Enhancing design coordination across disciplines through incremental model updates and Inter-discipline Conjunction Graphs

Sebastian Esser[1] and André Borrmann[1]

[1] Technical University of Munich, School of Engineering and Design, Chair of Computational Modeling and Simulation, Munich, Germany
sebastian.esser@tum.de

Abstract

Model-based collaboration is well-established to exchange design information in the Architecture, Engineering, and Construction (AEC) sector. Even though various tools provide comprehensive means to author such models containing discipline-specific elements and semantic information, there is great potential to improve how multiple discipline models are coordinated in an integrated environment. Today, project coordination is carried out through model federation by uploading (discipline) models to Common Data Environments (CDE), which implement the concept of BIM Level 2 according to ISO 19650. This standard, however, handles discipline BIM models primarily as monolithic files and does not foresee any connection among the individual objects the models contain across different disciplines. Subsequently, detecting and negotiating potential impacts on foreign models caused by modifications in a specific discipline model remains a manual task that must be carried out every time a new model version is populated.
Incremental version control of BIM models has recently gained increasing research interest. Such approaches aim to exchange only the modified parts of a model. Due to the missing references across discipline models, however, the potential of such methods leaves further potential unexploited. However, enhanced collaboration can be achieved if the actual model increment is exchanged, and its content is used to propose modifications to the other discipline models. This paper presents an approach to establishing additional references across distinct discipline models in the coordination environment to realize this concept. The proposed method helps assess the impact of model modifications across disciplines and allows for implementing automated reactions. The basis of the proposed concept is formed by an *Inter-discipline Conjunction Graph*, which provides graph-based representations of the different discipline models and several types of conjunction relationships that connect information across the different discipline models. By establishing these interdisciplinary references, a modification potentially affecting other disciplines can be assessed explicitly. At the same time, the accepted best-practice of working with distinct federated discipline models is still respected but significantly improved by a more comprehensive management of the consequences caused by model changes.

Keywords: Building Information Modeling, Version Control, Discipline Models, Design Changes, Modification Assessment.

1 Introduction

The AEC (Architecture, Engineering, and Construction) sector is characterized by a large number of highly specialized disciplines. Particularly in projects with diverse and often unique boundary conditions, numerous specialists must pool their expertise to devise a design solution that meets a construction project's diverse demands and dependencies. Consequently, the process of building design becomes a highly collaborative endeavor. Over the past years, the industry has increasingly adopted Building Information Modeling (BIM), providing model-based methods to collect, analyze, and exchange design and construction information. A BIM model typically contains geometric and semantic information about a built asset and is often encoded in vendor-neutral, open data models. The Industry Foundation Classes (IFC) data model is the most prominent example of available data models. These data models specify how information about a building or infrastructure asset is structured to enable its exchange across different applications. Furthermore, these data models provide a solid basis for exchanging information across various stakeholders and disciplines in a project.

Numerous standards and guidelines, both at the national and international levels, have been established to organize the workflows of collaborative design projects within the construction industry. Among these, the most significant is ISO 19650. The standard mainly describes aspects related to the exchange and management processes and advocates for BIM-based collaboration through federated discipline models. Following ISO 19650, each domain operates independently of other disciplines, focusing on their respective discipline models. Once a shareable state is reached, the design information is uploaded as a monolithic file or as a set of files in an Information Container to a Common Data Environment (CDE) [1]. Interdisciplinary coordination is then achieved by super-imposing all discipline models into a coordination model, finding potential collisions, and fixing them in the respective discipline models.

1.1 Problem Statement and Research Questions

While model-based collaboration already represents a significant achievement on the path to interconnected processes in construction, current systems still insufficiently account for the particularities of the iterative planning processes [2]. Further considerations are therefore necessary to enhance how model changes are handled and, above all, to understand the implications that changes in one discipline can have on other discipline models. Given this vision, connecting distinct representations during the ongoing design process becomes vital to accelerate the evaluation of design changes and their impacts on foreign discipline models. To enable quicker change assessment, we elaborate on possibilities to establish conjunctions across objects allocated in different discipline models. Once such relationships have been introduced, further analysis of changes can be assessed much quicker than in current BIM collaboration platforms.

In summary, the research presented in this paper aims to elaborate on the following two research questions:
Research Question 1:
How can heterogeneous design information be linked to represent interdisciplinary dependencies forming an *Inter-discipline Conjunction Graph* (ICG)?
Research Question 2:
How should we deal with updates that affect parts of a coordination graph?

1.2 Outline

Section 2 summarizes related works and fundamental concepts of model-based collaboration. Section 3 presents our methodological approach that (i) establishes various object links and (ii) discusses how incoming model changes can be populated to those disciplines affected by this update. Section 4 introduces a case study demonstrating the power of aligned discipline models. The paper closes with section 5, which summarizes the approach, critically assesses the achieved benefits, and provides an outlook on future research directions.

2 Related Works

2.1 Model-based Collaboration and Information Containers

Today, collaboration in the construction industry is mainly characterized by the authoring and exchange of BIM models. ISO 19650-1 describes various aspects concerning roles, states, and processes related to disseminating models to other project partners using Common Data Environments (CDEs) [3]. Although the continuous usage of such project platforms already enhances the transparency and traceability of discipline-specific datasets, two significant downsides can be identified. On the one hand, current CDE platforms must properly admit the need to handle heterogeneous data and exchange associated link sets. On the other hand, models (or, more generally, any deliverable to be exchanged) are still managed as monolithic files without further processing on the object level, which causes severe limitations if changes to a model must be made and exchanged accordingly.

Various standards that support the bundled exchange of heterogeneous information occurring in construction projects have been defined in the past years. For this purpose, Information Containers for linked Document Delivery (ICDD) have been introduced in DIN SPEC 91391-1 and DIN EN ISO 21597 [4], [5], [6]. Further specifications and refinements have recently been made in the successor norm DIN 18290 titled “Linked BIM data exchange of building information models with further specialist models”. Senthilvel, Oraskari, and Beetz have investigated how ICDDs align with Common Data Environments [7]. They report that the notion of ICDDs generally aligns with the principles of so-called Linked-Data platforms and the buildingSMART OpenCDE API specifications, which aims to harmonize the accessibility of CDE platforms in a vendor-neutral manner. However, some terms and meanings are kept very flexible, making the mapping process of containers to existing interfaces of these systems complicated. To date, barely any software application supports such containers.

2.2 Graph-based Representations and Incremental Version Control Methods for BIM Models

Besides the need for more support for container formats in existing tools, further considerations are required to meet the highly iterative nature of the construction domain. Given the circumstances that many experts of different disciplines must contribute to the design of a building, the design of buildings and infrastructure facilities remains a highly iterative task since it is a multi-objective challenge. Several researchers have recently investigated new approaches to exchanging design information in an incremental fashion. Their approaches aim to overcome limitations caused by file-based transfers and enable direct access to modified parts of a model by only sharing an update increment instead of the entire (modified) model repeatedly.

Many recent research activities have identified graph-based representations as a flexible yet leveraging means to represent data from various sources, including highly diverse discipline models [8]. In this regard, two major fashions of representing data in graphs exist: On the one hand, graphs following the Resource Description Framework (RDF) paradigms represent knowledge in triples consisting of a subject, a predicate, and an object, whereas on the other hand, Labeled Property Graphs (LPGs) allow the attachment of attribute sets directly to nodes and edges. Even though RDF and associated techniques of the Semantic Web are well documented and widely used in many disciplines, these representations have deficiencies in their application for construction data. Direct mappings of BIM models into RDF graphs often require specific workarounds and result in large graphs with many different types, reducing the benefits of such graphs [9]. In classical Semantic Web approaches, the used taxonomies are given and specified in ontologies, which typically provide only a limited vocabulary.

To overcome these limitations, various studies used Labeled Property Graphs (LPG) to represent all entities given in an IFC model following object-oriented principles [10], [11] and solve dedicated use cases like finding evacuation routes [12]. Other researchers have used such graph representations to identify differences between two IFC models [13]. Methodological approaches that enable proper version control tracking for BIM models on their pure data representation level are particularly interesting for solving the research questions stated in Section 1.1. Such approaches have been documented in [14], whereas [15] extends the basic diff-and-patch concept using branching and merging. This way, diverging models can be re-combined, and captured changes can be fused to create a consolidated model. The approach is closely aligned with established notions of graph theory and transformations and can be applied to various data models currently used in the AEC industry.

2.3 Strategies for Detecting Associated Objects Across Multiple Models

Interlinking heterogeneous datasets has been studied intensively in the field of the Semantic Web, where many different ontologies describing various relationship types exist [16]. In the specific design and engineering data domain, Teclaw et al. presented a method to find duplicated information in various discipline models later coordinated in a Common Data Environment [17]. They used selected geometrical features to identify similar elements in different discipline models and establish links across them. Their study aimed to identify similar spaces and levels in various discipline models from the architectural and HVAC disciplines. Even though their approach has shown appropriate results for their case study, it appears complex to generalize the criteria chosen to identify the same objects. A comparable approach has been discussed by Sibenik et al. [18], who transform architectural models into structural models based on geometric features. Wang et al. [19] reflect each discipline model as a subgraph in an extensive graph database and create additional constraint dependencies between objects of different models. The instantiation of these relationships is again based on certain design intent assumptions. Additionally, to the instantiation of such object constraints, they report on mechanisms for how to react to design changes applied to models that have already been integrated into the overall coordination model (or graph, respectively). In case of detected violations of existing constraints, a violation processing module is triggered to develop a resolution of the issue, which is ultimately sent to the model author for approval. Although this approach is promising and forward-thinking, the presented case study utilizes many onboard functions and constraint tooling of a specific vendor, which makes their approach difficult to generalize and apply to vendor-open model coordination.

To address this gap, initiatives related to query languages can help to further generalize the vision of interconnecting distinct discipline models. Query languages that support extracting elements based on spatial, logical, and topological operators are particularly interesting. These methods are grounded on processing the geometric representations stored in a BIM model. Comprehensive methodological approaches have been documented in [20] and [21]. Such spatial descriptors can help compute relations not explicitly modeled in the BIM model. Even though previous investigations were merely focused on processing single models, the methodologies are adaptable to scenarios with multiple (partial) models if any model in question can be represented using Boundary Representation (BRep) geometry.

In addition to these rather domain-specific approaches, various attempts have been undertaken to align knowledge graphs using Artificial Intelligence (AI) techniques [22].

2.4 Identified Research Gap

Previous research activities prove a high interest in transferring heterogeneous information with associated link sets. Even though few approaches have been documented to how such links can be established across different discipline models, no proper methodology utilizes such relationships to evaluate the impact of design changes on interlinked, foreign discipline models using proper vendor-neutral data standards. Hence, the next chapter will introduce a methodology that (i) uses a schema-agnostic representation for BIM models as graphs, which can be used to abstract and integrate incremental patches on object level, and (ii) establishes alignment links across these different graphs. By joining the power of object-based version control and the available interdisciplinary relationships, it is possible to evaluate the impact of model changes more comprehensively before they are integrated into the coordination models.

3 Methodological Approach

The core of the proposed methodology lies in establishing different action levels that must be undertaken if model changes occur on models that have already been integrated into coordination models. Therefore, we define different complexity levels of actions in ascending complexity that may be triggered if a partial model should be updated inside a coordination model:

- **Notify**: Inform all stakeholders about new model changes in a specific discipline model.
- **Report**: Let users know if model changes create potential impacts on their own models and precisely report on the affected objects in their corresponding models.
- **Auto-resolve**: Automatically react to incoming changes issued in foreign models and automatically modify associated objects in the affected discipline models.

Today, many CDE platforms are already capable of setting alerts for different types of events that occur on these platforms. Such events can comprise the publishing of new datasets or any issue reporting. Hence, the simplest case of *notifying* users can be considered solved and doesn't need further attention. On the contrary, complete automation, which is described as auto-resolve, is also not intended as it would significantly interfere with the authorship of responsible engineers. Instead, we aim for an approach where users can set up customized workflows to process further incoming updates issued for foreign models. Subsequently, we can limit the problem discussed in this article to establish interdisciplinary object relationships to precisely *report* to other domains about incoming modifications and affected components.

Figure 1 illustrates the overall approach and the steps to be taken. As a premise, all domains provide their design information in a graph representation. In most cases, domains are expected to deliver information as BIM models following object-oriented paradigms. These models are then curated on a central storage and management platform). Additional interdisciplinary references between objects of the different discipline models are instantiated and incorporated into the central coordination platform. Once this interconnected stage is achieved, we denote the interconnected stage as a *Conjunction Graph,* including all discipline models and the conjunction relationships. Now, incoming modifications can be evaluated in terms of their impact on the overall (i.e., coordinated) model. The effect of a transformation rule p applied to the graph G_i reflecting one specific discipline model is to be evaluated in the scope of the entire conjunction model (or graph, respectively).

Section 3.1 summarizes the authors' previous work on incremental version control applicable to any BIM model following object-oriented principles. Section 3.2 explains the approach that is used to analyze the interdisciplinary relationships using the ideas of query languages for BIM models. Section 3.3 describes how the chosen notion of update patches can be directly used to notify associated objects in foreign models.

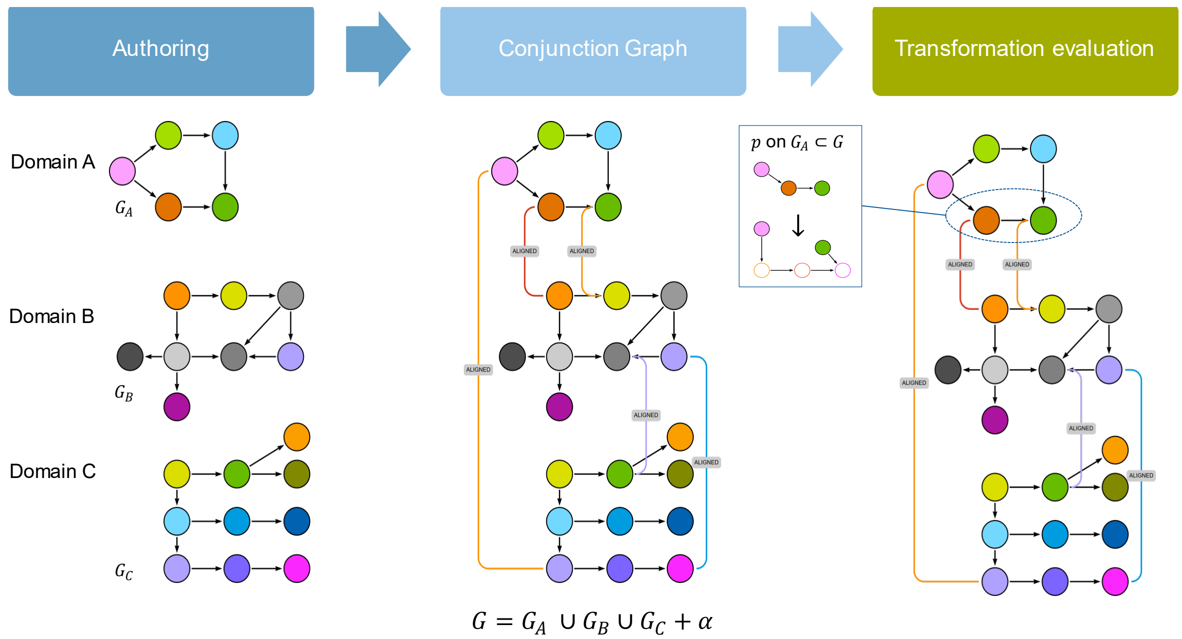

Figure 1: Overview of the proposed methodology to connect discipline models and validate model changes' impacts in an integrated fashion.

3.1 Chosen Model Representation and Version Control Approach

As discussed earlier, Labeled Property Graphs (LPGs) are closely aligned with the core principles of object-oriented data modeling, which build the base for most data models today available for the AEC sector. In essence, each class instance is reflected by a single node. All attributes specific to an instance are attached to the respective node as property sets. Furthermore, directed edges represent associations between instances in the graph. The graph reflecting one discipline model is called the *Discipline Graph*.

In [14], a comprehensive approach was presented that determines the changes from a given BIM model version based on LPGs, formalizes them in an update patch and applies this patch to outdated model versions.

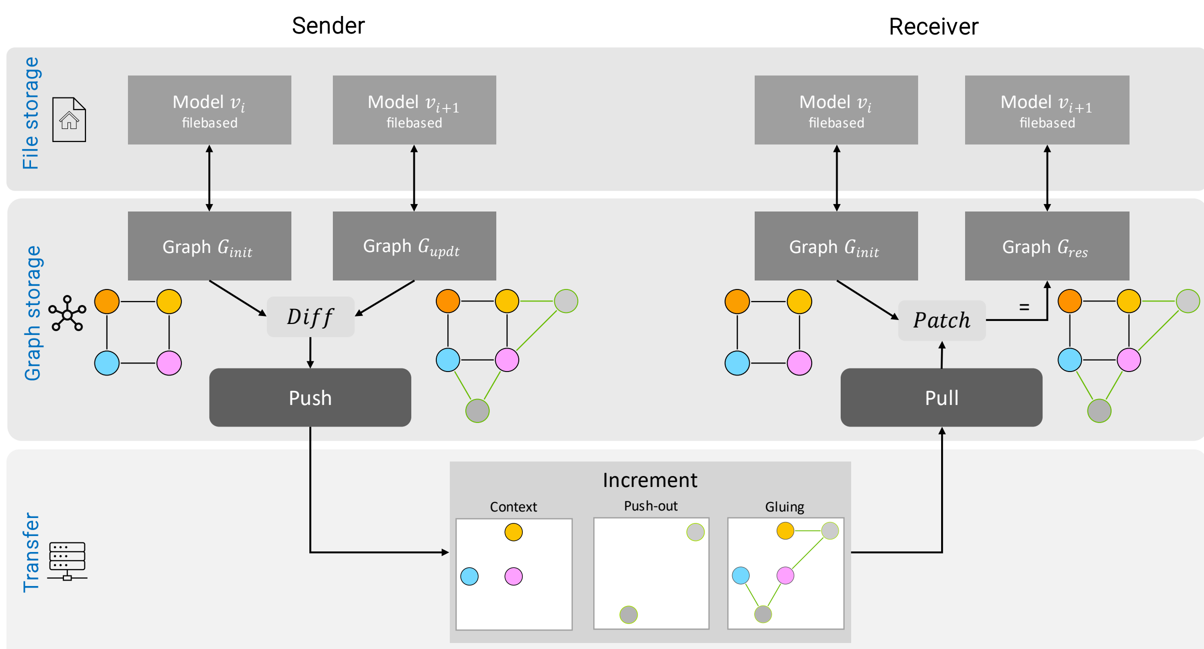

Figure 2: Overall Diff-and-Patch architecture presented in [14].

Three graph patterns describe the increments. The context pattern specifies a subgraph that must be matched in the host graph to properly (un-)connect the removed or inserted graphlet to the unaltered parts of the model (i.e., graph, respectively). The push-out pattern reflects inserted and removed nodes and edges representing added and deleted objects in the BIM model. The gluing pattern ultimately describes the edges, which connect the push-out pattern to the host graph. If only properties attached to a node have been edited, these modifications do not affect the graph's topological structure. Hence, this information is transferred as a semantic modification and consists of a unique path specifying the modified node unambiguously and the attribute data to be changed. Figure 3 illustrates the data model for increments as a UML diagram. A graph database capable of hosting LPGs is employed to store and interact with these graphs. This study used the database system neo4j because it implements the Cypher query language, which may eventually become a new standard for interacting with property graphs [23].

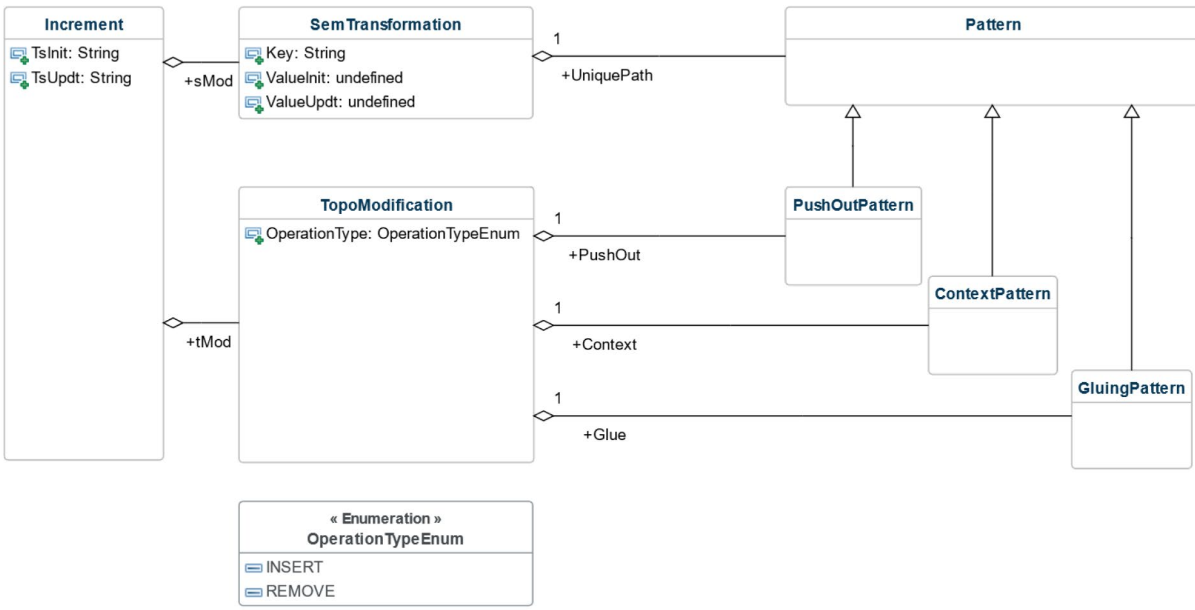

Figure 3: UML model of the increments capturing the modifications between two model versions

3.2 Establishing the Object Conjunction Graph

This processing step aims to identify connections between objects that can be established across distinct discipline models. Despite each domain overseeing a set of objects that reflects their part of the design, all diverse elements and systems must be integrated to realize the actual asset in the built environment. Frequently, disciplines rely on the preliminary works of other domains for their planning and simulation tasks. For instance, structural models are often derived from architectural models. However, the reference between elements in both models is typically not captured or exchanged later. Therefore, a specific focus is put on determining such relationships based on the ultimate deliverables shared by each domain on the collaboration platform.

The instantiation of conjunction relationships across objects of distinct discipline models is challenging and cannot be adequately generalized. Nevertheless, applying different spatial and logical operators can help to determine such relationships. In addition to the theoretical concepts presented in [20] and [21], we have studied their application in well-established software applications supporting model-checking and customized information extraction from BIM models. We have tested the capacities of SimpleBIM[1], Desite BIM[2], and Solibri[3] in detail and have decided to proceed with Solibri's Information Take Off (ITO) module for further testing. The tooling of this module includes:

- Several filtering options by object types, values, or spatial dependencies across all loaded discipline models.
- The extraction of various object quantities (either present as property sets or calculated based on the processed geometry).

[1] https://simplebim.com/ (last access 2023-11-30)
[2] https://thinkproject.com/products/desite-bim/ (last access 2023-11-30)
[3] https://www.solibri.com/ (last access 2023-11-30)

- The acquisition of related elements based on their location or geometric connectivity (e.g., location of objects, overlapping shapes, touching surfaces, containing bounding boxes, etc.).

The latter function has unveiled great potential to generate additional relationships across objects of one model and conjunctions amongst different disciplines.

3.3 Transformation Evaluation

As described, the version increments comprise topological and semantic modifications. Implementing semantic modifications is straightforward as these model changes only affect properties attached to a node without topological changes in the graph. Hence, the *UniquePath* pattern uniquely specifying the edited node can be further utilized to inform any foreign object aligned with the modified component.

In contrast, topological modifications have an advanced structure. These changes are expressed by a push-out pattern describing the inserted or removed graphlet, the Context pattern specifying nodes the push-out part is connected to, and the gluing pattern containing the actual edges that bind the Push-Out to the Context. The context and push-out patterns must be considered further to evaluate the impact of such an increment on other discipline models.

In case of a removal operation, the version control system starts with matching all three patterns. Next, the Gluing edges are deleted, and the push-out Pattern is removed from the graph. As object conjunctions can be established across any node, additional pre-processing is required. If aligning edges exist to nodes that are part of the push-out pattern, the operation must be stopped immediately to prevent the existence of dangling edges. Less critical are relationships to nodes that reflect dependencies with nodes contained in the context pattern. In this case, either the transformation should be declined, or the authors of the associated objects in all foreign discipline graphs should be informed. This strictness cannot be advised in a general manner but should be set with respect to the project's peculiarities.

Detecting affected components is more complicated if a topological modification inserts new graphlets into a discipline graph. Of course, the context pattern can still be used to identify elements associated with other models. However, the newly inserted push-out pattern does not contain additional object alignment information in the Gluing pattern, as discipline models are expected to be edited independently.

4 Case study

The following section introduces a case study with two disciplines to further discuss the benefits of the proposed object conjunction. During the design, the space model depicted in Figure 5 was initially derived from the architectural model shown in Figure 4. However, as the space model is intended to be used in Facility Management software, two distinct models have been exported and handled separately.

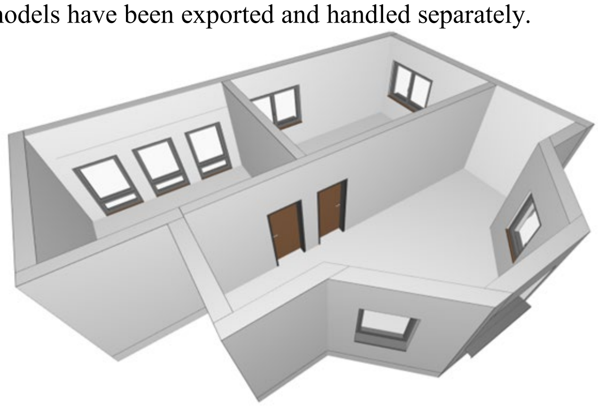

Figure 4: An architectural model containing walls, windows, and doors

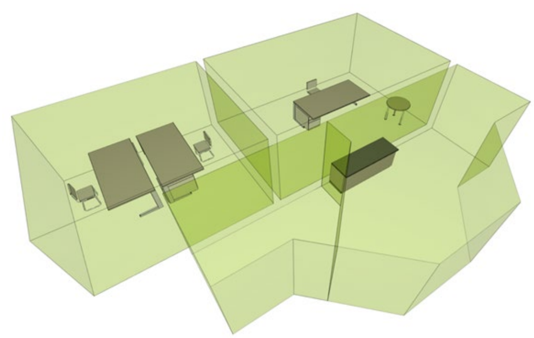

Figure 5: A space model containing space geometries and furniture items

4.1 Object Conjunction

Both models were translated into graph representations using the approach presented in [14]. This processing step resulted in two graphs reflecting the respective discipline models in our graph database. To establish sufficient object references, we performed geometric adjacency algorithms available in the software application Solibri Office [24]. In detail, we performed queries to unveil the following relationships:

- What is the containing space for each furniture item?
- What architectural components are adjacent to each room?

Besides these queries, both models were aligned in their spatial breakdown structures. In essence, all spatial containers (e.g., the IfcSite and IfcBuilding instances and the different storeys) were matched based on their entity type or their base elevation, respectively. The results of all aligning queries were then added to our graph storage, which already held the graph representations of both models.

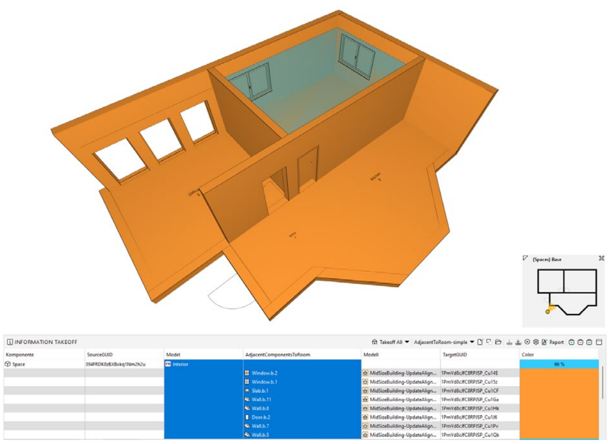

Figure 6: Extraction of all neighboring architectural elements around the Office 2 space

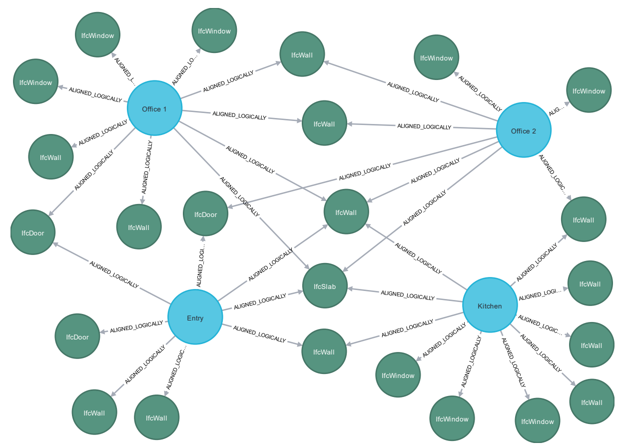

Figure 7: Subset of the model graphs reflecting the architectural and the space model components and their established object conjunctions based on the geometrical adjacency

4.2 Model Update: Moving the Inner Wall in the Architectural Model

Given the discipline models forming the conjunction graph G, a possible model change in the architectural model is considered. In the exemplary situation, the inner wall dividing the two office spaces is moved. In formal terms, this change results in a semantic modification, which is exchanged in an increment. If this adjustment has been disseminated among all project stakeholders, the aim is to provide accurate information regarding the modification to those disciplines impacted.

The affected disciplines can now consider how to respond to this change. A solution must be found that brings all models back into a coherent and consistent state. This can involve manual modifications to the discipline models or, in the case of recurring change events, setting up specific automation that converts and applies the incoming increment to the respective discipline-specific objects. For example, changes in component positions can be transferred to other linked components. In the illustrated case study, the interior designer can use the semantic modification caused by the wall movement to derive a procedural statement on how the space geometry must be altered. Subsequently, a vector describing the translation in the x and y direction can be formulated to automatically modify the affected spaces accordingly. However, such automation rules are once again challenging to define in a generalized manner because they require specific knowledge of the respective disciplines. Therefore, they should be explicitly created by the model owner for each discipline, and their application should be sensitively monitored.

Even though the illustrated modification is quite simple, further changes to the Conjunction Graph may be required to reach an overall consistent design after modifying the wall and the adjacent rooms. If the wall is moved in the illustrated manner, subsequent changes of the established conjunctions across elements of the interior and the architectural design can become necessary. Considering a wall movement, the window between the initial and the updated wall position should be reconnected to the other office space. Of course, the required transformation of the conjunction graph can be encoded into the exchanged increment as well, but again, it requires additional domain input into the formulation of each patch.

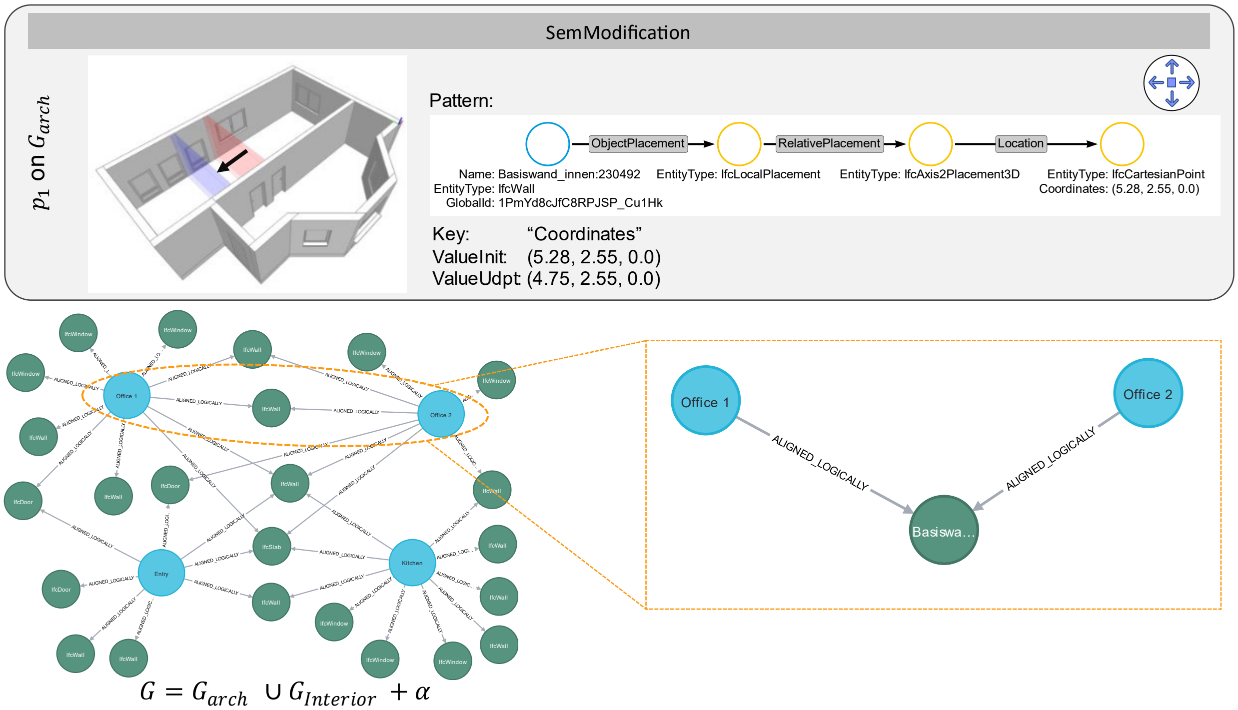

Figure 8: The transformation rule moving the inner wall and the Conjunction Graph of both models G_A and G_B together with its relationships α

5 Summary and Outlook

The proposed method maintains the established collaboration principle of federated discipline models but expands it by explicitly modeling dependencies at the element level. The demonstrated study shows how two initially disjoint models can be combined with appropriate conjunctions. By abstracting model changes into corresponding update increments, it is now possible to explicitly inform affected elements or, more precisely, the associated authors from other disciplines. Furthermore, these conjunctions can suggest suitable changes for recurring cases or even provide the basis for further automation. Additional investigation is required for model updates that insert new objects into a model, which results in an extension of the respective graph representation accordingly. Potential improvements can be reached by providing the actual update increment derived from two versions of a discipline model combined with additional conjunction information.

Besides this lack, further potential lies in the extraction process to determine sufficient object conjunctions. Applying AI-based techniques may unveil additional relationships an engineer does not think of when performing the manual extraction using the presented BIM query language approaches. We will report our results on using Graph Alignment Networks in future publications.

Acknowledgments

We gratefully acknowledge the financial support of the Deutsche Forschungsgemeinschaft (DFG) in the frame of the programs SPP 2187 "Adaptive modularized constructions made in a flux" (Project Number 423969184) and Transregio 277 "Additive Manufacturing in Construction—The Challenge of Large Scale" (Project Number 414265976).

References

[1] C. Preidel, A. Borrmann, H. Mattern, M. König, and S.-E. Schapke, "Common Data Environment," in *Building Information Modeling*, Cham: Springer International Publishing, 2018, pp. 279–291. doi: 10.1007/978-3-319-92862-3_15.

[2] K. Jaskula, E. Papadonikolaki, and D. Rovas, "Comparison of current common data environment tools in the construction industry," in *2023 European Conference on Computing in Construction & 40th International CIB W78 Conference*, Jul. 2023. doi: 10.35490/EC3.2023.315.

[3] CEN, "DIN EN ISO 19650-1:2018 Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) — Information management using building information modelling — Part 1: Concepts and principles," 2018. doi: 10.31030/3030494.

[4] DIN Deutsches Institut für Normung e.V., "DIN SPEC 91391-1: Gemeinsame Datenumgebungen (CDE) für BIM-Projekte-Funktionen und offener Datenaustausch zwischen Plattformen unterschiedlicher Hersteller-Teil 1: Module und Funktionen einer Gemeinsamen Datenumgebung," 2019. doi: 10.31030/3044838.

[5] DIN Deutsches Institut für Normung e. V., "DIN EN ISO 21597-1: Information container for linked document delivery - Exchange specification - Part 1: Container," 2021. doi: 10.31030/3137795.

[6] DIN Deutsches Institut für Normung e. V., "DIN EN ISO 21597-2: Information container for linked document delivery - Exchange specification - Part 2: Link types," 2021. doi: 10.31030/3192763.

[7] M. Senthilvel, J. Oraskari, and J. Beetz, "Common Data Environments for the Information Container for linked Document Delivery," in *Proceedings of the 8th Linked Data in Architecture and Construction Workshop-LDAC*, 2020, pp. 132–145. [Online]. Available: https://www.w3.org/community/lbd/

[8] P. Pauwels, A. Costin, and M. H. Rasmussen, "Knowledge Graphs and Linked Data for the Built Environment," 2022, pp. 157–183. doi: 10.1007/978-3-030-82430-3_7.

[9] P. Pauwels, W. Terkaj, T. Krijnen, and J. Beetz, "Coping with lists in the ifcOWL ontology," *EG-ICE 2015 - 22nd Workshop of the European Group of Intelligent Computing in Engineering*, 2015.

[10] A. Ismail, A. Nahar, and R. Scherer, "Application of graph databases and graph theory concepts for advanced analysing of BIM models based on IFC standard," in *24th International Workshop on Intelligent Computing in Engineering (EG-ICE 2017)*, 2017.

[11] J. Zhu, P. Wu, and X. Lei, "IFC-graph for facilitating building information access and query," *Autom Constr*, vol. 148, p. 104778, Apr. 2023, doi: 10.1016/j.autcon.2023.104778.

[12] E. Tauscher, H.-J. Bargstädt, and K. Smarsly, "Generic BIM queries based on the IFC object model using graph theory," in *Proceedings of the 16th International Conference on Computing in Civil and Building Engineering*, Osaka, 2016, pp. 6–14.

[13] X. Shi, Y. S. Liu, G. Gao, M. Gu, and H. Li, "IFCdiff: A content-based automatic comparison approach for IFC files," *Autom Constr*, vol. 86, no. June 2016, pp. 53–68, 2018, doi: 10.1016/j.autcon.2017.10.013.

[14] S. Esser, S. Vilgertshofer, and A. Borrmann, "Graph-based version control for asynchronous BIM collaboration," *Advanced Engineering Informatics*, vol. 53, p. 101664, Aug. 2022, doi: 10.1016/j.aei.2022.101664.

[15] S. Esser, S. Vilgertshofer, and A. Borrmann, "Version control for asynchronous BIM collaboration: Model merging through graph analysis and transformation," *Autom Constr*, vol. 155, p. 105063, Nov. 2023, doi: 10.1016/j.autcon.2023.105063.

[16] L. Zhou, M. Cheatham, A. Krisnadhi, and P. Hitzler, "A Complex Alignment Benchmark: GeoLink Dataset," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11137 LNCS, Springer Verlag, 2018, pp. 273–288. doi: 10.1007/978-3-030-00668-6_17.

[17] W. Teclaw, M. H. Rasmussen, N. Labonnote, J. Oraskari, and E. Hjelseth, "The semantic link between domain-based BIM models," in *CEUR Workshop Proceedings*, 2023. Accessed: Nov. 13, 2023. [Online]. Available: https://hdl.handle.net/11250/3101194

[18] G. Sibenik, I. Kovacic, V. Petrinas, and W. Sprenger, "Implementation of Open Data Exchange between Architectural Design and Structural Analysis Models," *Buildings*, vol. 11, no. 12, p. 605, Dec. 2021, doi: 10.3390/buildings11120605.

[19] Z. Wang, B. Ouyang, and R. Sacks, "Graph-based inter-domain consistency maintenance for BIM models," *Autom Constr*, vol. 154, p. 104979, Oct. 2023, doi: 10.1016/j.autcon.2023.104979.

[20] S. Daum and A. Borrmann, "Processing of topological BIM queries using boundary representation based methods," *Advanced Engineering Informatics*, vol. 28, no. 4, pp. 272–286, 2014, doi: 10.1016/j.aei.2014.06.001.

[21] W. Mazairac and J. Beetz, "BIMQL - An open query language for building information models," *Advanced Engineering Informatics*, vol. 27, no. 4, pp. 444–456, 2013, doi: 10.1016/j.aei.2013.06.001.

[22] K. Zeng, C. Li, L. Hou, J. Li, and L. Feng, "A comprehensive survey of entity alignment for knowledge graphs," *AI Open*, vol. 2, pp. 1–13, Jan. 2021, doi: 10.1016/j.aiopen.2021.02.002.

[23] "Graph Data Platform | Graph Database Management System | Neo4j." Accessed: Jan. 07, 2022. [Online]. Available: https://neo4j.com/

[24] Solibri, "Solibri Office." Accessed: Nov. 14, 2023. [Online]. Available: https://www.solibri.com/solibri-office