

Towards an IFC-based Construction Process Modeling Language

Jesús Muñoz-Cádiz¹[0000-0003-1386-5743], Gunakar Challa¹[0009-0007-2697-0215],
Simon Curty¹[0000-0002-2868-9001], and Hans-Georg Fill¹[0000-0001-5076-5341]

University of Fribourg, Research Group Digitalization and Information Systems, 1700
Fribourg, Switzerland

{jesus.munozcadiz,gunakar.challa,simon-curty,hans-georg.fill}@unifr.ch

Abstract. In this paper, we introduce a conceptual integration that maps the data product modeling language represented by the Industry Foundation Classes (IFC) to the Construction Process Modeling Language (CoPModL). The implementation is carried out on the ADOxx meta-modeling platform, emphasizing the CoPModL behavioral constraints, which are a common challenge in Building Information Modeling (BIM) environments. The applicability of our proposal is tested using an official buildingSMART International dataset.

Keywords: Construction Process · Industry Foundation Classes · Domain-specific Modeling Languages.

1 Introduction

Construction projects involve multiple stakeholders, including owners, contractors, engineers, architects, and different institutions, who collaborate with each other throughout the life-cycle of the project. As all of them work on the same built asset, there is a strong need to structurally store and share the data between them regarding the building and the activities they are going to perform on it. In addition, different stakeholders require a different level of detail of the project's components, which affects both the geometry [8] and the shared data [33,6]. The level of complexity increases as different activities are performed on several different items, potentially altering the state of those building elements.

Accordingly, this scenario requires the use of four-dimensional (4D) tools for project stage management. The Building Information Modeling (BIM) methodology addresses this need by providing a set of functionalities, including planning, scheduling, and three-dimensional (3D) simulations for the evaluation of construction sequences. However, a structural gap remains between product modeling, as represented by the Industry Foundation Classes (IFC), and declarative process modeling languages applied in BIM workflows. The IFC schema provides spatial containment hierarchies and structured representations of building components, but does not formally specify execution constraints. Conversely, process modeling languages symbolically represent abstract spatial structures but do not natively leverage the rich structural semantics of IFC models.

As a result, the definition of construction processes often remains disconnected from the detailed product information available in BIM models, limiting the validation and reuse of execution constraints across project stages. The research presented in this paper addresses this gap by proposing a metamodel that aligns IFC-based product modeling with the Construction Process Modeling Language (CoPModL). The contribution is conceptual and methodological: rather than transforming IFC into a full ontology or extending CoPModL with geometric reasoning, we integrate both modeling paradigms at the metamodel level. This enables detailed constraint-aware modeling of construction processes based on IFC semantics without requiring external ontology-based reasoning or intermediate IFC conversions.

We implement this approach within the ADOxx metamodeling platform [13], thereby enabling the initial definition of a domain-specific modeling language (DSML) that integrates structural (product) and behavioral (process) elements within a coherent modeling framework. This decision is supported by the lack of IFC-based frameworks for metamodeling implementations, followed by the need to include constraints on order, precedence, and scoped execution.

The remainder of the paper is structured as follows. Section 2 presents the foundations and related work, covering IFC, construction process modeling, and modeling languages. Section 3 describes language design and implementation, including requirements, metamodel definition, and implementation details. Section 4 presents a case study, enabling a consistency analysis of our approach in the context of a real BIM project. Section 5 discusses implications and limitations, and Section 6 concludes with future research directions.

2 Foundations and Related Work

This section presents the conceptual foundations and related work underlying our research. It first introduces the metamodeling perspective adopted in this study, based on the modeling method framework described by [18,13]. The method provides a general basis for the design of domain-specific modeling languages (DSML) across different disciplines, which we apply to the Architecture, Engineering, and Construction (AEC) industry. Then, it provides an analysis of the construction processes, which are characterized by heterogeneous practices and only partially standardized management approaches. Due to the complexity of the field, the section discusses existing approaches to construction processes and examines the EXPRESS modeling language and IFC schema as the formal basis for representing product-related building information. Together, these aspects define the theoretical context for the metamodel proposed in this paper.

2.1 Construction Processes

Construction projects are inherently process-oriented systems characterized by temporal dependencies and spatial constraints. Unlike manufacturing, where processes are often repetitive and structured, construction processes are site-specific,

and technical requirements make them highly dependent on the context. Modeling such processes requires not only temporal ordering but also consideration of spatial scope and multi-instance execution. The BIM methodology has significantly advanced these technical aspects by transforming the representation and management of information during the project’s lifecycle.

This way, a conventional project division includes two interconnected phases: the delivery and operational phase. Delivery includes steps such as design (conceptual first, and detailed after), documentation, and construction. Finally, the operational phase includes processes such as operation and maintenance, and facility management. In response to this degree of specialization at each stage, and to enhance interoperability, buildingSmart International (bSI)¹ develops and establishes standards for application within the BIM framework.

Classical construction process planning is largely based on network scheduling techniques such as Gantt charts, the Critical Path Method (CPM), and the Program Evaluation and Review Technique (PERT) [19]. Some of these methods, already tested in BIM frameworks [35,16], represent activities as nodes or edges in directed acyclic graphs and compute earliest and latest start times under precedence constraints. Although effective for time optimization, these techniques focus on deterministic scheduling rather than on semantic modeling of process coordination. Implementing it, we assume that precedence relations are explicitly enumerated and globally applicable.

However, construction processes often involve scoped dependencies, such as precedence relations restricted to a building level, zone, or sector. In this regard, traditional scheduling frameworks provide limited formal support for the expression of such context-sensitive constraints.

2.2 Construction Modeling Languages

In the field of construction, modeling languages are often discussed in terms of product modeling, process modeling, and DSML. In the context of process modeling, expressive formalisms have been adopted, including Petri Nets [25] and Business Process Model and Notation (BPMN) [27]. These general-purpose modeling languages are used to represent the construction process by explicitly enumerating the allowed control-flow paths.

Although these approaches support concurrency and synchronization, they suffer from scalability and complexity when modeling multi-instance construction processes distributed across spatial partitions [4]. For example, authors in [15] propose an agent-based simulation approach based on Petri Nets that derives simulation structures from BIM inputs. However, their work focuses on simulation-driven schedule analysis rather than on the definition of a modeling language or the integration of construction process models with the IFC data structure.

In construction contexts, where the same activity may recur across multiple locations, this approach can lead to large, rigid process models that lose the

¹ <https://www.buildingsmart.org/>

nuances of the AEC industry’s needs. In response, declarative process modeling has emerged as an alternative paradigm, particularly in the process mining community. The Declare framework [30] models processes as sets of constraints over possible execution traces rather than as explicit control-flow graphs.

Constraints are typically formalized using Linear Temporal Logic (LTL), enabling reasoning about admissible traces and model consistency. The theoretical foundation of declarative modeling is grounded in temporal logic and trace semantics [7], which enables reasoning about the consistency, satisfiability, and redundancy of constraints.

2.3 EXPRESS and IFC

The BIM methodology combines multiple standards and tools to support the exchange of geometric information and its associated metadata. The EXPRESS modeling language [32] is used inside the BIM ecosystem, and its rich syntactic constructs are validated as an International Organization for Standardization (ISO) normative [17]. EXPRESS is used to formalize IFC, enabling the structured and exchangeable representation of classes such as building elements (*IfcWall*), relationships (*IfcRelAggregates*), and a set of general and customized properties [31,9]. Although IFC effectively supports product-oriented information modeling and process entities (e.g., *IfcTask*), its core focus lies in the structural and semantic description of built assets rather than in the formalization of process coordination. Furthermore, IFC does not provide a unified formal framework for explicitly modeling temporal dependencies and complex declarative constraints across multiple entities.

Up to the IFC 4.3.2 specification, the design does not include either semantic reasoning or the inference of new relationships. Instead, it represents only explicit data, so any higher-level semantic meaning must be derived through external processing of the model structure. This is one of the main reasons why ontological approaches are added on top of IFC to improve interoperability at different stages [2,29,26]. In this context, the development of IFC 5 represents a significant shift from previous versions by introducing a modular, JSON-based, and component-oriented architecture intended to improve interoperability and scalability². Unlike IFC 4, which relied on the rigid EXPRESS/STEP schema, IFC 5 adopts USD-inspired layering and modularization, making it closer to ontology-driven models while supporting broader domains such as geographic information systems and the integration of other 3D objects, such as point clouds [3].

So far, the maturation process of IFC has been covered in the scientific literature. The IFC architecture, the changes through different versions, and discussions about other standards are deeply analyzed in different works [1,21,20]. Other research lines consist of wider ontological approaches regarding standards, as discussed in [5]. Although recent contributions have focused on improving

² <https://github.com/buildingSMART/IFC5-development>

interoperability between BIM-related modeling formalisms through transformation infrastructures [34], limited attention has been paid to the development of modeling languages that explicitly integrate product structures with declarative construction process constraints.

In other cases, this challenge is addressed by designing a DSML. CoPModL [22,23] represents a significant step in this direction. Designed for the construction process modeling, it separates a configuration layer (defining item structures, attributes, and perspectives) and a flow layer (defining tasks and declarative dependencies). Dependencies in CoPModL are defined with formal semantics grounded in LTL over finite traces, allowing satisfiability checking of construction coordination models. However, CoPModL abstracts spatial structure symbolically and does not directly exploit the rich containment hierarchy provided by IFC models.

The reviewed literature reveals that different gaps remain unsolved. In particular, IFC provides limited support for formal process semantics, while traditional scheduling offers strong capabilities but lacks semantic expressiveness. Additionally, imperative workflows also offer limited flexibility, and declarative process modeling still relies on abstract spatial structures. Consequently, a unified approach that integrates IFC’s structured spatial hierarchy with declarative process constraints remains underexplored. This work addresses this gap at the metamodel level.

3 Language Design and Implementation

Our study first analyzes the EXPRESS modeling language and the IFC schema by regarding its construction within the four-layered metamodeling hierarchy [18], represented in Figure 1. Specifically, the meta-metalevel (L3) is defined at the most abstract layer. Within this level, the EXPRESS metamodel defines the abstract syntax that the EXPRESS modeling language uses [28], which is formally specified in ISO 10303-11 [17]. EXPRESS describes the syntax, i.e., entities, types, attributes, etc., used at the metamodel level (L2). At this stage, the IFC Schema, written in EXPRESS, defines classes, relationships, and logical semantics used in BIM. Below that, we can mention the Model layer (L1) and the IFC instance file, in which we interact with the “BIM model”. An IFC project contains specific building elements, relationships, and property values, among a long list of associated metadata.

In the lower part, we represent the real item (L0). At this level of the hierarchy, we refer to the building in which the intervention will be executed. Still, it can also abstractly represent the future building that will be constructed, in case we are describing steps in the design stage. Given that IFC represents processes but lacks a formal verification, we map processes, extending CoPModL to address the formal temporal-logic semantics and logical satisfiability. To achieve this, we analyze the two components of the process model included in CoPModL, i.e., configuration and flow. Our interpretation of CoPModL must support spatially grounded execution constraints while remaining conceptually aligned with the

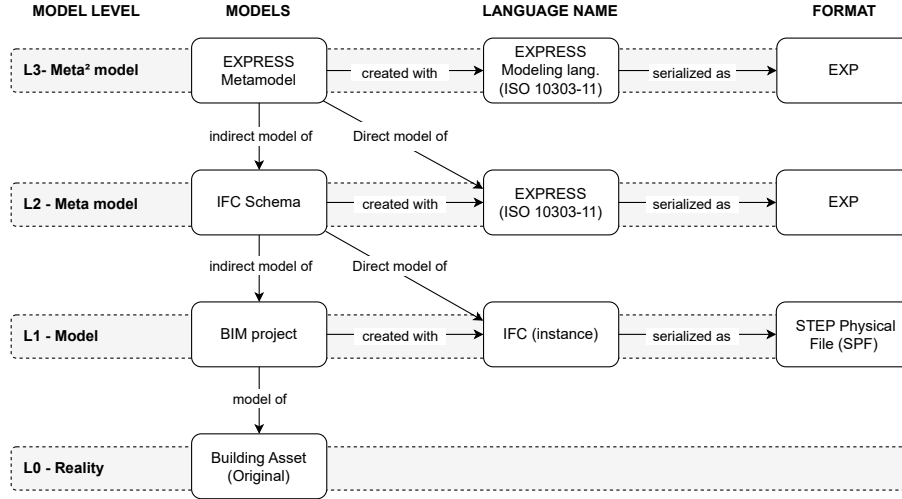


Fig. 1: Metamodeling tree based on the IFC 4X3_ADD2 Schema, including used formats in each level under the EXPRESS modeling language.

hierarchy of IFC at the meta-model level. Due to the size of the IFC schema and the nature of the implementation, we focus on the process extension, considering other extensions included in the core layer³.

To guide the design and conceptualization of the language, we follow the *macro process* by Frank [14], comprised of seven cyclic phases:

1. *Clarification of scope and purpose:* Whilst CoPModL allows the modeling of construction processes, it does not regard IFC hierarchies. Conversely, IFC contains a rich set of concepts to describe a construction project's spatial characteristics but lacks declarative process modeling capabilities. Our approach adapts CoPModL and integrates its process concepts with spatial product configurations expressed with the IFC schema.
2. *Analysis of generic requirements:* We adopt the catalog of generic requirements by Frank [14], applicable to all DSML, but with shifted weighting. These requirements are aligned with our purposes in Section 3.1.
3. *Analysis of specific requirements:* These requirements relate to the developed modeling language in particular. We will outline these in Section 3.1.
4. *Language specification:* The metamodel, presented in the Unified Modeling Language (UML), will be described in Section 3.2.
5. *Design and documentation of graphical notation:* Construction process models use a graphical notation akin to CoPModL, while spatial IFC hierarchy configurations resemble UML class diagrams. The notation will be presented along with the case study in Section 4.

³ The official IFC documentation is accessible through the following link: <https://ifc43-docs.standards.buildingsmart.org/IFC/RELEASE/IFC4x3/HTML/chapter-5/>

6. *Development of modeling tool*: The language was implemented in the ADOxx metamodeling platform. Section 3.3 will detail the implementation.
7. *Evaluation and refinement*: The macro process proposes continuous refinements, e.g., based on modeling scenario evaluation. In Section 4, we use such a scenario, a case study informed by the bSI dataset, to demonstrate the applicability of the language.

3.1 Requirements

The generic requirements are based on the catalog listed in [14] but consolidated and reformulated to fit the goals of this research. This led to four generic requirements (\mathbf{GR}_{1-4}): Language concepts are recognizable and understandable to actors in the AEC sector (\mathbf{GR}_1). Discrete concepts corresponding to spatial or behavioral concerns are distinguishable and invariant in their use (\mathbf{GR}_2). The conceptualization enables the modeling of a rich range of construction processes and spatial hierarchies, containing the relevant information for use in the AEC sector (\mathbf{GR}_3). The language must offer levels of abstraction to prevent overloading models. In particular, the language must segregate structural and process-related concepts (\mathbf{GR}_4).

In addition to these generic requirements, a set of specific requirements was collected and refined through the course of the development process. At this stage, this has yielded four specific requirements (\mathbf{SR}_{1-4}): CoPModL concepts are mapped to equivalent or conceptually similar IFC classes, whenever possible. Table 1 shows these mappings (\mathbf{SR}_1). The language includes concepts for modeling structural aspects, the spatial configuration of the site’s construction processes that operate on. This is represented with IFC product hierarchies, replacing the configuration layer of CoPModL (\mathbf{SR}_2). The language includes concepts for describing processes that remain in adherence with the flows of CoPModL, including the description of dependencies between task items (\mathbf{SR}_3). Models may be subjected to diverse processing algorithms, analyzing processes, and product hierarchies to produce new insights (\mathbf{SR}_4).

3.2 Metamodel

In this section, we define the modeling language as a product of the mapping process. The proposed language integrates the spatial hierarchy of the IFC metadata schema with declarative behavioral constructs inspired by CoPModL. Table 1 illustrates this integration and the differences. The authors of CoPModL have implemented the language in two parts: configuration and flow. In the *configuration part*, the spatial configuration of the construction project is described using declarative constructs at various levels of abstraction (ranging from *Sector* to *Item*). The spatial configuration is utilized in the *flow part*, where it is referenced to specify the location and scope of tasks to be performed. With this approach, logically coherent dependencies between tasks can be modeled. However, the data model of configurations is non-standard, and the contained constructs are non-instantiable.

CoPModL	This work	Description (adapted)
Configuration	IFC Product Hierarchy	
n/a	IfcSite	Area of land on which the project construction is to be completed.
Sector	IfcBuilding	Structure that provides shelter for its occupants.
Level	IfcBuildingStorey	Horizontal aggregation of spaces that are vertically bound.
Section	IfcSpace	Area or volume with a certain function, bounded actually or theoretically.
Unit	IfcLabel	Enumeration of items, sequential
Item	IfcBuiltElement	Components of the construction.
n/a	IfcRelAggregates ^r	Composition of spatial elements.
n/a	IfcRelContained- InSpatialStructure ^r	Assignment of elements to a certain level of the spatial project structure.
Flow	Flow	
Task	IfcTask	Unit of work to be carried out in a construction project.
Precedence ^r	Precedence ^r	Task 1 must be executed before Task 2 on instances at the same scope.
Alternate Precedence ^r	Alternate Precedence ^r	Task 1 precedes Task 2, and once Task 2 is started on an instance, Task 1 cannot progress on an instance at a different scope until Task 2 is finished.
Chain Precedence ^r	Chain Precedence ^r	Task 1 precedes Task 2, and their executions cannot be interrupted by other tasks on the instances at the same scope.
Execute ^r	Execution ^r	Definition of a specific ordering of executing tasks, fixing the path of execution.
Ordering Constraint ^a	Ordering Constraint ^{a*}	Order of work on spatial elements specified by a spatial constraint on task locations.
Exclusive Constraint ^a	Exclusive Constraint ^{a*}	Constraint on the parallel execution of tasks on spatial elements. Specified by referring to a level of the linked product hierarchy.
Scope ^a	Scope ^{a*}	The scope of a precedence dependency is defined by the spatial structures for which precedence between tasks must hold.

^r: relation, ^a: attribute, ^{*}: adapted definition and value range

Table 1: Overview of modeling language concepts and the mapping of CoPModL concepts to IFC classes.

In this work, we adopt the general two-part structure of CoPModL (**GR**₄), but align each part with the IFC schema. In particular, the metamodel depicts construction processes closely resembling the original *flows* in CoPModL, while aligning or replacing language constructs with IFC, when a conceptually similar concept exists in the schema (**SR**_{1,3}). The configuration part of CoPModL has largely been replaced by the IFC product hierarchy (**SR**₂).

In Figure 2, this two-part metamodel is represented in UML. The lower part of the metamodel describes processes, consisting of one class and four relations. It aims to model behavioral aspects of the construction process. While most attributes of *IfcTask* are intuitive, the attribute *Locations* can refer to any number of instantiable *IfcProduct* classes, implying the prescribed process is to be implemented in these items. In our proposal, the relation classes *Precedence*, *Chain Precedence*, and *Alternate Precedence* from *IfcTask A* to *IfcTask B* constrain that Task *B* follows Task *A*, Task *B* strictly follows Task *A*, and finally, Task *B*

follows Task *A* while both the tasks operate in different scopes. This is commonly represented, in a more general way, with the sequencing relation *IfcRelSequence*, linking *RelatingProcess* (predecessor) and *RelatedProcess* (successor). However, we are not mapping this class in the current implementation.

Additionally, the scope expresses the granularity of these precedence dependencies and is specified with classes from the spatial element hierarchy. Because there may be multiple paths of process execution, the modeler can explicitly plan or record a specific path using the relation *Execution*. Constraints on tasks allow expressing details on how a task is executed on structural elements and referenced locations.

Although based on CoPModL, the constraints have been aligned with the product hierarchy. Consequently, the granularity of constraints is not equivalent. An *Exclusive Constraint* expresses that spatial elements are exclusive to the task for its duration. The exclusive elements are specified by referring to a level of the spatial element hierarchy. The *Ordering Constraint* expresses a required order of products (spatial and built elements) according to spatial characteristics (vertical or horizontal ordering), for task execution.

The classes constituting an IFC product hierarchy are primarily divided into *IfcStructuralElements* and *IfcBuildingElements*; all available under the parent abstract class *IfcProduct*. All class names are intentionally canonical to enable interoperability across BIM projects. The data is modeled hierarchically across multiple classes like *IfcSite*, *IfcBuilding*, *IfcWall*, etc., to preserve the inherited features. The relationship classes *IfcRelAggregates* and *IfcRelContainedInSpatialStructure* further delineate this organization.

3.3 Implementation

Analogous to many existing visual modeling languages, we implement our variant using the ADOxx metamodeling platform [13]. The implementation of the entire language is segregated into two model-types, *Flow*, for modeling the construction process, and the *IFC Product Hierarchy* for modeling the spatial configuration. In ADOxx, model-types group instantiable elements and relations. Through a special attribute type, elements may establish references across models. These references are constrained by the target's class and model type. To interlink pairs of *Flow* and *IFC Product Hierarchy* models, we create a record class called *IfcProductRef* which points to all the instantiable elements of *IfcProduct* class.

While maintaining the hierarchical structure, the classes *IfcSite*, *IfcBuilding*, *IfcBuildingStorey*, and *IfcSpace*, as well as subclasses of *IfcBuiltElement*, are defined under their respective abstract classes in root class *D_Construct*. Although attribute *Name* is given to all the classes by default by ADOxx, the attribute *IfcGloballyUniqueId* is inherited by all the children of *IfcProduct*. In ADOxx, relations with graphical representation can be realized with specialized *relation classes*. Such relationship classes were implemented for the *IFC Product Hierarchy*-specific relations, namely *IfcRelAggregates* and *IfcRelContainedInSpatialStructure*.

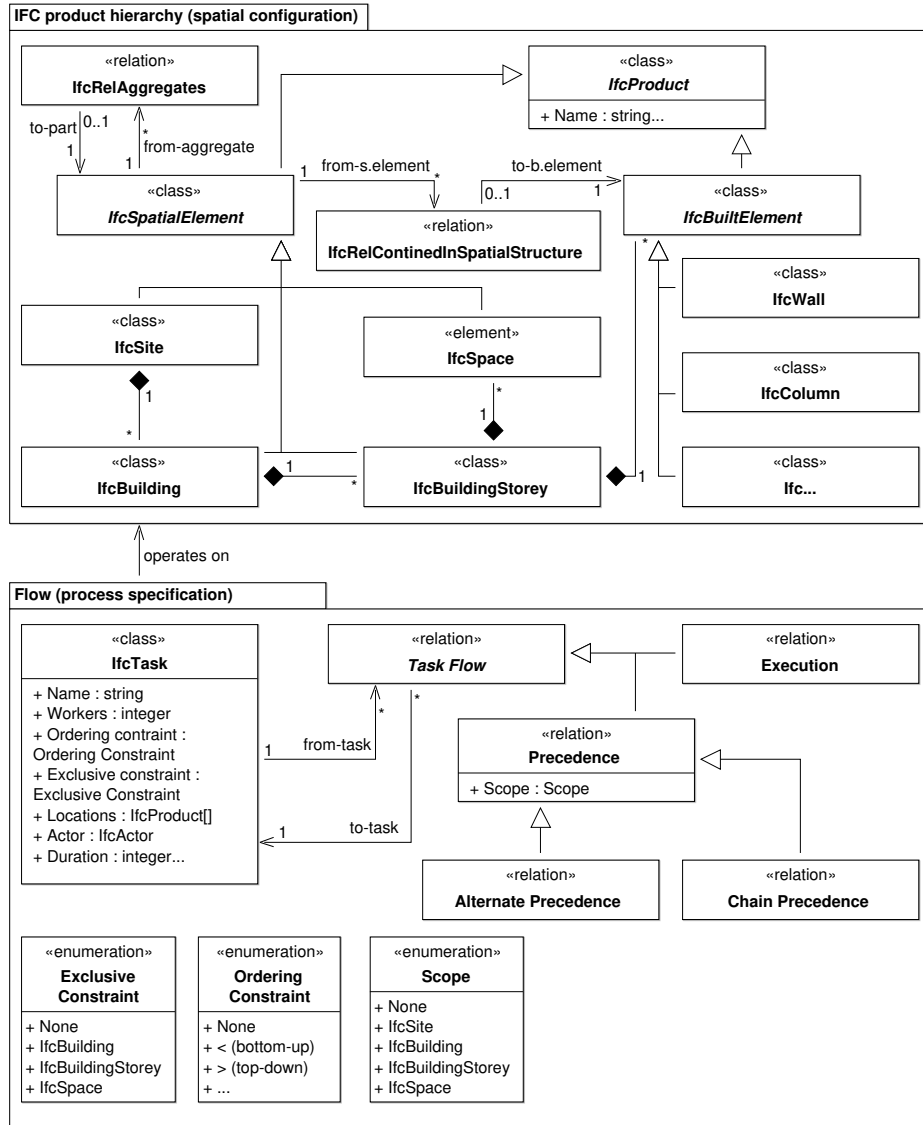


Fig. 2: Metamodel of the CoPModL adaptation integrating IFC. For brevity, some class attributes are omitted. The product hierarchy is an abstraction of the IFC product extension (included in the core layer of IFC), removing unneeded classes, attributes, and relations. The logical product hierarchy is denoted explicitly as compositions.

Consistent with the elements of configuration, the *IfcTask* class is also defined under the same *D_Construct* root class. Also, the relation classes *Precedence*, *Alternate Precedence*, *Chain Precedence*, and *Execution* are defined to connect *IfcTask* classes. The attributes *Scope*, *IfcActor*, *Ordering Constraint*, *Exclusive*

Constraint are implemented using an enumeration data type. All visual representations are produced using the GraphRep language [13], and most are dynamic to accommodate the variable number of locations (or buildings, building storeys, spaces, depending on the context) to be referenced.

4 Case Study

In this section, we apply our approach to a sample IFC test file provided by bSI⁴. These kinds of files are typically used by bSI for software certification purposes, as they include spatial decomposition and other critical aspects of a BIM project. The chosen IFC file provides a good example for testing our spatial constraints and scopes derived from CoPModL in combination with IFC semantics. Specifically, the project contains a single-storey residential house, partially modeled. Based on an inspection of the model, a suitable intervention scenario was identified, namely the construction of the missing outer north wall and a column attached to it. The new wall defines the physical border of the living room area.

Figure 3 presents the BIM project, with its geometric representation and associated metadata, visualized using the BIMVision platform⁵. The original metadata has been extended to accommodate our test. This includes renaming some objects and properties, including new ones, and defining a small number of interventions using the *IfcTask* class. The representation and organization of this information may vary across different software; therefore, environments with native IFC support are preferable for preserving the structure and semantics of the data. Additionally, general property sets have been added to specify typology and the material used in the new building element, as well as the construction status, highlighted in red.

The state of the BIM project obtained after the performed changes is considered the starting point for testing the usefulness of our research. In this regard, the designed metamodel is used to create both a configuration and a flow model of the single-family house project. Figure 4 shows the configuration part implemented in the ADOxx platform, which partially aligns with the hierarchical spatial elements of the IFC metadata schema. The objects represented are properly included as subtypes of the abstract class *IfcSpatialStructuralElement*.

Moreover, the model incorporates a subset *IfcBuiltElement* instances, including *IfcWall*, *IfcColumn*, and *IfcSlab*, with the latter restricted to the FLOOR and ROOF enumeration types. Regarding the relationship classes, *IfcRelContainedInSpatialStructure* is used to represent those building elements contained in the *IfcBuildingStorey*, which at the same time has two different *IfcSpace* aggregated using the *IfcRelAggregates* class. Those relationships are also specified in the legend in the upper-right corner. The adopted symbology is consistent with the overall mapping methodology, employing clearly defined arrows to illustrate the distribution of IFC entities across the IFC schema.

⁴ <https://github.com/buildingSMART/Sample-Test-Files/tree/main>

⁵ <https://bimvision.eu/>

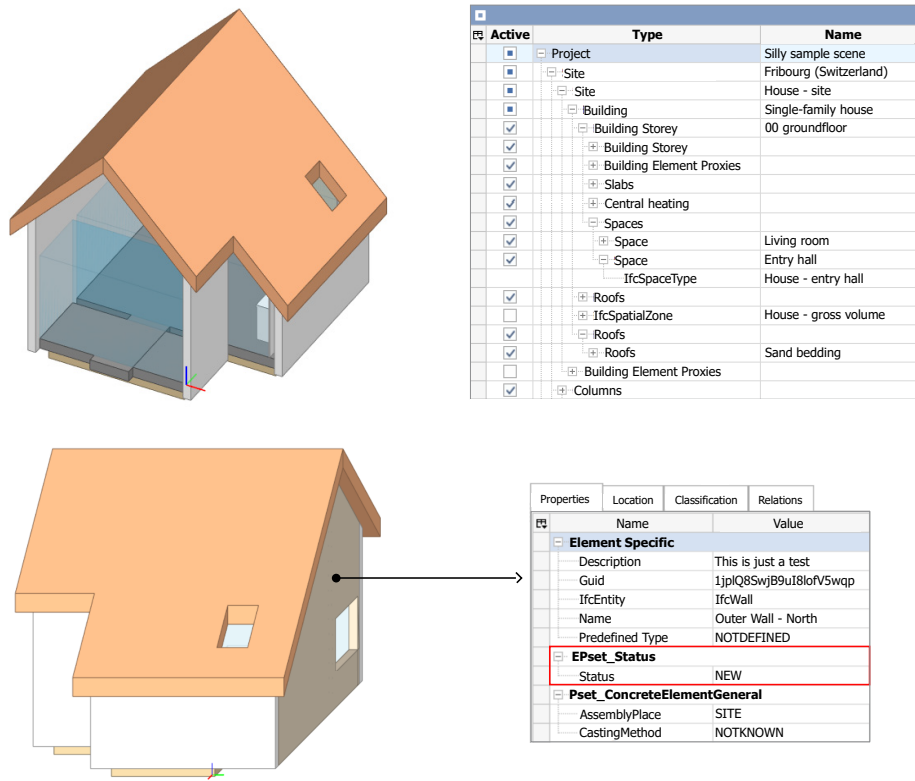


Fig. 3: The extended BIM project from bSI samples. The 3D model is shown on the left, while the enriched metadata is displayed on the right, with the project’s hierarchy at the top and properties at the bottom.

The limited use of relationship classes is motivated by the implementation-oriented nature of the approach, which primarily focuses on the flow (process specification) part. Building on this structural mapping, Figure 5 presents the flow part of the project, in which a 4D approach is described for the intervention in a specific part of the building. A list of planned interventions is modeled using the extended version of CoPModL proposed in this research. More specifically, the flow model represents an intervention scoped in the *IfcSpace* "Living Room". Inside it, the *IfcWall* "Outer Wall - North" is constructed. This kind of intervention requires the participation of different actors, and the execution order is also crucial.

The example, following real-scenario requirements, consists of the Scaffolding Installation (SI) with the "Scaffolder" actor role, followed by Pouring Concrete (PC) using the alternate precedence dependency. The process continues with different tasks, constraints, and scopes until the *IfcTask* Wall Finishing (WF) is completed. The item where each intervention will be performed is specified in the last row of each task, corresponding to the parameter *Locations* in the

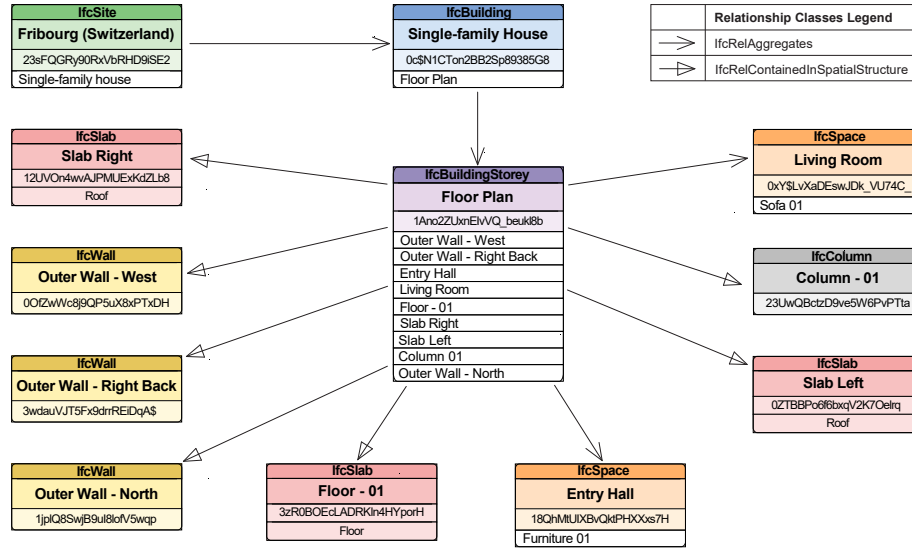


Fig. 4: Implementation in ADOxx of the configuration part with mapped IFC entities.

original CoPModL, and is assigned in IFC using the *IfcRelAssignsToProcess* class. In certain cases, relations are modeled without an explicit scope, which is represented in the figure with the value "None".

The rest of the dependencies are specified on the top right legend, and the used attributes on the bottom left, respectively. The implementation materials include the original IFC project provided by bSI, our customized version, and the ADOxx library developed during this research. Those materials, along with supplementary explanations, are available through a public GitHub repository⁶.

5 Discussion

This research presents the results of implementing a metamodeling approach using the BIM methodology. The standardization of data structure remains incomplete in the construction field, while new IFC versions are gradually resolving various technical issues. In this context, we aimed to enhance the interoperability of CoPModL by incorporating IFC data products during process modeling. Still, the *IfcProduct* class mapping process requires a deeper extension to ensure such interoperability. This indicates that, although the proposed extension improves semantic alignment, additional refinement of the product mapping mechanism is necessary before broader practical adoption can be expected.

IFC is constantly evolving due to the complexity of the AEC industry; however, the modeling of flow processes still requires special attention. In the context

⁶ <https://github.com/jmc-96/copmodl-ifc>

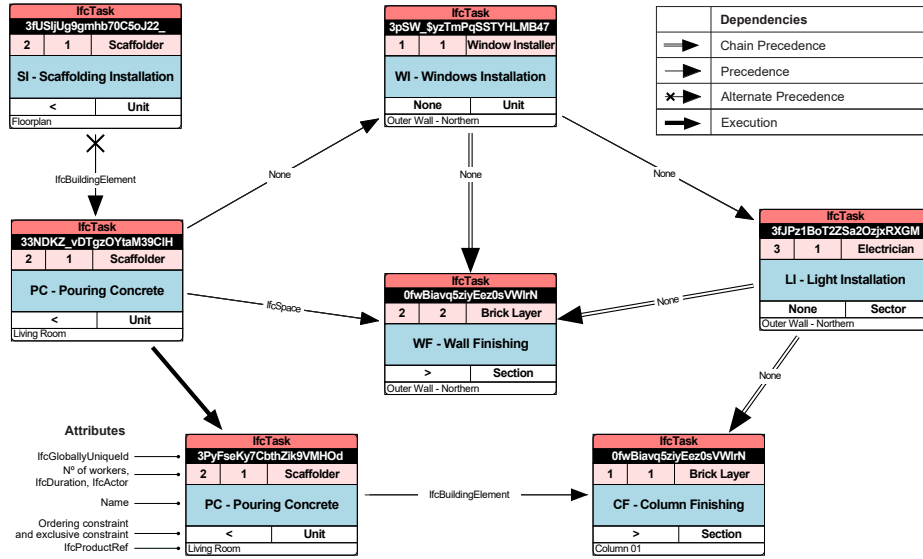


Fig. 5: Implementation of the flow part, with special focus on the *IfcTask* class.

of our research, the direct integration of ADOxx models with BIM environments remains an important topic for the future. This challenge could be addressed through an extension tool that connects the models to native IFC-based ecosystems. Implementing such a construction process language on a metamodeling platform, such as ADOxx, as opposed to developing a custom tool, provides numerous benefits. Specifically, rapid prototyping capabilities support short design and development iterations, while model processing algorithms can be implemented comparatively easily (SR₄). In the context of construction processes, such analytical capabilities are crucial for addressing complex temporal and spatial interdependencies, resource requirements, and stakeholder participation.

6 Conclusions and Outlook

In this paper, we presented a domain-specific modeling language for construction processes. The language is based on CoPModL, with significant changes in the form of an alignment with IFC. To achieve this, spatial characteristics of a construction project are modeled according to the IFC product hierarchy, forming spatial hierarchies. Processes are modeled according to CoPModL, whose behavioral concepts have been adjusted to operate on IFC spatial hierarchies. The applicability of the languages has been demonstrated through a case study based on a readily available IFC file used for software testing and certification by bSI.

A promising direction for future research is to integrate our extended IFC-based CoPModL models into an advanced metamodeling environment, such as the web-based MM-AR platform [24]. In the current formulation, CoPModL

abstracts from the physical geometry and topological relations of the building elements, assuming that the location structure is predefined. A systematic integration with IFC would enable the automatic derivation of the configuration layer (item structures, attributes, perspectives) from the spatial decomposition and classification mechanisms already present in BIM models (e.g., spatial containment). In this context, it will be worthwhile to also explore the use of AI approaches for creating, analyzing, and transforming models [12,11,10].

Beyond vocabulary alignment, the incorporation of geometric and topological information would open up the possibility of extending process constraints with spatially informed semantics. For example, precedence scopes could be dynamically derived from IFC spatial hierarchies, and exclusivity constraints could consider geometric proximity rather than purely symbolic item identifiers. Implementing this integration in a modeling ecosystem would allow the definition of a unified metamodel combining structural (IFC-based) and behavioral (CoPModL-based) layers, enabling formal validation of construction process coordination directly against real BIM projects. Such an approach could support automated consistency checks between the planned execution logic and the actual building geometry, thereby bridging the gap between spatial modeling and declarative process reasoning.

Acknowledgments. This work was supported by the *Smart Living Lab* (<https://www.smartlivinglab.ch/en/>), funded by the University of Fribourg, EPFL, and HEIA-FR.

6.1 Disclosure of Interests.

The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Amor, R.: Analysis of the evolving IFC schema. In: 32nd CIB W78 Information Technology for Construction Conference (CIB W78 2015), Eindhoven, Netherlands (2015), <https://api.semanticscholar.org/CorpusID:63975271>
2. Beetz, J., van Leeuwen, J., de Vries, B.: IfcOWL: A case of transforming express schemas into ontologies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* **23**(1), 89–101 (2009). <https://doi.org/10.1017/S0890060409000122>
3. Berlo, L., Krijnen, T., Tauscher, H., Liebich, T., Kranenburg, A., Paasiala, P.: Future of the industry foundation classes: towards IFC 5. In: Proceedings of the 38th International Conference of CIB W78. pp. 123–137. CIB W78 conference series (2021), 38th International Conference of CIB W78 ; Conference date: 11-10-2021 Through 15-10-2021
4. Börger, E.: Approaches to modeling business processes: a critical analysis of BPMN, workflow patterns and YAWL. *Software and Systems Modeling* **11**(3), 305–318 (2012). <https://doi.org/10.1007/s10270-011-0214-z>

5. Borgo, S., Sanfilippo, Emilio M. and Šojić, A., Terkaj, W.: Ontological Analysis and Engineering Standards: An Initial Study of IFC, pp. 17–43. Springer International Publishing, Cham (2015). https://doi.org/10.1007/978-3-319-15326-1_2
6. Camposano, J.C., Smolander, K.: Inter-organizational integration in the AEC/FM industry. In: Giorgini, P., Weber, B. (eds.) *Advanced Information Systems Engineering*. pp. 64–79. Springer International Publishing, Cham (2019). https://doi.org/10.1007/978-3-030-21290-2_5
7. De Giacomo, G., Vardi, M.Y.: Linear temporal logic and linear dynamic logic on finite traces. In: *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, Beijing, China*. p. 854–860. IJCAI '13, AAAI Press (2013)
8. Döllner, J., Buchholz, H.: Continuous level-of-detail modeling of buildings in 3D city models. In: *Proceedings of the 13th Annual ACM International Workshop on Geographic Information Systems*. p. 173–181. GIS '05, Association for Computing Machinery, New York, NY, USA (2005). <https://doi.org/10.1145/1097064.1097089>
9. Domer, B., Bernardello, R.: *Interoperability: an introduction to IFC and buildingSMART standards integrating infrastructure modeling*. EPFL Press (2023), <https://www.epflpress.org/produit/1455/9782889154869/interoperability>
10. Fill, H.G., Fettke, P., Köpke, J.: Conceptual Modeling and Large Language Models: Impressions From First Experiments With ChatGPT. *Enterp. Model. Inf. Syst. Archit. Int. J. Concept. Model.* **18**, 3 (2023). <https://doi.org/10.18417/EMISA.18.3>
11. Fill, H.G., Horkoff, J., Fettke, P., Köpke, J.: Generative AI and Conceptual Modeling. *Business & Information Systems Engineering* **68**(1), 1–5 (Feb 2026). <https://doi.org/10.1007/s12599-025-00978-8>
12. Fill, H.G., Härer, F., Vasic, I., Borcard, D., Reitemeyer, B., Muff, F., Curty, S., Bühlmann, M.: CMAG: A Framework for Conceptual Model Augmented Generative Artificial Intelligence. In: Gallinucci, E., Yasar, H., Liaskos, S., Marcel, P., Chen, P.P., Cesare, S.d., Gailly, F. (eds.) *Companion Proceedings of the 43rd International Conference on Conceptual Modeling: ER Forum, Special Topics, Posters and Demos Co-located with ER 2024, Pittsburgh, Pennsylvania, USA, October 28-31, 2024*. CEUR Workshop Proceedings, vol. 3849, pp. 56–69. CEUR-WS.org (2024), <https://ceur-ws.org/Vol-3849/forum5.pdf>
13. Fill, H.G., Karagiannis, D.: On the conceptualisation of modelling methods using the ADOxx meta modelling platform. *Enterp. Model. Inf. Syst. Archit.* **8**, 4–25 (2013). <https://doi.org/10.18417/emisa.8.1.1>
14. Frank, U.: Domain-specific modeling languages: Requirements analysis and design guidelines. In: *Domain Engineering: Product Lines, Conceptual Models, and Languages*, pp. 133–157. *Domain Engineering: Product Lines, Conceptual Models, and Languages* (2013). https://doi.org/10.1007/978-3-642-36654-3_6
15. Gnerlich, R., Tiesler, A., Franz, V.: Simulation of construction disturbances with agent-based petri nets and employing a BIM in expert building time analyses. *Simul. Notes Eur.* **30**(4), 203–212 (2020). <https://doi.org/10.11128/SNE.30.TN.10542>
16. Häußler, M., Borrmann, A.: Knowledge-based engineering in the context of railway design by integrating BIM, BPMN, DMN and the methodology for knowledge-based engineering applications (MOKA). *J. Inf. Technol. Constr.* **26**, 193–226 (2021). <https://doi.org/10.36680/j.itcon.2021.012>
17. International Organization for Standardization: *Industrial automation systems and integration – product data representation and exchange – part 11: Description*

- methods: The EXPRESS language reference manual. ISO 10303-11:2004 (2004), available at: <https://www.iso.org/standard/38047.html>
18. Karagiannis, D., Kühn, H.: Metamodelling platforms. In: Bauknecht, K., Tjoa, A.M., Quirchmayr, G. (eds.) *E-Commerce and Web Technologies*, vol. 2455, pp. 182–182. Springer Berlin Heidelberg. https://doi.org/10.1007/3-540-45705-4_19, http://link.springer.com/10.1007/3-540-45705-4_19, series Title: Lecture Notes in Computer Science
 19. Kelley, J.E., Walker, M.R.: Critical-path planning and scheduling. In: *Papers Presented at the December 1-3, 1959, Eastern Joint IRE-AIEE-ACM Computer Conference*. p. 160–173. IRE-AIEE-ACM '59 (Eastern), Association for Computing Machinery, New York, NY, USA (1959). <https://doi.org/10.1145/1460299.1460318>
 20. Laakso, M., Kiviniemi, A.: The IFC standard - A review of history, development, and standardization. *J. Inf. Technol. Constr.* **17**, 134–161 (2012), <https://www.itcon.org/paper/2012/9>
 21. Liebich, T.: Highlights of the development process of industry foundation classes. In: Lacasse, M.A., Vanier, D.J. (eds.) *Information technology in construction*. vol. 18. Vancouver, Canada (1999), <http://itc.scix.net/paper/w78-1999-2758>
 22. Marengo, E., Nutt, W., Perktold, M.: CoPMod: Support for construction process modeling. In: *Proceedings of the Dissertation Award, Demonstration, and Industrial Track at BPM 2018*. CEUR Workshop Proceedings, vol. 2196, pp. 61–65. CEUR-WS (2018)
 23. Marengo, E., Nutt, W., Perktold, M.: CoPModL: Construction process modeling language and satisfiability checking. *Information Systems* **103**, 101457 (2022). <https://doi.org/10.1016/j.is.2019.101457>
 24. Muff, F., Borcard, D., Fill, H.G.: MM-AR: a web-based open-source metamodeling platform for spatial conceptual modeling. *Software and Systems Modeling* (Feb 2026). <https://doi.org/10.1007/s10270-025-01351-9>
 25. Murata, T.: Petri nets: Properties, analysis and applications. *Proceedings of the IEEE* **77**(4), 541–580 (1989). <https://doi.org/10.1109/5.24143>
 26. Muñoz-Cádiz, J., Mariotti, C., Nespeca, R., Bolognese, L.: A methodology for integrating the CIDOC-CRMba ontology into the ifc schema to support spatial analysis in archaeological heritage. *Digital Applications in Archaeology and Cultural Heritage* **37**, e00431 (2025). <https://doi.org/10.1016/j.daach.2025.e00431>
 27. Object Management Group: Business Process Model and Notation (BPMN), Version 2.0. OMG Specification formal/2011-01-03, Object Management Group (OMG) (Jan 2011), <https://www.omg.org/spec/BPMN/2.0/PDF>. Accessed: 2026-02-27
 28. Object Management Group: EXPRESS – express metamodel, version 1.1. Formal Specification formal/15-05-01, Object Management Group (May 2015), <https://www.omg.org/spec/EXPRESS/1.1/PDF>, accessed 2026-03-06
 29. Pauwels, P., Terkaj, W.: EXPRESS to OWI for construction industry: Towards a recommendable and usable ifcOWL ontology. *Automation in Construction* **63**, 100–133 (2016). <https://doi.org/10.1016/j.autcon.2015.12.003>
 30. Pestic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: *11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007)*, 15-19 October 2007, Annapolis, Maryland, USA. pp. 287–300. IEEE Computer Society (2007). <https://doi.org/10.1109/EDOC.2007.14>

31. Sacks, R., Lee, G., Burdi, L., Bolpagni, M.: BIM Handbook: A Guide to Building Information Modeling for Owners, Designers, Engineers, Contractors, and Facility Managers. John Wiley & Sons, fourth edn. (2025). <https://doi.org/10.1002/9781394222254>
32. Schenck, D.A., Wilson, P.R.: Information modeling: the EXPRESS way. Oxford University Press, Inc., USA (1994)
33. Sibenik, G., Kovačić, I.: Assessment of model-based data exchange between architectural design and structural analysis. *Journal of Building Engineering* **32** (2020). <https://doi.org/doi:10.1016/j.jobbe.2020.101589>
34. Zech, P., Kelter, C., Hammes, S., Michael, J., Breu, R.: Meta-metamodel-independent model transformations. In: Taibi, D., Smite, D. (eds.) *Software Engineering and Advanced Applications (SEAA 2025)*. Lecture Notes in Computer Science. vol. 16081, pp. 329–348. Springer Nature Switzerland, Cham (2026). https://doi.org/10.1007/978-3-032-04190-6_20
35. Zhao, Y., J.H.Y.M.L.J.: Efficient project management of historical building groups: a BIM-based approach with critical chain method and 5D simulation. *International Journal of Construction Management* **25**(7), 826–839 (2025). <https://doi.org/10.1080/15623599.2024.2366711>