



## INTRODUCTION OF THE ONTOLOGY FOR CHRONOLOGICAL CONSTRUCTION PROCESSES (OCCP)

placeholder for the names of the authors

<sup>1</sup>Technical University, Cambridge, MA

<sup>2</sup>Another Institution, Some City, Some Country

The names and affiliations MUST NOT be included in the draft submitted for review.

The header consists of 10 lines with exactly 14 point spacing.

The last line below should be left blank

### Abstract

The Ontology for Chronological Construction Processes (OCCP) provides a semantic foundation for a chronological structure aimed at managing information in the Architecture, Engineering, Construction and Operations (AECO) sector. Based on the W3C TIME ontology (Cox et al., 2006), the OCCP introduces a lifecycle-oriented concept tailored to Building Information Modelling (BIM) workflows and advances the accurate management of time stamps by introducing phases, cycles, milestones and transitions to ensure traceability of states throughout and beyond the lifetime of assets. This paper describes the concept and composition of OCCP, its SHACL-based validation rules, and provides an example to demonstrate its practical application.

### Introduction

The AECO sector faces growing challenges in managing complex project information effectively. While Building Information Modeling (BIM) provides a structured approach, current workflows lack robust mechanisms for integrating and utilizing time-sensitive data. Existing standards like Industry Foundation Classes (IFC) (ISO 16739-1) and Information Container for Linked Document Delivery (ICDD) (ISO 21597-1) ensure spatial and semantic interoperability, but their support for temporal data remains fragmented. This results in disconnected records of planned and actual timelines, leading to errors, inefficiencies, and reduced traceability (Autodesk, 2018). Current approaches, such as XML and relational databases, provide only basic timestamp storage, failing to capture semantic relationships between lifecycle stages. They also struggle with scalability, consistency, and integration within BIM and multimodal frameworks, issues that graph-based ontologies address more effectively (Berners-Lee et al., 2001). The W3C TIME ontology provides a foundational model for representing time-related information, but lacks domain-specific constructs needed for chronological consistency, overlapping phases, iterative cycles, and project-specific adaptations. The Ontology for Chronological Construction Processes (OCCP) introduces a structured

approach for managing construction lifecycles by defining phases, cycles, transitions, and instants, with SHACL rules ensuring logical sequencing and data integrity. Designed for seamless integration with BIM and ICDD, OCCP enhances temporal data management, lifecycle validation, and modeling flexibility. This paper presents its core structure, logic, and applications, demonstrating its potential to transform time-data management in the AECO sector.

### State of the art

The AECO sector increasingly relies on digital tools and standards to manage complex building projects. IFC, developed by buildingSMART International, has become the de facto standard for BIM interoperability, supporting geometric, semantic, and topological data exchange across disciplines (Sacks et al., 2018; buildingSMART International, 2023). While IFC provides detailed lifecycle management, it lacks robust support for dynamic temporal and process-oriented data (Pauwels et al., 2017). Multimodels emerged to address the limitations of single, centralized models (Scherer & Schapke, 2011). These integrate multiple specialized data sources (e.g., cost, schedule, energy models) into a unified project representation (Borrmann et al., 2018; Grilo et al., 2010). By linking different data layers, multimodal containers enhance information accessibility (Xie et al., 2024), but their lack of semantic integration and validation mechanisms can lead to inconsistencies and reduced reliability (Singh et al., 2011). Similarly, while ICDD (ISO 21597-1) improves traceability and version control, it does not inherently support semantic reasoning or structured chronological data management (Pauwels et al., 2016). Ontologies, expressed using RDF (Lassila & Swick, 1999), provide a powerful approach for structuring, validating, and reasoning about BIM data (Beetz et al., 2009). They improve data consistency and interoperability, making them suitable for semantic integration. The W3C TIME Ontology (Cox et al., 2006) is widely used for temporal data representation, supporting instants, intervals, and relationships such as time:Before and time:After. It enables integration with IfcOWL and ICDD for semantic temporal modeling (Pauwels et al., 2016) and facilitates advanced temporal

queries, such as detecting overlapping project phases (Car et al., 2024). However, TIME lacks domain-specific constructs for AECO workflows, such as phases, iterative cycles, and planned vs. actual timestamps (Volk et al., 2014). Furthermore, it does not inherently validate chronological consistency, making it difficult to detect conflicts like overlapping timelines (Karlupudi et al., 2021). To address these shortcomings, SHACL enables the validation of RDF-based temporal structures (Knublauch et al., 2017; Debruyne et al., 2020), allowing the enforcement of logical constraints in AECO workflows. However, current IFC implementations still lack structured temporal representations, leading to fragmented data across multiple tools (Volk et al., 2014; Pauwels, 2017). Similarly, ICDD and multimodal containers structure heterogeneous data but lack inherent mechanisms for ensuring temporal consistency (Singh et al., 2011). Versioning systems, such as Global Information Tracker (GIT), Apache Subversion (SVN), and Ontology-based Programming Models (OPM), are widely used for tracking changes in software development and ontologies. However, these methods are not inherently designed for structured chronological modeling in AECO workflows, where updates must be linked to specific lifecycle events. The OCCP addresses this gap by integrating chronological records directly within the semantic model. Through the use of IFC-linked timestamps, OCCP captures the evolution of a component across different versions, providing a bidirectional version tracking system that can be used in combination with GIT-based repositories. This enables a fine-grained traceability approach, allowing stakeholders to track changes within both the IFC model and its associated temporal records, ensuring complete transparency in model evolution.

## The core concept of OCCP

The OCCP is based on the W3C TIME ontology and extends its classes and object properties to provide a semantic representation of the building lifecycle. This ontology is part of the concept of chronological models (cMod) (Vaatz et al., 2023), which combines IFC-based models with the OCCP and other ontologies, such as the Bridge Topology Ontology (BROT) (Hamdan et al., 2020) or the Building Topology Ontology (BOT) (Rasmussen et al., 2020), and model-related data sets using ICDD. In this concept, the OCCP provides the temporal structure for storing the timestamps of model-related data and assigning the correct phase or milestone within the lifecycle. The general temporal structure of the OCCP, as shown in Figure 1, is designed to ensure applicability to a wide range of AECO projects by not adapting any national specific construction process definitions such as the German HOAI (Honorarordnung für Architekten und Ingenieure).

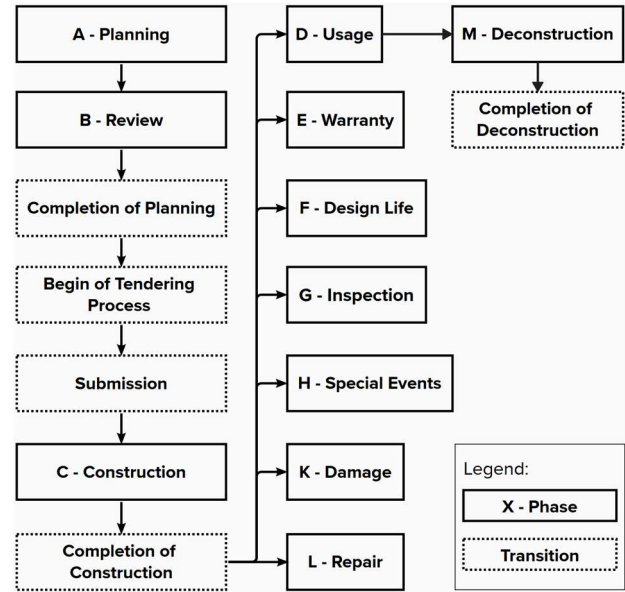


Figure 1: Overview of the temporal structure of the OCCP

## Methodology of development

The development of the OCCP followed a structured methodology based on domain-specific requirements, expert insights, and semantic validation. Given the complexity of AECO workflows, the ontology was designed to represent real-world temporal constructs, lifecycle transitions, and iterative review cycles while ensuring adaptability across different project structures by extending the W3C TIME ontology with specialized classes and properties, enabling dependency tracking, logical sequencing, and versioned event recording. Industry expert input further refined OCCP's classifications, ensuring it accommodates iterative planning, review processes, and real-world deviations such as delays, rework cycles, or phased repairs. OCCP's structure and reasoning capabilities were validated using a competency question-driven approach, where key queries tested the ontology's ability to structure, retrieve, and enforce chronological consistency. These competency questions were formalized into SHACL rules, ensuring correct sequencing of phases, review cycles before approvals, and structured tracking of IFC updates.

Among others, the following three competency questions guided the validation of OCCP's semantic constraints:

CQ1: How many review cycles occurred before approval, and how did this impact the planned vs. actual completion time?

CQ2: What changes were made to an IFC component during planning and review?

CQ3: When was an event recorded, and how many updates were made?

These questions demonstrate the information demand that OCCP is designed to answer and OCCP's ability to track changes, compare planned vs. actual schedules, and maintain structured, machine-readable chronological records within BIM and multimodal environments.

## Classes

The W3C TIME ontology defines temporal entities, distinguishing between `time:Instant` (specific points in time without duration) and `time:Interval` (spans between two instants). The OCCP extends the TIME ontology by introducing `occp:Phase` and `occp:Cycle` as subclasses of `time:Interval`. All phases shown in Figure 1 (e.g., the `occp:PhaseE_Usage`) are subclasses of `occp:Phase`, while cycles, such as `occp:CycleA_PlanningReview`, capture iterative processes like multiple review cycles in planning.

To structure milestone events, OCCP extends `time:Instant` by defining `occp:Transition` and phase-related instants. Transitions mark key milestones between phases or trigger subsequent project steps, such as `occp:Submission`, which signals the transition from planning to construction. Similarly, `occp:CompletionOfPlanning` defines the finalization of planning, ensuring proper sequencing of activities. Phase-related instants, such as `occp:EditBSTR` (BSTR - before submission to review) or `occp:BeginningOfPlanning`, provide precise temporal markers within each phase. Following this pattern, OCCP introduces dedicated event classes for each defined phase. Figure 2 illustrates the OCCP class hierarchy, showcasing its extension of the TIME ontology, while Figure 3 provides a detailed representation of the temporal framework for the first three phases, their transitions, and associated instants. Figure 3 highlights OCCP's structured approach to temporal modeling, demonstrating how `occp:PhaseA_Instant`, `occp:PhaseB_Instant`, and `occp:PhaseC_Instant` organize phase-specific instants within project lifecycles.

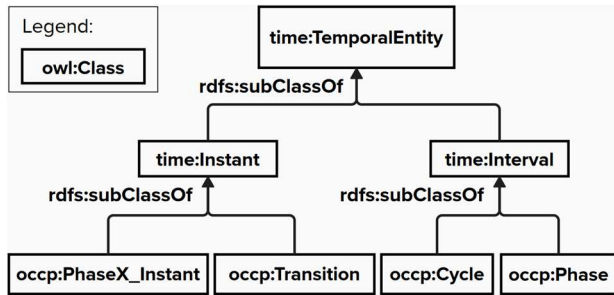


Figure 2: General class hierarchy of OCCP

The `occp:BeginningOfPlanning` serves as a foundational event marking the start of planning, while `occp:SubmissionToReview` signifies the handoff between planning and review. Transitions such as `occp:Submission` and `occp:CompletionOfConstruction` define clear boundaries between phases, supporting chronological consistency and milestone validation. The subclass relationships in Figure 3 illustrate how OCCP seamlessly integrates into the TIME ontology while introducing domain-specific instants and transitions. The ontology's modular structure enhances semantic clarity, distinguishing between different

temporal entities through explicit object properties. This modularity ensures adaptability to diverse project needs, supporting both high-level lifecycle planning and granular activity tracking.

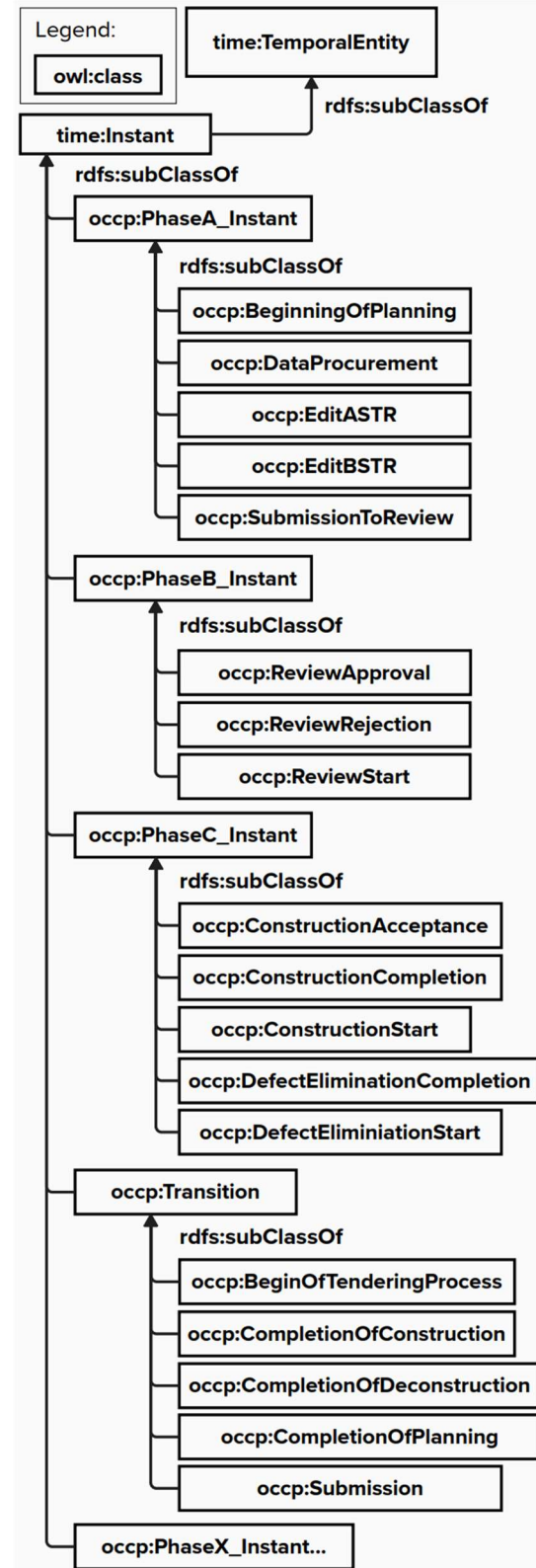


Figure 3: Extract of phase-related instant classes of OCCP

## Object properties

The TIME ontology provides fundamental object properties for describing general relationships between `time:Instant` and `time:Interval`, but lacks the expressiveness needed for structured lifecycle management in AECO projects. The OCCP extends these capabilities by introducing additional object properties, as shown in Figure 4, refining both the `owl:topObjectProperty` and `time:hasTime` (including `time:hasBeginning` and `time:hasEnd`) to define chronological relationships between phases, cycles, milestones, and instances. To establish clear temporal associations, OCCP introduces `occp:hasPhase` and `occp:isInPhase`. The `occp:hasPhase` property embeds events within structured lifecycle phases, ensuring temporal instances are correctly classified. For example, `occp:SubmissionToReview`, which represents the submission of a completed plan for review, is explicitly linked to Phase A (Planning), maintaining a logically ordered structure for querying and validation. This ensures event traceability and enables automated reasoning over project lifecycles. The `occp:isInPhase` property, in contrast, captures nested or overlapping processes within phases. A practical example is the repair of a damaged component during the Usage phase, where the repair lifecycle must be contextualized within the broader phase. The `occp:isInPhase` relationship enables the representation of concurrent activities, acknowledging the reality of complex, interwoven construction processes. Beyond these structural properties, OCCP also refines temporal granularity by distinguishing between actual and estimated time values. The `occp:hasActualTime` and `occp:hasEstimatedTime` properties differentiate between verified timestamps and planned or uncertain events. Similarly, `occp:hasActualBeginning` and `occp:hasEstimatedEnd` extend `time:hasBeginning` and `time:hasEnd`, allowing precise modeling of both anticipated and confirmed project timelines. To enforce logical sequencing, OCCP introduces relational temporal properties such as `occp:beginsBefore`, `occp:beginsAfter`, `occp:beginsWith`, `occp:endsBefore`, `occp:endsAfter`, and `occp:endsWith`, enabling the semantic representation of phase dependencies and overlaps. Additionally, hierarchical lifecycle properties (`occp:startsCycle`, `occp:endsCycle`, `occp:startsPhase`, and `occp:endsPhase`) structure iterative processes, supporting the modular definition of project milestones, transitions, and recurrent events. This enriched semantic framework ensures that OCCP not only tracks time-related data, but also structures, validates, and integrates it within complex lifecycle models, supporting flexibility, consistency, and traceability in construction project management.

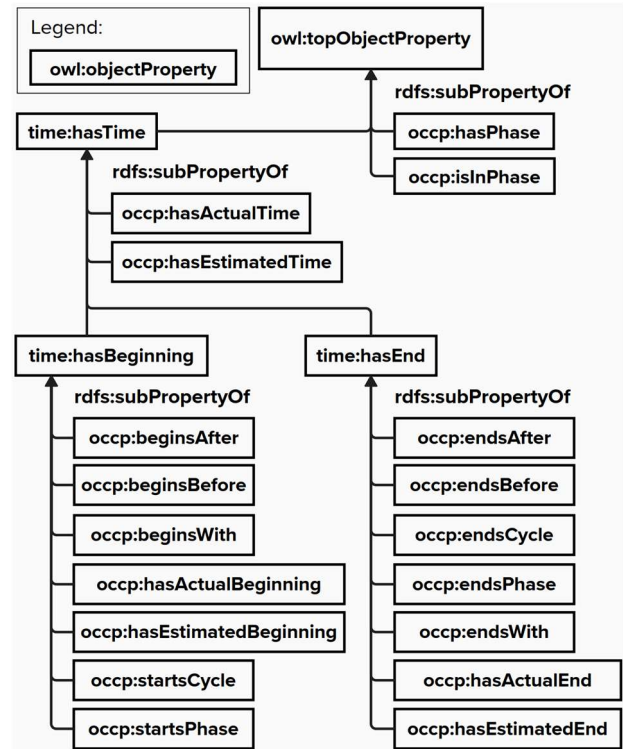


Figure 4: Object properties of the OCCP

## SHACL ruleset

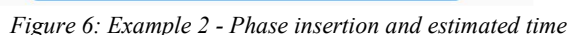
The SHACL ruleset embedded in OCCP plays a critical role in validating and ensuring the logical consistency of temporal and semantic data. By defining and enforcing structural and semantic constraints on RDF data, SHACL ensures that relationships and properties within OCCP adhere to predefined logical rules. This validation layer bridges the gap between theoretical modeling and practical implementation, enabling robust verification of temporal and semantic correctness in an OCCP-enhanced framework.

A primary function of the SHACL ruleset is to maintain the integrity of OCCP's temporal structure by governing relationships between phases, transitions, and instants. Rules enforce sequential and hierarchical order, ensuring that each phase has a well-defined beginning and end, validated using properties such as `occp:hasActualBeginning` and `occp:hasActualEnd`. This mechanism prevents erroneous overlaps or gaps between phases—an essential requirement in multi-stakeholder construction projects, where precise chronology is crucial for lifecycle management. A distinct feature of OCCP's SHACL ruleset is its ability to validate iterative and nested temporal structures. Within the planning phase, for instance, iterative cycles such as `occp:CycleA_PlanningReview` regulate review processes. SHACL shapes ensure that each iteration only begins upon a defined trigger event, such as `occp:ReviewRejection`, and concludes with either another iteration or `occp:ReviewApproval`. This ensures





Example 2 extends this logic by embedding the planning-review cycle from Figure 5 into a repair phase using `occp:isInPhase`, creating a nested lifecycle. This scenario, applicable to cases such as structural damage repair, could represent a bridge bearing replacement, where the damage is recorded through inspections and reaches a critical threshold, requiring immediate intervention. The repair lifecycle, containing its own planning, review, and execution phases, coexists with and impacts the broader usage phase (`occp:PhaseD_Usage`), affecting the overall bridge operation. OCCP enables clear documentation of these interdependencies, ensuring full traceability of how the repair process affects and integrates with the broader lifecycle of the structure.



highlight OCCP’s capability to structure, validate, and adapt lifecycle representations across varied use cases. The ontology’s SHACL ruleset enforces compliance with defined chronological relationships, ensuring logical consistency while allowing flexibility for diverse project requirements.

The example shown in Figures 5 and 6 illustrates OCCP's ability to structure lifecycle events and track updates to IFC components within planning, review, and repair cycles. The following competency questions further demonstrate how OCCP provides structured answers to key project-related queries:

1. How many review cycles occurred before approval, and how did this impact the planned vs. actual completion time?
  - OCCP records each iteration of a review cycle using `occp:CycleA_PlanningReview`, which can be counted to determine the number of review attempts before approval.
  - The estimated and actual phase completion times, stored under `occp:hasEstimatedEnd` and `occp:hasActualEnd`, allow for direct comparisons of planned vs. actual timelines.
2. What changes were made to an IFC component during planning and review?
  - Updates to IFC elements are recorded as instances of `occp:Update`, linking components to modification timestamps (`occp:hasUpdateTime`) and property changes (`occp:hasUpdatedProperty`).
  - The ontology maintains a record of previous (`occp:hasPreviousValue`) and new (`occp:hasNewValue`) values for each change, ensuring full traceability.
3. When was an event recorded, and how many updates were made?
  - The first recording of an event is tracked using `occp:hasRecordingTime`, while subsequent modifications are logged with `occp:hasUpdateTime`, allowing for a reconstruction of event history.

```
ex:Update123 a occp:Update ;
    occp:hasUpdateTime
        "2025-02-01T12:00:00"^^xsd:dateTime ;
    occp:hasUpdatedEvent occp:ReviewApproval ;
    occp:hasPreviousIFCIdentifier "1A2B3C4D5E" ;
    occp:hasNewIFCIdentifier "9X8Y7Z6W5V" ;
    occp:hasUpdatedProperty occp:hasEstimatedTime ;
    occp:hasPreviousValue
        "2025-01-15T08:00:00"^^xsd:dateTime ;
    occp:hasNewValue
        "2025-01-20T08:00:00"^^xsd:dateTime .
```

Figure 7: Example of data structuring in OCCP

reference an updated property. Due to space limitations, only the core validation rules concerning the modification of recorded events, IFC components, and temporal properties are shown. The complete ruleset encompasses additional constraints to ensure chronological consistency across various lifecycle events, covering scenarios such as sequential update dependencies, version tracking across multiple IFC identifiers, and logical validation of estimated versus actual timestamps. These rules collectively reinforce OCCP's ability to maintain structured, machine-readable chronological records while preserving data integrity. By enforcing conditions on updates through SHACL, OCCP ensures that changes in recorded lifecycle information remain transparent, verifiable, and systematically integrated within multimodal BIM workflows.

```
ex:UpdateShape a sh:NodeShape ;
  sh:targetClass occp:Update ;
  sh:property [
    sh:path occp:hasUpdateTime ;
    sh:minCount 1 ;
    sh:datatype xsd:dateTime ;
    sh:message "Each update must include a
    timestamp." ;
  ] ;
  sh:property [
    sh:path occp:hasUpdatedProperty ;
    sh:minCount 1 ;
    sh:message "Each update must specify the
    property that was modified." ;
  ] .
```

Figure 8: Minimal example of SHACL rules for updates

## Outlook and conclusion

The OCCP is nearing public release, including its source code and documentation, marking a significant milestone in its development. The next phase focuses on its practical implementation within chronological models (cMod), designed to unify lifecycle-based temporal data with broader lifecycle information management (LIM) systems. The cMod framework will operationalize OCCP's semantic structure, enabling real-time validation, version tracking, and predictive analysis. By linking model-specific OCCP (MsOCCP) datasets to IFC models and supporting graph-based reasoning, cMod will facilitate automated consistency checks and lifecycle simulations. A dedicated software interface is under development to support cMod creation, management, and model-stage visualization. OCCP extends the W3C TIME ontology with domain-specific constructs such as phases, cycles, transitions, and phase-specific instants, providing a structured framework for integrating time-based data into BIM workflows. Its SHACL-based validation ensures logical consistency, minimizing errors and improving data reliability. By supporting iterative cycles, overlapping phases, and complex lifecycle structures,

OCCP enhances timeline visibility, traceability, and automated consistency checks. The introduced update tracking mechanism strengthens OCCP's ability to record, query, and validate IFC modifications, ensuring seamless version tracking and bidirectional synchronization with Git-based repositories. Furthermore, OCCP was tested against domain-relevant competency questions, demonstrating its capability to retrieve key lifecycle insights, validate planned vs. actual schedules, and track iterative processes such as review cycles and component modifications. The competency-question-driven approach, supported by SHACL validation, ensures that OCCP does not merely store time-related data but actively verifies chronological consistency and logical sequencing. As a flexible and extensible ontology, OCCP contributes to advancing lifecycle-oriented construction informatics. While further refinement is needed to optimize its integration with industry tools, its structured approach to time management in BIM and multimodal workflows provides a solid foundation for improving consistency, traceability, and automation of chronological records. Future work will focus on expanding its application, enhancing interoperability, and evaluating its real-world impact through practical case studies and software implementations.

## References

- Autodesk (2018) Construction Disconnected: The High Cost of Poor Data and Miscommunication. <https://www.autodesk.com/blogs/construction/construction-disconnected-fmi-report/> . Accessed date: 30th January 2025.
- Beetz, J., van Leeuwen, J. P., & de Vries, B. (2009) IfcOWL: A case of transforming EXPRESS schemas into ontologies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, 23(1), 89-101. DOI: 10.1017/S0890060409000122
- Berners-Lee, T., Hendler, J. & Lassila, O. (2001) The Semantic Web. *Scientific American*, 284(5), 34-43. <https://doi.org/10.1038/scientificamerican0501-34>
- Borrmann, A., König, M., Koch, C., & Beetz, J. (2018) *Building Information Modeling: Technology Foundations and Industry Practice*. Springer International Publishing. DOI: 10.1007/978-3-319-92862-3
- buildingSMART International (2023) *Industry Foundation Classes (IFC) – Standard*. <https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/> . Accessed date: 30th January 2025.

- Car, N.J., Homburg, T., Perry, M., Knibbe, F., Cox, S.J.D., Abhayaratna, J., Bonduel, M., Cripps, P.J., & Janowicz, K. (2024). OGC GeoSPARQL - A Geographic Query Language for RDF Data (Version 1.1). Open Geospatial Consortium (OGC). <https://docs.ogc.org/is/22-047r1/22-047r1.html>. Accessed date: 29th January 2025.
- Cox, S. & Little, C. (2006) Time Ontology in OWL. <https://www.w3.org/TR/owl-time/>. Accessed date: 29th January 2025.
- Debruyne, C. & McGlinn, K. (2020). Reusable SHACL Constraint Components for Validating Geospatial Linked Data. In: GeoLD@ESWC 2020 – Proceedings of the 1st International Workshop on Geospatial Linked Data. Heraklion, Greece. <https://chrdebru.github.io/papers/2021-geold-preprint.pdf>. Accessed date: 30th January 2025.
- Fuchs, S., Kaddolsky, M. & Scherer, R. J. (2011) Formal description of a generic multi-model. In 2011 IEEE 20th International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 205-210.
- Grilo, A., & Jardim-Goncalves, R. (2010) Value proposition on interoperability of BIM and collaborative working environments. *Automation in Construction*, 19(5), 522-530. DOI: 10.1016/j.autcon.2009.11.003
- Hamdan, A. H. & Scherer, R. J. (2020) Integration of BIM-related bridge information in an ontological knowledgebase. In: Proceedings of the 8th Linked Data in Architecture and Construction Workshop (LDAC)
- Karlapudi, J., Valluru, P., & Menzel, K. (2021). Ontology Approach for Building Lifecycle Data Management. Proceedings of the ASCE International Conference on Computing in Civil Engineering (i3CE2021). Orlando, Florida, USA.
- Knublauch, H. & Kontokostas, D. (2017). Shapes Constraint Language (SHACL). <https://www.w3.org/TR/shacl/>. Accessed date: 30th January 2025.
- Lassila, O., & Swick, R. R. (1999). *Resource Description Framework (RDF) Model and Syntax Specification*. W3C Recommendation. <https://www.w3.org/TR/1999/REC-rdf-syntax-19990222/>. Access date: 30th January 2025.
- Pauwels, P., Zhang, S., & Lee, Y.-C. (2017). Semantic web technologies in AEC industry: A literature overview. *Automation in Construction*, 73, 145–165. <https://doi.org/10.1016/j.autcon.2016.10.003>
- Pauwels, P., & Terkaj, W. (2016) EXPRESS to OWL for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction*, 63, 100-133. DOI: 10.1016/j.autcon.2015.12.003
- Rasmussen, M. H., Lefrançois, M., Schneider, G. & Pauwels, P. (2020) BOT: the Building Topology Ontology of the W3C Linked Building Data Group. In: Semantic Web. DOI: 10.3233/SW-200385.
- Sacks, R., Eastman, C., Lee, G., & Teicholz, P. (2018). *BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers* (3rd ed.). John Wiley & Sons. DOI: 10.1002/9781119287568
- Scherer, R.J., & Schapke, S.-E. (2011). A distributed multi-model-based management information system for simulation and decision-making on construction projects. *Advanced Engineering Informatics*, 25(4), 582–599.
- Singh, V., Gu, N. & Wang, X. (2011) A Theoretical Framework of a BIM-Based Multi-Disciplinary Collaboration Platform. *Automation in Construction*, 20(2), 134-144. DOI: 10.1016/j.autcon.2010.09.011
- Vaatz, A., Hamdan, A.-H., Al-Sadoon, N., Wogan, M., Menzel, K., (2023). Integration of semantic temporal information in BIM using ontologies. European Conference on Computing in Construction. Crete, Greece, July 10-12. DOI: 10.35490/EC3.2023.281
- Volk, R., Stengel, J., & Schultmann, F. (2014). Building Information Modeling (BIM) for existing buildings — Literature review and future needs. *Automation in Construction*, 38, 109–127. <https://doi.org/10.1016/j.autcon.2013.10.023>
- Xie, Y., Zhan, N., Zhu, Q., Zhan, J., Guo, Z., Qiao, C., Zhu, J., & Xu, B. (2024). Multimodal data visualization method for digital twin campus construction. *International Journal of Digital Earth*, 17(1), 145–165. <https://doi.org/10.1080/17538947.2024.2431624>