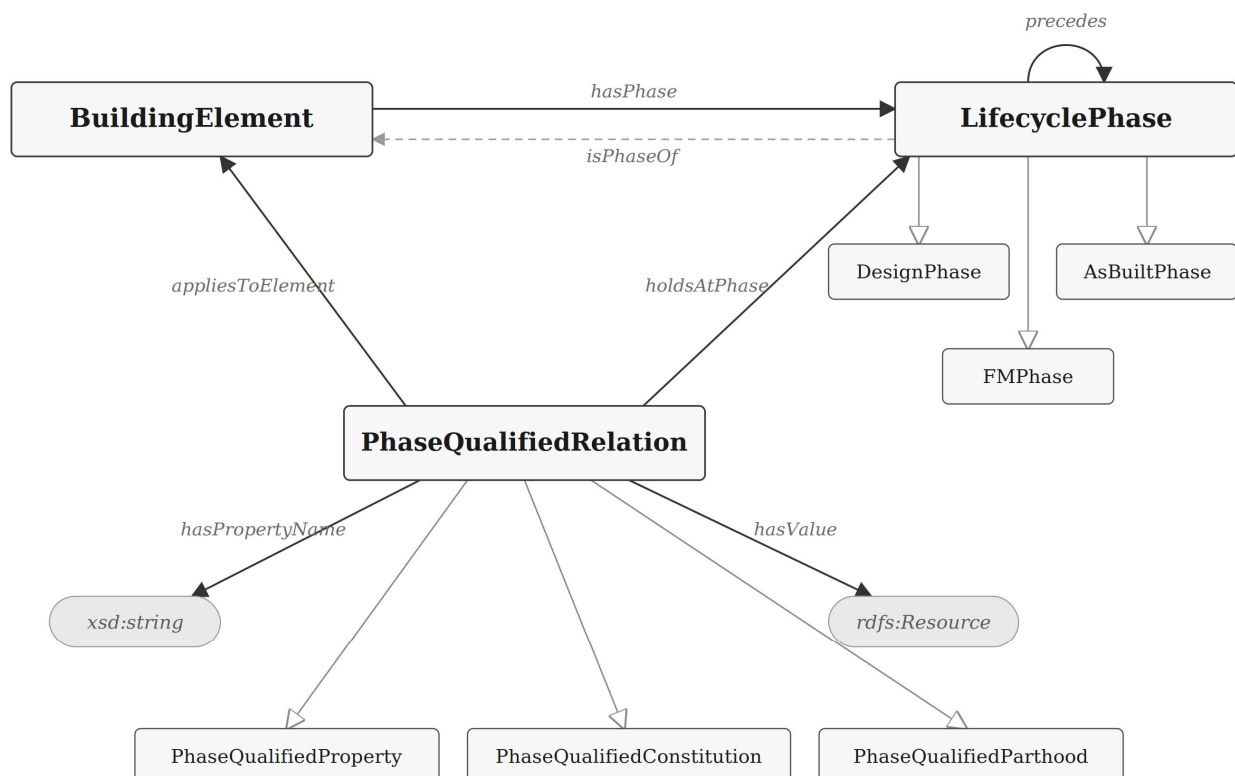


Figure 5. Complete schema of the BIM-Phase ontology. Solid arrows denote object and data properties; light arrows denote subsumption (rdfs:subClassOf). Rounded boxes represent datatypes. Source: own elaboration.

3.3. IFC Translation Rules

The BIM-Phase ontology operates at the level of a knowledge graph, while the source data comes from IFC files exported from authoring tools (in this work, from Autodesk Revit 2026.2). To transition from a set of IFC files to an ontology instance, it is necessary to define translation rules. These rules do not constitute an automatic pipeline but rather a formal description of the mapping between IFC structures and BIM-Phase constructs, which can be implemented manually via scripting (e.g., in Python 3.11 using the `IfcOpenShell` library) or using SPARQL CONSTRUCT rules. The starting point is the assumption that, for a single building object, there are at least two IFC files corresponding to different phases of the life cycle, and they are typically the following: the design model, the as-built model, and the operational model. Each file represents an independent snapshot of a given state, in which the same physical element may have a different GUID, different property values, and different geometries.

Rule one concerns the identification of the endurant. For each physical element to be tracked across phases, a single instance of the `BuildingElement` class is created with its own URI identifier (e.g., `bph:wall-W01`). This identifier is independent of IFC GUIDs and serves as a persistent identifier for the entity in the knowledge graph. The association with specific IFC instances in individual files is stored as a phase attribute (e.g., `hasIFCGuid "202Fr$t4X7Zf8NOew3FNr2"`), which enables backtracking to the source file. In practice, identifying the same element across different IFC files requires expert knowledge or heuristics (approximate matching methods) based on geometric location and element type, as IFC does not guarantee GUID preservation across exports.

Rule two concerns the creation of phases. Each IFC file corresponds to a single instance of the `LifecyclePhase` subclass. The design model file generates a `DesignPhase` instance, the as-built file generates an `AsBuiltPhase` instance, and the operational file generates a `FacilityManagementPhase` instance. The instances of the phases are linked to the endurant via the `hasPhase` relationship and to each other via the `precedes` relationship, in accordance with the chronology of file creation.

Rule three concerns the extraction of phase properties. For each property of an IFC element that is relevant for change tracking, an instance of the appropriate `PhaseQualifiedRelation` subclass is created. Scalar properties from IFC `PropertySets` (e.g., thickness, height, technical condition) are mapped as instances of `PhaseQualifiedProperty`. The material assigned to an element (`IfcMaterialLayerSetUsage` or `IfcMaterial`) is mapped as an instance of `PhaseQualifiedConstitution`. Meronomic relationships (`IfcRelAggregates`, `IfcRelContainedInSpatialStructure`) are mapped as instances of `PhaseQualifiedParthood`. Each instance of a reification class is associated with an endurant (`appliesToElement`), a specific phase (`holdsAtPhase`), a property name (`hasPropertyName`), and a value (`hasValue`).

The fourth and final rule concerns the scope of the translation. Not all properties and relations from the IFC file are transferred to the knowledge graph, as the translation covers only those attributes that are the subject of competency questions defined in Section 4.3. This restriction stems from the principle of ontological minimalism adopted in Section 3.1 and avoids a large number of reification instances for properties that are irrelevant from the perspective of tracking inter-phase changes.

The proposed rules can be compared with two alternative mapping approaches. The first alternative is the direct conversion of IFC to `ifcOWL`. This conversion is complete and lossless. However, it reproduces the full complexity of the EXPRESS schema and yields very large graphs. It also preserves the snapshot character of a single file, so the cross-phase identity problem remains unsolved. The second alternative is the `IFCtoLBD` converter [18]. It produces compact graphs structured by the `BOT` and `PROPS` ontologies. Yet each conversion still covers one file at a time. The outputs of consecutive conversions

are not linked, and the same physical element appears as several unrelated resources. The rules proposed here differ from both alternatives in two respects. They are selective, because the scope of the translation is driven by competency questions. They are also multi-file by design, because the enduring identifier deliberately spans several IFC files. In this sense, the proposed mapping is complementary to the existing converters. The output of IFCtoLBD could serve as the input layer from which phase-qualified instances are constructed.

The degree to which the rules can be automated differs between the rules. Rules two, three, and four are deterministic. Given a set of IFC files with an assigned phase order, the creation of phase instances, the extraction of the selected properties, and the restriction of the scope can be executed mechanically, for example with a script based on the IfcOpenShell library or with SPARQL CONSTRUCT queries over an ifcOWL graph. Rule one is the only rule that requires judgement. Matching the same physical element across files with different GUIDs needs expert knowledge or heuristics based on geometric location and element type. The manual effort of the whole translation is therefore concentrated in one well-defined step. This observation defines a realistic path to partial automation, in which a converter executes rules two to four automatically and presents candidate matches for rule one to a human operator for confirmation.

3.4. Event-Based Extension of BIM-Phase

In its basic form, the BIM-Phase ontology represents the states of building elements in the various phases of their life cycle. However, it does not introduce an explicit mechanism for describing the transitions between these states. Change is present in it indirectly: that is, as the difference between property values in two consecutive phases. This approach is sufficient for tracking the fact of change, but it does not allow it to be treated as a separate entity in the knowledge graph.

This paper proposes an event-based extension of the BIM-Phase ontology, which introduces a layer of transitions as separate model instances. The central element of the extension is the `PhaseTransitionEvent` class, defined as a subclass of `DOLCE:Perdurant`. Each instance of this class is associated with exactly one source phase via the `transitionsFrom` relation, with exactly one target phase via the `transitionsTo` relation, and with exactly one building element via the `affectsElement` relation. All three relations are functional in nature, which ensures unambiguous interpretation of the event in the knowledge graph.

A `PhaseTransitionEvent` instance is additionally associated with a pair of instances of the reifying classes introduced in Section 3.2. One of them represents the state of the element in the source phase, and the other represents its state in the target phase. Thanks to this, the event is not a detached entity but serves as a bridge between two explicitly reified states of the same enduring. The detailed cardinality axioms and the domain and codomain restrictions of relations are consistent with the logical decidability of OWL 2 DL.

The extension is designed to provide three reasoning and querying capabilities that the basic model does not offer. First, a change becomes a queryable entity. A single SPARQL query can retrieve a transition event together with the pair of reified states it connects and can therefore answer what exactly changed during a given transition. Second, the functional character of the three relations makes integrity violations detectable by a standard reasoner. An event with two target phases renders the ontology inconsistent. Third, events are composed using the transitive `precedes` relation. This makes it possible to reconstruct the history of an element as an ordered chain of transitions. These capabilities are relevant for lifecycle analytics tasks, such as auditing which interventions affected an element and in what order.

This layer is a first-class extension of the proposed framework and not a peripheral addition, although it is kept modular to preserve flexibility (see Section 3.4). Some applications of the BIM-Phase ontology require only the representation of phase states: for example, in tasks involving the verification of model compliance with information requirements. Other applications, such as life cycle analytics or digital twins of buildings, utilize the full event layer. Separating these two levels allows the ontology to be applied flexibly depending on the needs of a specific scenario.

The introduction of the event layer opens up the possibility of treating a change as a process with its own temporal structure rather than merely as a difference between states. The application of the event extension in a case study is presented in Section 4.2, which demonstrates how successive phases of the W-01 exterior wall's life cycle are linked by instances of PhaseTransitionEvent.

4. Implementation and Case Study

4.1. Technical Architecture and Toolchain

The validation of the BIM-Phase ontology was conducted using the example of a single-family residential building developed in Autodesk Revit 2026.2 [19]. The goal of the case study is not to build an automated translation pipeline but to demonstrate that the ontological constructs defined in Section 3—i.e., durants, temporal phases, and reified properties—find a direct mapping in BIM model data created in a typical authoring tool.

The model was prepared in three separate Revit project files (.rvt), corresponding to the design phase, the as-built phase, and the facility management phase. Each file was exported to IFC format using the built-in exporter. Since the three Revit files were created independently, the GUIDs of IFC elements are not preserved across files. The same exterior wall W-01 has three different GUIDs in the design, as-built, and facility management phases, even though ontologically it represents one and the same physical entity. This situation reflects a typical investment scenario and provides a direct motivation for introducing a separate durant identifier in the BIM-Phase ontology, independent of the GUIDs in the source files.

The BIM-Phase ontology was implemented in the Protégé 5.6 editor [20] in OWL 2 DL format (Turtle serialization) and validated using the built-in HermiT reasoner for logical consistency and compliance with the cardinality constraints defined in Section 3.2.3. Case study instances, i.e., durants, phases, and reified properties corresponding to data from the Revit model, were manually entered into the knowledge graph and queried using SPARQL 1.1 queries in the SPARQL Query interface built into Protégé. The full ontology, knowledge graph with instances, and SPARQL queries are available in the open research data repository of the Warsaw University of Technology. This material makes the case study reproducible. The instance graph can be checked triple by triple against the translation rules of Section 3.3. The reasoner validation can be repeated in Protégé. All SPARQL queries can be executed on the published graph without any additional software.

The complete technical workflow of the case study can be summarized in six steps. In step one, three independent Revit project files were prepared, one for each lifecycle phase of the building. In step two, each file was exported to IFC using the built-in exporter. In step three, the BIM-Phase ontology was implemented in Protégé in the OWL 2 DL profile. In step four, the instances were created in the knowledge graph according to the four translation rules of Section 3.3, including the assignment of one persistent durant identifier per physical element. In step five, the ontology together with the instances was validated with the HermiT reasoner. In step six, the eight competency questions were executed as SPARQL queries, and the complete material was published in the open research

data repository. Each step has a defined input and output, so the workflow can be repeated for any other building.

4.2. Case Study: The Three-Phase Lifecycle of a Residential Building

The subject of this case study is a single-family residential building with a usable area of approximately 120 m², designed using masonry construction with a gable roof. The model was developed in Autodesk Revit 2026.2 and includes typical structural elements: exterior and interior walls, floors, columns, foundations, and window and door frames. From among these elements, three representative objects were selected for detailed ontological analysis: exterior wall W-01, ground floor slab S-01, and structural column C-01. This selection ensures coverage of the three types of phase changes modeled by the BIM-Phase ontology: changes in scalar properties (PhaseQualifiedProperty), changes in material constitution (PhaseQualifiedConstitution), and meronomic changes (PhaseQualifiedParthood).

For each of the three elements, three model variants were prepared, corresponding to successive phases of the life cycle. In the design phase, the model reflects the designer's intentions in accordance with the technical documentation. In the as-built phase, the model accounts for changes introduced during construction, documented in the as-built documentation. In the facility management phase, the model records the current state of the facility, including damage, material replacements, and changes in the status of elements resulting from technical inspections [21].

The W-01 exterior wall serves as the main demonstration object, as it undergoes changes across all three ontological dimensions. In the design phase (Figure 6), the wall is designed as a single-layer structure of 300 mm thick aerated concrete, without an insulation layer and without internal installations.

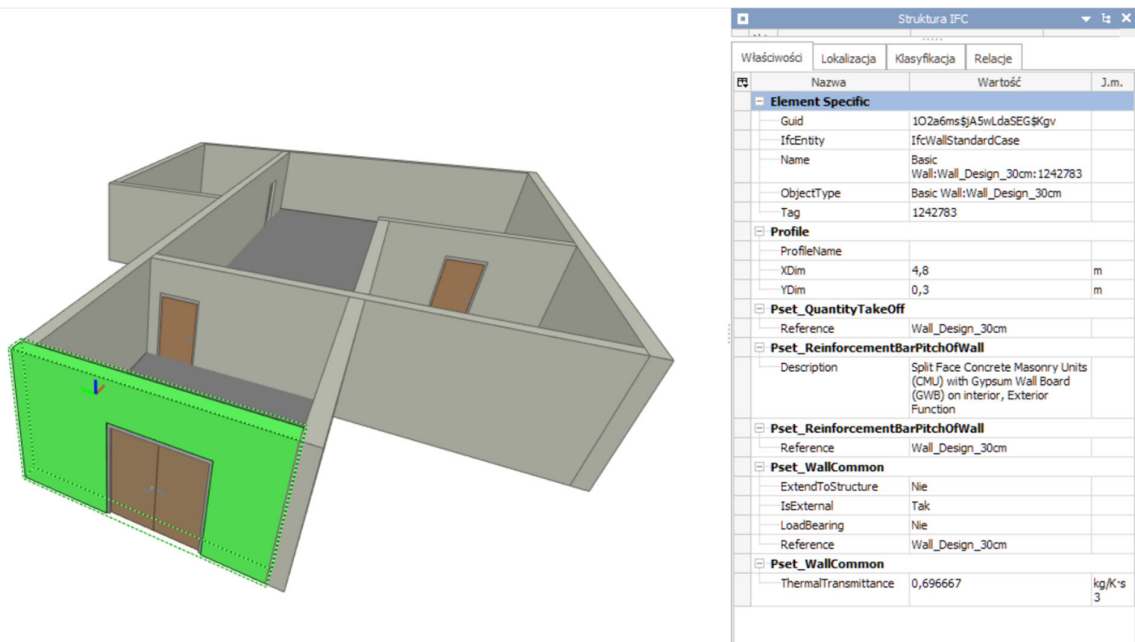


Figure 6. Design phase of the case-study wall W-01: 300 mm aerated concrete, no insulation layer, and no internal installations. Source: own elaboration in BIMvision 3.1.2.

In the as-built phase (Figure 7), the thickness of the structural layer is reduced to 250 mm, and a 50 mm layer of expanded polystyrene (EPS) insulation is added to the exterior; this constitutes a change in material constitution (PhaseQualifiedConstitution). At the same time, electrical wiring is installed inside the wall, constituting a mereological

change (PhaseQualifiedParthood)—an element that had no subparts in the design phase now gains a subcomponent.

Family: Basic Wall
 Type: Wall_AsBuilt_25+5cm
 Total thickness: 0.3000 (Default)
 Resistance (R): 19.1362 (h·ft²·°F)/BTU
 Thermal Mass: 7.7266 BTU/(ft²·°F)

Sample Height: 3.0000

Layers

EXTERIOR SIDE							
	Function	Priority	Material	Thickness	Wraps	Structural Material	Variable
1	Core Boundary		Layers Above Wrap	0.0000			
2	Structure	1	EIFS, Exterior Insulation	0.0500	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
3	Structure	1	Concrete, Lightweight	0.2500	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
4	Core Boundary		Layers Below Wrap	0.0000			

INTERIOR SIDE

Insert Duplicate Delete Up Down

Default Wrapping
 At Inserts: Do not wrap
 At Ends: None

Modify Vertical Structure (Section Preview only)
 Modify Merge Regions Sweeps
 Assign Layers Split Region Reveals

OK Cancel Help

Figure 7. As-built phase of wall W-01: thickness changed to 250 mm structural layer + 50 mm EPS insulation; electrical installation added inside the wall (mereological change). Source: own elaboration in Autodesk Revit 2026.2.

In the operational phase (Figure 8), the EPS insulation is replaced with mineral wool (another constitutional change), a crack is recorded in the structural layer, and the element’s status changes from “constructed” to “requires repair” (PhaseQualifiedProperty change).

The ground floor slab S-01 illustrates a simpler scenario—a change in a scalar property without a change in material. In the design phase, the slab is 200 mm thick (reinforced concrete C25/30). During construction, the thickness is increased to 220 mm as a result of a contractor’s decision. The material composition and meronymy remain unchanged across all three phases, while only the thickness and the lifecycle status are updated, demonstrating that the ontology correctly records both changes and continuity.

Structural column C-01 demonstrates a change in material composition during the service life. In the design and as-built phases, the column is identical—C30/37 reinforced concrete with a cross-section of 300 × 300 mm. In the FM phase, following a technical inspection, the column is reinforced with carbon-fiber-reinforced polymer (CFRP) [22] overlays, which constitutes a change in material composition without altering the geometry. Table 2 summarizes the phase-qualified changes for the three analyzed elements.

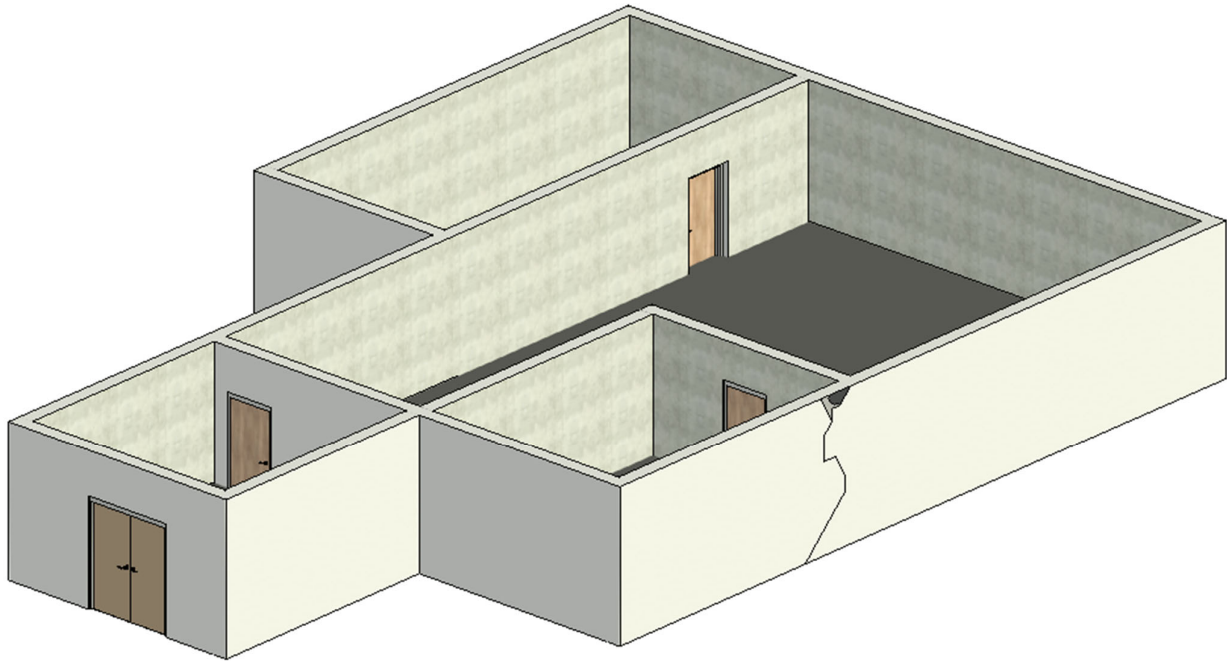


Figure 8. Facility management phase of wall W-01: insulation replaced with mineral wool (constitution change), crack damage recorded, and status changed from “constructed” to “requires repair”. Source: own elaboration in Autodesk Revit 2026.2.

Table 2. Summary of phase-qualified changes across three lifecycle phases for the case-study elements. Cells marked with Δ indicate a change relative to the preceding phase. Source: own elaboration.

Element	Property	Design	As-Built	FM
Wall W-01	thickness	300 mm	Δ 250 + 50 mm	250 + 50 mm
	constituent material	aerated concrete	Δ aerated concrete + EPS	Δ aerated concrete + MW
	hasPart	none	Δ electrical inst.	electrical inst.
	status	designed	constructed	Δ requires repair
Slab S-01	thickness	200 mm	Δ 220 mm	220 mm
	constitutive material	RC C25/30	RC C25/30	RC C25/30
	status	designed	constructed	good condition
Column C-01	cross-section	300 × 300 mm	300 × 300 mm	300 × 300 mm
	constitutiveMaterial	RC C30/37	RC C30/37	Δ RC C30/37 + CFRP
	status	designed	constructed	Δ reinforced
Note:	Δ = change relative to the preceding phase		highlighted cells	

Each row of the table marked with the symbol Δ corresponds to a single instance of a reification class in the knowledge graph for which its value (hasValue) differs from the value in the previous phase. In total, the case study generates three endurants, nine phases, and 27 instances of reification classes, 7 of which record an inter-phase change. The complete knowledge graph in Turtle format is available in open research data.

Within the proposed event-based extension, transitions between lifecycle phases of wall W-01 can be represented as distinct events. The transition between the design and as-built phases can be interpreted as a construction event (ConstructionEvent), which is responsible for changes in material composition (addition of the insulation layer) and the introduction of new mereological components (electrical installation). Similarly, the transition between the as-built and facility management phases can be modeled as a maintenance event (MaintenanceEvent), involving the replacement of insulation material and a change in the element’s status. Consequently, each change marked with Δ in Table 2

can be associated with a corresponding transition event, providing a causal interpretation of the element's evolution.

4.3. Competency Questions and Ontology-Driven Validation

The validation of a domain ontology is not limited to checking logical consistency using a reasoner. It also requires demonstrating that the proposed conceptual framework actually answers the questions for which it was designed. In the NeOn methodology and in the classical Uschold–King approach, the tool used for this purpose is the so-called competency questions (CQs). These are queries formulated in natural language to which the ontology should enable a response [23,24]. Competency questions serve a dual purpose. First, they constitute a criterion for the model's scope. This has already been utilized in the fourth rule of IFC translation (Section 3.3), which limits the translation to those properties that are the subject of CQs. Second, they provide a criterion for a posteriori validation. For each CQ, one must identify the ontology constructs (classes and relations, axioms) that collectively enable the formulation of an answer in the knowledge graph.

Eight competency questions have been defined for the BIM-Phase ontology. They have been grouped into three categories corresponding to the three types of phase changes modeled by the ontology. These are changes in scalar properties, changes in material constitution, and meronomic changes. The set was supplemented with questions regarding the identity of an element over time and the order of phases. This set reflects the design intentions of the ontology formulated in Section 1 and is directly linked to the case study scenarios discussed in Section 4.2.

The competency questions for the BIM-Phase ontology are as follows. CQ1: Which building elements changed the value of a given scalar property between two specified life cycle phases? CQ2: What was the value of a specific property of a given element in a particular phase? CQ3: Which elements changed their constituent material between the design phase and the as-built phase? CQ4: What materials constituted a given element in the successive phases of its life cycle? CQ5: Which elements gained or lost subparts (meronomic relation) between the two phases? CQ6: What subparts did a given element have in a specific phase? CQ7: Does a given endurant occur in all three phases of the life cycle or only in selected ones? CQ8: What is the temporal order of a given element's phases according to the precedes relation?

Table 3 presents the mapping of competency questions to constructs of the BIM-Phase ontology. For each question, the classes, relations, and axioms whose co-occurrence in the knowledge graph enables the formulation of an answer are indicated. This mapping constitutes a validation trace linking functional requirements to the structural elements of the model.

An analysis of the table yields three observations relevant to the evaluation of the ontology. First, each of the six axioms defined in Section 3.2.3 is required by at least one competence query. This confirms that the axiomatics is minimal in the sense of necessity. Removing any axiom leads to the loss of the ability to answer at least one CQ. Second, the reification class *PhaseQualifiedRelation*, together with its three subclasses, covers all questions concerning inter-phase changes (CQ1, CQ3, and CQ5). This provides empirical confirmation of the validity of choosing the reification pattern as the primary mechanism for modeling changes. Third, CQ7 and CQ8 regarding identity and temporal order are handled exclusively by relations between *BuildingElement* and *LifecyclePhase*, without the involvement of reification classes. This confirms that the endurant/perdurant distinction carried over from DOLCE is a real criterion for the division of responsibility in the model and not merely a declarative theoretical embellishment.

Table 3. Mapping of competency questions to BIM-Phase ontology constructs. Source: own elaboration.

CQ	Category	Classes	Relations	Axioms
CQ1	scalar change	BuildingElement, LifecyclePhase, PhaseQualifiedProperty	hasPhase, appliesToElement, holdsAtPhase, hasValue, hasPropertyName	A1, A2
CQ2	scalar state	BuildingElement, LifecyclePhase, PhaseQualifiedProperty	appliesToElement, holdsAtPhase, hasValue	A2
CQ3	constitution change	BuildingElement, DesignPhase, AsBuiltPhase, PhaseQualifiedConstitution	appliesToElement, holdsAtPhase, hasValue	A2, A5
CQ4	constitution state	BuildingElement, LifecyclePhase, PhaseQualifiedConstitution	hasPhase, appliesToElement, holdsAtPhase, hasValue	A2
CQ5	mereological change	BuildingElement, LifecyclePhase, PhaseQualifiedParthood	appliesToElement, holdsAtPhase, hasValue	A2
CQ6	mereological state	BuildingElement, LifecyclePhase, PhaseQualifiedParthood	appliesToElement, holdsAtPhase, hasValue	A2
CQ7	identity continuity	BuildingElement, LifecyclePhase	hasPhase, isPhaseOf	A1, A3, A6
CQ8	phase ordering	LifecyclePhase, DesignPhase, AsBuiltPhase, FacilityManagementPhase	precedes	A4, A5

The logical consistency of the BIM-Phase ontology, along with a knowledge graph containing three endurants, nine phases, and 27 instances of reification classes (Section 4.2), was verified using the HermiT reasoner. The reasoner detected no contradictions, correctly classified the hierarchy, and deduced the transitive closure of the precedes relation for pairs of phases not directly connected. The full ontology in OWL 2 DL (Turtle) format, a knowledge graph with case study instances, and sample SPARQL queries implementing all eight CQs are available in the open research data repository of the Warsaw University of Technology. This enables independent reproduction of the validation results.

5. Discussion

5.1. Evaluation

The evaluation of the BIM-Phase ontology covers three complementary aspects: formal correctness, functional completeness, and substantive meaning. In terms of formal correctness, the ontology was validated using the HermiT reasoner in the Protégé 5.6 environment. The reasoner detected no contradictions in the class hierarchy or in the knowledge graph containing the case study instances, and all cardinality constraints defined in Section 3.2.3 were satisfied. The third axiom (disjointness of the BuildingElement and LifecyclePhase classes) was tested by intentionally introducing an individual with dual classification. The reasoner reported an inconsistent ontology, which confirms the validity of the disjointness axiom. In terms of functional completeness, each of the eight competency questions (Table 3) is fully covered by the ontology constructs. Each of the six axioms is utilized by at least one CQ, which demonstrates the minimalism of the axiomatics in the sense of

necessity. The three reification classes cover the three dimensions of phase changes and find natural application in a real-world design-construction-operation scenario.

In terms of substantive meaning, BIM-Phase introduces a conceptual layer into the BIM domain that was previously missing. The classic IFC schema treats a building element as a snapshot of a single state, and existing ontologies from the Linked Building Data ecosystem (ifcOWL, BOT) inherit this limitation. This gap, though not explicitly named as such, is also recognized by broader critiques of the direction of BIM development, which point out that current paradigms based on proprietary tools and file-based data exchange limit interoperability, scalability, and data access and that the future lies in data granularity [25]. The BIM-Phase ontology addresses this criticism from a semantic perspective by extending the concept of granularity to include temporal and identification dimensions—that is, the ability to consistently track the same object despite changes in its states throughout its lifecycle.

It is worth noting that some of the practical limitations of static BIM are already being mitigated by IoT-enabled monitoring systems. Long-term sensor-based monitoring of building operation is an established practice. It supports the prediction of operational performance with high temporal and spatial resolution, as demonstrated by a long-term study of a zero-energy office building in Singapore [26]. Such systems, however, operate at the level of measurement data streams. They record the behavior of a building, but they do not represent the identity of its elements or the history of their states in a machine-interpretable form. The BIM-Phase ontology is complementary to this trend. It provides the semantic anchor to which operational data streams can be attached. Sensor readings, inspection results, and maintenance events can then refer to one and the same persistent element across the whole life cycle.

5.2. Limitations

The BIM-Phase ontology in its current form has several limitations resulting from both the adopted design assumptions and the scope of the case study. First, the validation was based on a single, single-family residential building with a limited number of structural elements. This scale is sufficient to demonstrate the feasibility of the proposed approach, but it does not allow for an assessment of the ontology's behavior with large BIM models comprising thousands of element instances and their phases. Because the ontology stays within OWL 2 DL and uses only binary properties, the reasoning complexity remains that of the standard DL reasoner, and the practical cost grows with the number of reification instances rather than with the expressivity of the model. Second, the knowledge graph instances were entered manually based on the Revit model rather than generated automatically from an IFC file. The translation rules described in Section 3.3 are conceptual in nature and require further operationalization in the form of a full converter. The analysis of the automatability of the individual rules, added in Section 3.3, shows that the manual effort is confined to the identity-matching step. Third, the event extension introduced in Section 3.4 is a conceptual contribution at this stage. Its application in the case study is illustrative; its reasoning capabilities follow from the design of the axioms rather than from a separate empirical demonstration, and it does not cover scenarios in which a single event affects multiple building elements simultaneously.

The second set of limitations consists of ontological assumptions. The number of life cycle phases has been narrowed down to three key stages: the design phase, the as-built phase, and the operational phase. Applications involving the pre-design phase, the demolition phase, or intermediate phases require an extension of the set of instances of the LifecyclePhase class. The ontology also does not model uncertainty or competing model versions within the same phase, which may be relevant in scenarios involving multiple

design teams working in parallel. These limitations do not undermine the proposed conceptual framework but rather point the way for further work, as outlined in Section 5.3.

5.3. Future Work

The first direction for future work concerns the operationalization of the BIM-Phase ontology in design practice tools. The key task is to develop an automatic converter from IFC files to a knowledge graph compliant with the proposed ontology, which will implement the conceptual translation rules described in Section 3.3. Such a converter should be integrated with the IfcOpenShell library and should support the sequential loading of multiple IFC files corresponding to successive phases of the same object's life cycle. Concurrently, we plan to develop a plugin for the Autodesk Revit editor and the BlenderBIM environment, which will enable direct export of the model along with phase metadata. The result of this work is to be a complete toolchain leading from the author's model to SPARQL queries on the knowledge graph.

The second direction concerns expanding the conceptual scope of the ontology itself. We plan to extend the set of instances of the LifecyclePhase class to include a pre-design phase, a demolition phase, and intermediate phases related to modernization and renovation. The event extension introduced in Section 3.4 requires supplementation with a class of multi-element events, in which a single phase transition affects multiple building elements simultaneously. The planned work also includes a full empirical demonstration of the event layer, in which transition events will be instantiated in the knowledge graph, and their reasoning behavior will be verified with the same procedure as the core axioms. Further work will also include ontology validation on larger BIM models and integration with applications in the areas of digital twins and lifecycle analytics. In particular, it is planned to integrate the semantic layer with machine learning systems, in which the BIM-Phase ontology will serve as a conceptual layer for classification and prediction tasks based on building operation data.

6. Conclusions

This paper presents the BIM-Phase ontology, which formalizes temporally qualified relationships between building elements and their states in various phases of a building's life cycle. The ontology is embedded within the conceptual framework of the DOLCE fundamental ontology, which provided the apparatus for distinguishing between endurants and perdurants, as well as mechanisms for the temporal qualification of properties and relations. Building elements are treated as endurants, and life cycle phases are treated as perdurants. Scalar properties, material constitution, and meronomic relations are represented as phase-dependent constructs using a reification pattern. This approach allows for the separation of an object's identity from its temporal states and enables the consistent tracking of changes in an element throughout its entire life cycle.

The ontology was implemented in the Protégé 5.6 editor in OWL 2 DL format and validated for logical consistency using the HermiT reasoner. The ontology's functionality was demonstrated using a case study of a single-family residential building developed in the Autodesk Revit environment, encompassing three representative types of phase changes. Eight competency questions confirmed the ontology's functional completeness and the minimalism of its axioms. The work was supplemented with an event extension in which transitions between life cycle phases are modeled as separate instances of the PhaseTransitionEvent class. This extension allows a change to be treated not only as a difference between states, but as a process with its own temporal structure.

The main findings can be summarized explicitly. First, the identity of a building element can be preserved across independent IFC files by separating the endurant from

its phases. This solves the problem of unstable GUIDs at the semantic level. Second, a deliberately minimal set of nine classes, seven relations, and six axioms is sufficient to answer all eight competency questions. This demonstrates that lifecycle change tracking does not require a heavyweight ontology. Third, the reification pattern keeps the model within OWL 2 DL, so standard reasoners and SPARQL engines remain fully applicable. Fourth, the translation from IFC files to the knowledge graph can be defined by four explicit rules, of which only the identity-matching step requires human judgement.

The proposed conceptual framework introduces a semantic layer to the BIM domain that has been missing in existing construction ontologies. Its significance extends beyond a technical representational exercise and addresses the broader need for a granular, persistent, and interoperable description of building objects throughout their full life cycle. The BIM-Phase ontology provides a foundation for further work on digital twins, life cycle analytics, and the integration of the semantic layer with machine learning methods. Thus, this work contributes to the development of the semantic foundations of construction engineering, in which a design object is understood as an entity existing over time rather than as a single moment in time.

These findings should be read together with the limitations of the study. The validation covered a single residential building with manually created instances. The set of phases was narrowed to three milestones. The event extension remains a conceptual contribution at this stage. Future work follows directly from these limitations. The priorities are an automatic converter implementing the translation rules, validation on large and federated BIM models, an extended set of phases covering renovation and demolition, the empirical demonstration of the event layer, and the integration of the ontology with digital twins and machine learning systems, as detailed in Section 5.3.

Author Contributions: Conceptualization, A.S.B.; methodology, A.S.B.; validation, A.S.B. and P.J.; formal analysis, A.S.B.; resources, P.J.; data curation, A.S.B. and P.J.; writing—original draft preparation, A.S.B., P.J., and A.S.; writing—review and editing, A.S.B., P.J., and A.S.; visualization, A.S.B., P.J., and M.K.; supervision, A.S.B.; funding acquisition, A.S.B. All authors have read and agreed to the published version of the manuscript.

Funding: This paper was co-financed under the research grant of the Warsaw University of Technology supporting scientific activity in the discipline of Civil Engineering, Geodesy, and Transport.

Data Availability Statement: Open research data is available in the Warsaw University of Technology Knowledge Base at <https://doi.org/10.71724/be6h-n995>.

Acknowledgments: The authors would like to thank the reviewers for their feedback, insightful comments, and assistance in improving this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Borkowski, A.S. A Literature Review of BIM Definitions: Narrow and Broad Views. *Technologies* **2023**, *11*, 176. [[CrossRef](#)]
2. Beetz, J.; van Leeuwen, J.; de Vries, B. IfcOWL: A Case of Transforming EXPRESS Schemas into Ontologies. *Artif. Intell. Eng. Des. Anal. Manuf.* **2009**, *23*, 89–101. [[CrossRef](#)]
3. Borgo, S.; Masolo, C. Foundational Choices in DOLCE. In *Handbook on Ontologies*, 2nd ed.; Staab, S., Studer, R., Eds.; International Handbooks on Information Systems; Springer: Berlin/Heidelberg, Germany, 2009; pp. 361–381. [[CrossRef](#)]
4. Rasmussen, M.H.; Lefrançois, M.; Schneider, G.F.; Pauwels, P. BOT: The Building Topology Ontology of the W3C Linked Building Data Group. *Semant. Web* **2021**, *12*, 143–161. [[CrossRef](#)]
5. Jia, J.; Ma, H.; Zhang, Z. Integration of Industry Foundation Classes and Ontology: Data, Applications, Modes, Challenges, and Opportunities. *Buildings* **2024**, *14*, 911. [[CrossRef](#)]
6. Borkowski, A.S.; Maroń, M. Semantic Enrichment of Non-graphical Data of a BIM Model of a Public Building from the Perspective of the Facility Manager. *Big Data Cogn. Comput.* **2024**, *8*, 138. [[CrossRef](#)]

7. Bloch, T. Connecting Research on Semantic Enrichment of BIM—Review of Approaches, Methods and Possible Applications. *J. Inf. Technol. Constr.* **2022**, *27*, 416–440. [[CrossRef](#)]
8. Porello, D.; Vieu, L.; Terkaj, W.; Borgo, S.; Compagno, F.; Sanfilippo, E.M. DOLCE in OWL: The Core Theory. In *Joint Ontology Workshops (JOWO 2024)*; CEUR-WS: Enschede, The Netherlands, 15–19 July 2024; p. 3882.
9. Arp, R.; Smith, B.; Spear, A.D. *Building Ontologies with Basic Formal Ontology*; The MIT Press: Cambridge, MA, USA, 2015; ISBN 978-0-262-52781-1.
10. Guizzardi, G.; Botti Benevides, A.; Fonseca, C.M.; Porello, D.; Almeida, J.P.A.; Prince Sales, T. UFO: Unified Foundational Ontology. *Appl. Ontol.* **2022**, *17*, 167–210. [[CrossRef](#)]
11. Rasmussen, M.H.; Pauwels, P.; Lefrançois, M.; Schneider, G.F.; Hviid, C.A.; Karlshøj, J. Recent Changes in the Building Topology Ontology. In Proceedings of the 5th Linked Data in Architecture and Construction Workshop (LDAC 2017), Dijon, France, 13–15 November 2017; pp. 1–13.
12. Rasmussen, M.H.; Lefrançois, M.; Bonduel, M.; Hviid, C.A.; Karlshøj, J. OPM: An Ontology for Describing Properties That Evolve over Time. In Proceedings of the 6th Linked Data in Architecture and Construction Workshop (LDAC 2018), London, UK, 19–21 June 2018; CEUR Workshop Proceedings: Bonn, Germany, 2018; Volume 2159, pp. 24–33.
13. Werbrouck, J.; Pauwels, P.; Beetz, J.; Verborgh, R.; Mannens, E. ConSolid: A Federated Ecosystem for Heterogeneous Multi-Stakeholder Projects. *Semant. Web* **2024**, *15*, 429–460. [[CrossRef](#)]
14. Boje, C.; Guerriero, A.; Kubicki, S.; Rezgui, Y. Towards a Semantic Construction Digital Twin: Directions for Future Research. *Autom. Constr.* **2020**, *114*, 103179. [[CrossRef](#)]
15. Jiang, L.; Shi, J.; Wang, C.; Pan, Z. Intelligent Control of Building Fire Protection System Using Digital Twins and Semantic Web Technologies. *Autom. Constr.* **2023**, *147*, 104728. [[CrossRef](#)]
16. Hamdan, A.-H.; Bonduel, M.; Scherer, R.J. An Ontological Model for the Representation of Damage to Constructions. In Proceedings of the 7th Linked Data in Architecture and Construction Workshop (LDAC 2019), Lisbon, Portugal, 19–21 June 2019; pp. 64–77.
17. Cox, S.; Little, C. Time Ontology in OWL; W3C Recommendation. 19 October 2017. Available online: <https://www.w3.org/TR/owl-time/> (accessed on 10 June 2026).
18. Bonduel, M.; Oraskari, J.; Pauwels, P.; Vergauwen, M.; Klein, R. The IFC to Linked Building Data Converter: Current Status. In Proceedings of the 6th Linked Data in Architecture and Construction Workshop (LDAC 2018), London, UK, 19–21 June 2018; CEUR Workshop Proceedings: Bonn, Germany; Volume 2159, pp. 34–43.
19. Autodesk. Autodesk Revit 2026. Available online: <https://www.autodesk.com/products/revit/> (accessed on 14 March 2026).
20. Musen, M.A. The Protégé Project: A Look Back and a Look Forward. *AI Matters* **2015**, *1*, 4–12. [[CrossRef](#)] [[PubMed](#)]
21. Eastman, C.; Teicholz, P.; Sacks, R.; Liston, K. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2011; ISBN 978-0-470-54137-1.
22. Hollaway, L.C.; Teng, J.G. *Strengthening and Rehabilitation of Civil Infrastructures Using Fibre-Reinforced Polymer (FRP) Composites*; Woodhead Publishing: Cambridge, UK, 2008; ISBN 978-1-84569-448-7.
23. Suárez-Figueroa, M.C.; Gómez-Pérez, A.; Fernández-López, M. The NeOn Methodology for Ontology Engineering. In *Ontology Engineering in a Networked World*; Suárez-Figueroa, M.C., Gómez-Pérez, A., Motta, E., Gangemi, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 9–34. [[CrossRef](#)]
24. Uschold, M.; King, M. Towards a Methodology for Building Ontologies. In Proceedings of the Workshop on Basic Ontological Issues in Knowledge Sharing, Held in Conjunction with IJCAI-95, Montreal, Canada, 20–25 August 1995. AIAI-TR-183.
25. Borkowski, A.S. A critical analysis of the direction of BIM development: From CAD through file-based approaches and SaaS to granular data. *J. Inf. Technol. Constr.* **2026**, *31*, 380–393. [[CrossRef](#)]
26. Kent, M.G.; Gao, Y.; Mosalam, K.M. Operational Energy Consumption in a Zero-Energy Office Building in Singapore. *J. Build. Eng.* **2025**, *116*, 114641. [[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.