

Integrating Case-Based Reasoning with BIM Collaboration Format to Support Model Checking with Resolution Suggestions

Çeltina Balaj^{1,2}, Jiabin Wu¹, André Borrmann¹

¹Chair of Computing in Civil and Building Engineering & TUM Georg Nemetschek Institute
Technical University of Munich, Germany

²AEC3 Deutschland GmbH
celtina.balaj@tum.de

Abstract. Building Information Modeling (BIM)-based model checking primarily focuses on issue detection, with diagnostic results commonly documented and exchanged through BIM Collaboration Format (BCF) files. While BCF provides a standardized framework for issue documentation and exchange, its potential to serve as a structured repository for automated issue resolution remains largely underexplored. To support issue resolution in BIM-based model checking, this paper introduces an integrated framework that combines Case-Based Reasoning (CBR) with issue information stored in BCF files. The proposed approach reuses knowledge from previous model checking cases, dynamically enriched with spatial and semantic context extracted from Industry Foundation Classes (IFC) models to ensure case-specific relevance. A multi-dimensional similarity measure is used to retrieve relevant precedent cases. The CBR-BCF framework was developed and tested using diverse coordination scenarios. Results demonstrate that the proposed approach can identify recurring issue patterns, retrieve relevant precedent cases, and support more informed resolution decisions in BIM-based model checking workflows.

1. Introduction

Building Information Modeling (BIM) provides integrated representations of geometry, properties, and metadata across disciplines in the Architecture, Engineering, and Construction (AEC) industry. Within BIM-based design workflows, model checking plays a central role in identifying design inconsistencies and code compliance issues (Hjelseth, 2016). The results of model checking are typically documented and exchanged using the BIM Collaboration Format (BCF), which provides a standardized, tool-independent mechanism for communicating coordination issues across software environments. While BCF effectively supports issue documentation and communication, current coordination workflows remain largely focused on issue detection, with limited support for automated issue resolution.

In practice, resolution-related information in BCF files is often provided inconsistently, if at all, and is typically embedded informally in textual descriptions or comments. Consequently, practitioners must rely heavily on manual interpretation and individual expertise to determine appropriate corrective modifications (Wu, Nousias and Borrmann, 2025). This limits the systematic reuse of knowledge from previously resolved coordination issues and leads to repeated effort across projects. The lack of systematic, reusable resolution support highlights a persistent gap between issue identification and effective guidance for issue resolution.

Case-Based Reasoning (CBR) offers a promising approach for supporting the reuse of prior knowledge by linking new engineering problems to similar previously resolved cases (Kuo and Oraskari, 2016; Lee *et al.*, 2019). At its core, CBR relies on the structured storage and representation of past cases, including both problems and their corresponding solutions, so that similarities between cases can be identified and past solutions can be reused or adapted to address new problems. However, its potential to generate resolution suggestions for design issues in BIM-based model-checking workflows remains underexplored.

To support automated issue resolution in model checking, this paper presents a CBR-based framework that structures BCF issues as reusable cases enriched with IFC-derived semantic, geometric, and relational attributes. A multi-dimensional similarity metric is introduced to retrieve comparable cases by considering textual, categorical, and geometric features. Retrieved issue resolution knowledge is refined using a language model to standardize and clarify the textual resolution descriptions. The proposed approach is developed on a structured benchmark dataset and then evaluated on a range of coordination issues to assess retrieval effectiveness and the practical applicability of the generated resolution suggestions.

By reusing relevant past cases, the approach demonstrates how automated reasoning can be integrated into BIM-based model checking workflows to bridge the gap between issue detection and resolution support, reduce repetitive manual effort, and promote more efficient decision-making. Embedding CBR into BIM workflows supports knowledge reuse, lowers cognitive effort for practitioners, and contributes to a more efficient model-checking and revision process.

2. Research Background

2.1 BIM-based Model Checking and Coordination

BIM-based model checking ensures that building designs meet essential quality requirements and comply with both regulatory standards and project-specific conditions (Eastman *et al.*, 2009). The output of model checking is typically a set of identified issues that are documented, categorized, and managed through BIM coordination platforms, facilitating communication among stakeholders. BCF is commonly used for structuring checking feedback, providing a structured format to capture issue metadata such as descriptions, priorities, responsible parties, and model viewpoints (Pauwels and Zhang, 2015). Through BCF files, design issues are exchanged between different BIM tools to support design coordination.

Design issues are typically coordinated through iterative BIM coordination meetings or model review cycles. Stakeholders review BCF topics, add comments, assign responsibilities, and discuss potential corrective actions. Issues are resolved by modifying the model in discipline-specific authoring tools, after which updated models are rechecked and the corresponding BCF topics are marked as closed. While BCF enables structured issue communication, it currently provides limited assistance in guiding designers towards corrective modifications. Information on how to resolve identified issues is optional and may be embedded informally within textual descriptions or comments. Consequently, issue-resolution processes mainly rely on individual experience, leading to repeated manual effort across projects. Model checking and issue management practices remain largely focused on detection (Schulz, Oraskari and Beetz, 2021), leaving a persistent gap in structured support for effective issue resolution.

2.2 Leveraging Design Knowledge for Issue Resolution

Resolving design issues requires not only an understanding of the relevant design context but also the application of domain-specific knowledge. While concrete design modifications must ultimately be implemented within BIM authoring tools, many building design issues exhibit recurring characteristics that make similar resolution strategies transferable across projects. For instance, clashes between structural and mechanical elements and inconsistencies in space definitions often stem from comparable underlying causes and can therefore be addressed through similar approaches (Lin and Huang, 2019). Recognizing recurring resolution patterns creates opportunities to capture and reuse design knowledge more systematically.

The integration of prior design knowledge into building design processes can improve decision support, reduce repetitive manual effort, and support more efficient problem-solving in complex engineering contexts (Zhou *et al.*, 2022). When similar issues arise in different contexts, knowledge from their previous resolutions can inform subsequent decisions, thereby improving the consistency and efficiency of problem-solving. For example, some research has combined automatic rule checking with case-based reasoning to support the reuse of earlier solutions (Lee *et al.*, 2019), while other work has applied semantic analysis of BCF topics to anticipate adaptation needs from documented issue records (Kuo and Oraskari, 2016). Ontology-driven frameworks such as bcfOWL further illustrate how structured knowledge representations can support data retrieval and reasoning in BIM coordination processes (Schulz, Oraskari and Beetz, 2021, 2023).

2.3 Case-Based Reasoning for Design Knowledge Reuse

The CBR method provides a cognitively inspired methodology for solving new problems by reusing prior knowledge from similar problems, formalized through the cycle of retrieval, reuse, revision, and retention (Kolodner, 1992; Agnar and Plaza, 1994). In engineering design, CBR has been investigated as a method for supporting design reasoning by reusing previously solved design cases to inform new design problems (Maher and De Silva Garza, 1997). For instance, CBR has been applied to facility maintenance (Motawa and Almarshad, 2015; Jiang *et al.*, 2020), risk management, and construction safety (Okudan, Budayan and Dikmen, 2021).

The effectiveness of CBR systems depends on maintaining a consistent and meaningful case base, as unstructured or incomplete case representations can hinder retrieval and adaptability (Yan and Cheng, 2024). Recent studies have explored combining CBR with machine learning and natural language processing (NLP) to improve case representation and retrieval performance. For example, Wang, Meng and Zhu, (2022) combined BERT-based NLP with BIM parameters for knowledge reuse. Kuo and Oraskari, (2016) used latent semantic analysis of BCF descriptions to predict feasible design adaptations.

Despite these developments, practical implementations that directly support issue resolution within BIM-based coordination workflows remain limited. Compared with rule-based approaches that rely on predefined rules, CBR methods have the potential to support the reuse of resolution knowledge for evolving design problems through accumulated project experience. In contrast to data-intensive machine learning methods, CBR can operate effectively with relatively small case bases while preserving the interpretability of reasoning processes.

2.4 Identified Research Gap

With advances in automated model checking, most existing workflows focus mainly on issue detection. The post-checking stage, including the formulation and application of appropriate issue resolution strategies, remains underexplored and relies largely on the practical experience and domain knowledge of individual experts. As a result, an essential gap remains between issue detection and issue resolution support. Current design support approaches rarely leverage prior knowledge of design checking and issue resolution in a systematic and reusable manner. Establishing structured links between newly detected issues and already resolved cases could enable context-aware resolution suggestions, thereby improving design revision efficiency.

3. Proposed Approach

The proposed approach extends BIM-based model checking workflows by integrating CBR with BCF to support design issue resolution. The framework is applied after model checking as a post-processing layer, using BCF files as the primary input and linking them to information extracted from the associated IFC model. Each identified issue in BCF is connected to geometric, semantic, and relational information extracted from the corresponding IFC elements. As illustrated in Figure 1, the CBR system processes BCF issues, identifies similar precedent cases, and forwards the associated textual resolution suggestions for reuse and designer review.

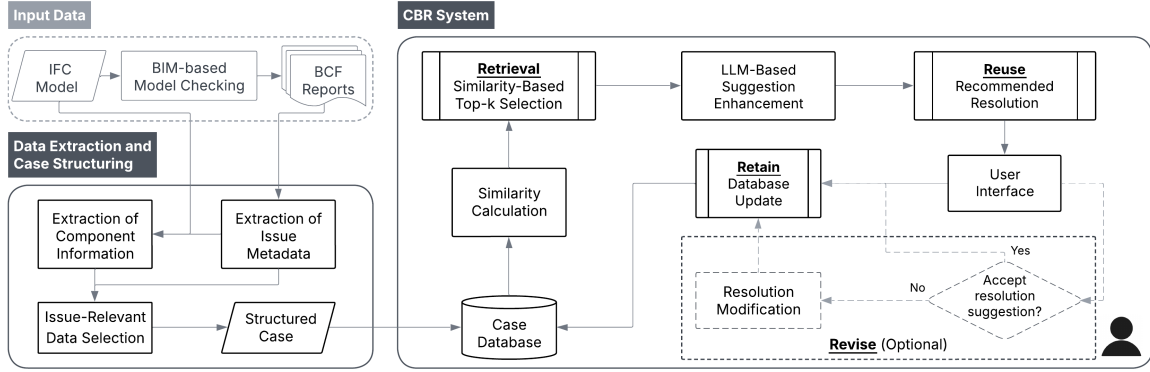


Figure 1. Proposed CBR-based framework for BCF issue resolution suggestions.

3.1 Case Structuring and Data Integration

Metadata extracted from BCF, such as the issue title, description, status, priority, labels, and referenced element GUIDs, form the initial issue representation. The GUIDs enable reliable mapping to the corresponding IFC elements, allowing the retrieval of relevant element information. Several types of IFC data are integrated: semantic attributes (e.g., IfcType, property sets, material definitions), geometric data (e.g., dimensions, bounding box coordinates, volume), and relational context (specifically, IfcRelContainedInSpatialStructure to capture spatial hierarchy and IfcRelFillsElement to identify host-filling connections). Non-essential attributes are filtered out to retain only case-relevant information, and certain geometric parameters are normalized to support meaningful comparison during retrieval.

Integrating these IFC-derived attributes ensures that each case captures the semantic, geometric, and relational context required for meaningful comparison, rather than relying solely on unstructured textual descriptions of the identified issues in BCF files. In addition to the issue context, each case includes a resolution field derived from the BCF issue description and practitioner comments, which stores precedent resolution knowledge for reuse during retrieval. Figure 2 illustrates the overarching case structure, including BCF metadata, IFC-derived attributes, and resolution suggestions.

Case_id: [str] (UUID)		
Topic / Data extracted from BCF	Topic / Data extracted from IFC	Resolution Suggestions
- Title: [str] - Type: [str] - Labels: list[str] - Priority: [str] - Description: [str] - Creation date / Author: [str] (ISO format)	- Type and Name: [str] - Dimensions: dict [str,float] - Materials: list[str] - Bounding Box: dict [str, list[float]] - Relationships: [str] / null	- Resolved by: [str] - Comments: [str] - Resolution Date: [str]

Figure 2. Case structure combining BCF, IFC-derived data, and resolution knowledge.

3.2 Similarity-Based Retrieval

Once cases are structured, the retrieval module identifies the most similar past cases for a newly reported issue by applying a multi-dimensional similarity function. The similarity assessment combines textual, categorical, geometric, and material-based comparisons to capture different aspects of issue similarity. Textual similarity s_{tex} is computed using TF-IDF vectorization and cosine similarity, enabling the system to recognize semantically similar issues even when they are described with different wording. Categorical similarity s_{cat} evaluates IFC entity types and naming conventions to ensure that retrieved cases correspond to functionally comparable elements (e.g., IfcDoor vs. IfcWall). Geometric similarity s_{geo} incorporates bounding-box intersection to capture spatial overlaps, and to compare the geometric configuration of elements in a scale-independent manner. Material similarity s_{mat} is computed using the Jaccard index, reflecting the degree of commonality between material sets.

In addition, dimension similarity s_{dim} compares the geometric dimensions of the involved elements using normalized ratios to ensure comparability across different scales. Component count similarity s_{num} evaluates the similarity in the number of elements involved in the issue. Each similarity metric contributes to the final similarity score S through predetermined weights. The similarity weights were empirically determined during the experiments according to the relative importance of each attribute for distinguishing coordination issues, ensuring a balanced comparison across all considered attributes and reducing sensitivity to missing or incomplete entries. The retrieval module ranks all stored cases according to this composite similarity score and returns the top-k most similar precedent cases for further use.

3.3 Reuse, Revision, and Retention

In the reuse step, resolution suggestions from retrieved cases are proposed as solutions for the new issue. However, practitioner-provided resolution suggestions in previously resolved cases may be incomplete or inconsistent. Therefore, a language-model-based module reformulates both the issue description and the resolution comment into a standardized format. This reduces ambiguity and harmonizes terminology across suggestions. For example, a vague instruction such as “adjust element position” is reformulated as “shift the HVAC duct upwards to meet clearance requirements.” These suggestions express the intended design modification in an actionable form, but they do not directly execute modeling commands within BIM models.

Resolution suggestions may require further adjustment before reuse, as practitioners may accept, modify, or reject the proposed solution. These revisions ensure the practical applicability of the recommended action before case retention. Once revised, the finalized case, combining the issue description with the corresponding resolution suggestion, is stored in the case base to support consistent reuse in subsequent retrieval cycles. Additionally, a maintenance strategy is applied to the CBR system by pruning cases through similarity verification. A new case is stored only if it is sufficiently distinct from existing ones, ensuring diversity and efficiency. A new case C_{new} is retained only if:

$$\max_{C_i \in C_B} S(C_{new}, C_i) < \theta \quad (1)$$

where C_B is the case base and θ is a similarity threshold.

4. Case Study and Results

4.1 Experiment Setup

To demonstrate the effectiveness of the proposed approach, a dataset of BCF-IFC issue cases was used across two phases of development and evaluation. In the development phase, a structured benchmark subset of 50 cases was compiled from selected BCF-IFC pairs (Table 1). This benchmark subset provided controlled input conditions for CBR system development and served as the basis for defining the similarity weights used in the retrieval function (Table 2).

Table 1. Issue topics covered in the development phase.

Check Type	BCF Topic
Clearance	IfcFurniture is violating the required clearance space in front of IfcDoor.
Window Area Ratio	Space windows do not cover at least 50% of the total area.
Space Accessibility	Space does not contain any entry point.
Space Area	Space has a floor area smaller than 5 m ² .
Wrong Property Set	The interior wall incorrectly contains isExternal = true.

Table 2. Similarity weights determined during the development phase.

Similarity Metric	s_{tex}	s_{cat}	s_{geo}	s_{mat}	s_{dim}	s_{num}
Weight	0.20	0.30	0.15	0.05	0.05	0.15

In the evaluation phase, performance was assessed using quantitative metrics, including top-k retrieval relevance (Manning, Raghavan and Schütze, 2008), similarity scores, and a qualitative evaluation of the suitability of the generated resolution suggestions (Table 3). To evaluate the approach’s ability to retrieve relevant precedent cases and generate appropriate resolution suggestions, 56 new issue cases, including geometric conflicts, spatial compliance issues, component deficiencies, model information quality issues, and project-specific requirement violations, were included. Compared with the development subset, these cases were intentionally formulated to reflect inconsistent phrasing, diverse issue categories, and typical irregularities found in practical coordination workflows.

Table 3. Performance indicators for assessment.

Indicator	Description
Top 1 Match Relevance	Indicates whether the most similar retrieved case is relevant to the query issue.
Top 3 Match Coverage	Indicates whether at least one of the top three retrieved cases corresponds to the type or nature of the query issue.
Resolution Acceptance Rate	Indicates whether the system-generated resolution was accepted without revision.
Average Similarity Score	Quantitative measure of similarity between the query and the retrieved top case.

For each query case, similarity-based retrieval was performed using the previously defined multi-metric similarity function, which incorporates textual, categorical, geometric, material, and IFC-derived relational attributes. Retrieved cases were ranked according to the composite similarity score, and the three most similar cases were presented through the prototype interface (Figure 3). The system then generated a textual issue description and resolution suggestion, which could optionally be reviewed and adjusted by the user.

CBR-based Resolution Suggestion Support for Model Checking

Retrieve similar model-checking issues and reuse prior knowledge for resolution.

Figure 3. Example of retrieved results for an issue related to window area ratio.

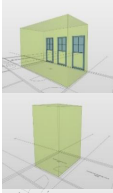
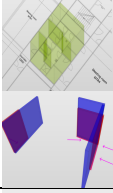
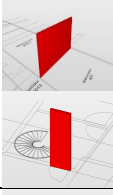
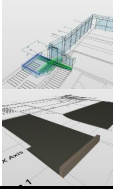
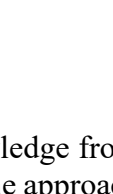
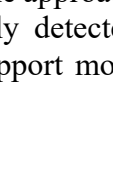
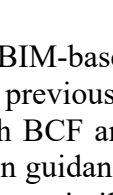
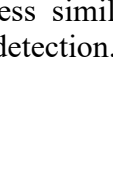
4.2 Results and Main Findings

During the development phase, all queries successfully returned a relevant top-1 match, and at least one of the top-3 retrieved cases corresponded to the correct issue type. In the evaluation phase, 62.5% of the query cases returned a relevant top-1 match, and 72.5% of the generated suggestions were accepted without modification. The experimental issues, their average similarity scores in the evaluation, and representative illustrations are presented in Table 4.

Issue categories with strong geometric or spatial characteristics, such as space accessibility, window requirements, and component duplication, achieved the highest average similarity scores. Although only door clearance issues were explicitly represented in the development phase, the system was still able to retrieve similar clearance-related and other geometry-driven issues with relatively high similarity scores (58.85%). This indicates that the proposed similarity function can capture comparable spatial and geometric patterns even across different issue formulations. In contrast, more complex compliance-related or interpretive issue types, such as path analysis, stair requirements, and architecture-structure conformity, showed lower average similarity scores. This was mainly due to the limited number of relevant precedent cases in the system during the development phase. The results show that the proposed CBR-based framework can support the transition from issue detection to structured resolution guidance by providing context-aware precedent cases and their resolution suggestions.

Table 4. Summary of coordination issues used in the case study.

Category	Issue Topic	Development Phase	Evaluation Phase	Cases	Average Similarity	Illustration
Geometric and Clearance Conflicts	Door Clearance	x	x	6	58.98%	
	Window Clearance		x	4	56.88%	

	Geometry Overlaps and Clashes		x	8	56.47%	
	Component Duplicates		x	4	68.54%	
Spatial Compliance Issues	Space Window Requirements	x	x	3	69.26%	
	Space Area/Accessibility	x	x	3	77.42%	
	Escape Route Analysis		x	4	54.72%	
	Free Floor Space		x	4	63.52%	
Deficiency Detection	Missing Components	x	x	3	65.39%	
	Architecture / Structure Conformity		x	4	49.46%	
Information Quality	Wrong Property / PropertySet	x	x	6	60.99%	
	Missing Property / PropertySet		x	5	49.80%	
Project-Specific	Stair Standard Requirements		x	3	57.54%	
	Horizontal Guarding		x	3	54.09%	

5. Conclusions

This paper presents a CBR-based approach for BIM coordination that reuses knowledge from previously resolved design issues. Through a multi-dimensional similarity metric, the approach retrieves relevant precedent cases and provides resolution suggestions for newly detected issues. The results demonstrate its potential to identify recurring patterns and support more informed issue resolution decisions in model checking workflows.

5.1 Contribution

This research has two main contributions. The first is the integration of CBR into BIM-based model checking and issue coordination processes, enabling practitioners to draw on previously resolved design issues. By combining and structuring issue-related data from both BCF and IFC files, the proposed approach captures recurring patterns and provides resolution guidance for newly detected issues. CBR-based retrieval allows users to efficiently access similar resolved cases, moving model checking and coordination workflows beyond issue detection.

The second contribution lies in the structured reuse of prior knowledge for issue resolution. Through the proposed CBR structuring principles and retrieval mechanisms, implicit knowledge embedded in model checking results and previous resolution suggestions can be systematically captured and reused in future coordination scenarios. Recognizing recurring patterns enables the context-aware retrieval of relevant precedent cases, thereby supporting more effective resolution decisions.

5.2 Limitations and Future Work

Several limitations are acknowledged in this paper. First, the similarity function in the current CBR system relies on fixed, predefined weights. Although these weights reflect the relative importance of different attributes, they remain static and do not adapt to varying issue contexts. For example, geometric attributes may be more important for clash-related problems, whereas material properties may be more relevant for fire-safety-related issues. This lack of adaptability represents a methodological limitation that should be addressed in future work through context-aware or learning-based weighting strategies. Second, while the developed CBR framework provides textual resolution suggestions as its primary output, implementing a resolution still depends on the specific design context. Even for the same issue type, different design models may require different executable modifications, although the underlying high-level resolution strategy may remain similar. This context dependency is not captured by the current system, which therefore only supports resolution guidance rather than direct modifications.

Future work will focus on expanding the case base to cover a wider variety of design issues, thereby improving the applicability of the approach across different building projects. Adaptive similarity weighting may also be introduced to better align case retrieval with the specific characteristics of each design issue. Another important step is to transform textual resolution suggestions into executable design changes through BIM authoring tool APIs, enabling more effective design corrections and further reducing the manual effort required for issue resolution. Ultimately, this research aims to proactively reuse prior issue resolution knowledge to advance BIM-based model checking and coordination toward a more efficient workflow.

Acknowledgments. This research was funded by the TUM Georg Nemetschek Institute Artificial Intelligence for the Built World (GNI).

References

- Aamodt, A. and Plaza, E. (1994) “Case-Based reasoning: Foundational issues, methodological variations, and system approaches,” *AI Communications*, 7(1). Available at: <https://doi.org/10.3233/AIC-1994-7104>.
- Eastman, C. *et al.* (2009) “Automatic rule-based checking of building designs,” *Automation in Construction*, 18(8), pp. 1011–1033. Available at: <https://doi.org/10.1016/j.autcon.2009.07.002>.
- Hjelseth, E. (2016) “Classification of BIM-based model checking concepts,” *Journal of Information Technology in Construction*, 21, pp. 354–370.
- Jiang, X. *et al.* (2020) “A decision method for construction safety risk management based on ontology and improved CBR: Example of a subway project,” *International Journal of Environmental Research and Public Health*, 17(11). Available at: <https://doi.org/10.3390/ijerph17113928>.
- Kolodner, J.L. (1992) “An introduction to case-based reasoning,” *Artificial Intelligence Review*, 6(1). Available at: <https://doi.org/10.1007/BF00155578>.
- Kuo, V. and Oraskari, J. (2016) “A predictive semantic inference system using BIM Collaboration

Format (BCF) cases and machine learning,” in *CIB World Building Congress 2016*, pp. 368–378.

Lee, P.-C. *et al.* (2019) “An efficient design support system based on automatic rule checking and case-based reasoning,” *KSCE Journal of Civil Engineering*, 23(5), pp. 1952–1962. Available at: <https://doi.org/10.1007/s12205-019-1750-2>.

Lin, W.Y. and Huang, Y.H. (2019) “Filtering of irrelevant clashes detected by BIM software using a hybrid method of rule-based reasoning and supervised machine learning,” *Applied Sciences (Switzerland)*, 9(24). Available at: <https://doi.org/10.3390/app9245324>.

Maher, M. Lou and De Silva Garza, A.G. (1997) “Case-based reasoning in design,” *IEEE Expert-Intelligent Systems and their Applications*, 12(2). Available at: <https://doi.org/10.1109/64.585102>.

Manning, C.D., Raghavan, P. and Schütze, H. (2008) “Introduction to Information Retrieval,” *Cambridge: Cambridge University Press*. Available at: <https://doi.org/DOI:10.1017/CBO9780511809071>.

Motawa, I. and Almarshad, A. (2015) “Case-based reasoning and bim systems for asset management,” *Built Environment Project and Asset Management*, 5(3), pp. 233–247. Available at: <https://doi.org/10.1108/BEPAM-02-2014-0006>.

Okudan, O., Budayan, C. and Dikmen, I. (2021) “A knowledge-based risk management tool for construction projects using case-based reasoning,” *Expert Systems with Applications*, 173, p. 114776. Available at: <https://doi.org/10.1016/j.eswa.2021.114776>.

Pauwels, P. and Zhang, S. (2015) “Semantic rulechecking for regulation compliance checking: an overview of strategies and approaches,” in *Proceedings of the 32nd CIB W78 Conference*.

Schulz, O., Oraskari, J. and Beetz, J. (2021) “bcfOWL: A BIM collaboration ontology,” in *Proceedings of the 9th Linked Data in Architecture and Construction Workshop, LDAC*.

Schulz, O., Oraskari, J. and Beetz, J. (2023) “Lessons learned from designing and using bcfOWL,” in *LDAC2023 - 11th Linked Data in Architecture and Construction Workshop*.

Wang, H., Meng, X. and Zhu, X. (2022) “Improving knowledge capture and retrieval in the BIM environment: Combining case-based reasoning and natural language processing,” *Automation in Construction*, 139, p. 104317. Available at: <https://doi.org/10.1016/j.autcon.2022.104317>.

Wang, R. *et al.* (2024) “Decision-guidance method for knowledge discovery and reuse in multi-goal engineering design problems,” *Advanced Engineering Informatics*, 61, p. 102502. Available at: <https://doi.org/10.1016/j.aei.2024.102502>.

Wu, J., Noursias, S. and Borrmann, A. (2025) “Design Healing framework for automated code compliance,” *Automation in Construction*, 171, p. 106004. Available at: <https://doi.org/10.1016/j.autcon.2025.106004>.

Yan, A. and Cheng, Z. (2024) “A Review of the Development and Future Challenges of Case-Based Reasoning,” *Applied Sciences (Switzerland)*. Available at: <https://doi.org/10.3390/app14167130>.

Zhou, X. *et al.* (2022) “Automated structural design of shear wall structures based on modified genetic algorithm and prior knowledge,” *Automation in Construction*, 139, p. 104318. Available at: <https://doi.org/10.1016/j.autcon.2022.104318>.