



Model checking methods for BIM: The state-of-the-art analysis

Modelprüfungsmethoden für BIM:
Eine Analyse des aktuellen Stands der Technik

by

Vladislav Jos

Matriculation number: 5059603

The project submitted to the Institute of Construction Informatics,
Faculty of Civil Engineering, Dresden University of Technology,
in partial fulfilment of the requirements for the degree of
Master of Science

Examiners:

Prof. Dr.-Ing. habil. Karsten Menzel

Prof. Dr.-Ing. Raimar J. Scherer

Supervisor:

- M.Sc. Prathap Valluru

Dresden, May 2023

Task Sheet. 1 page

Task Sheet. 2 page

DECLARATION

I hereby declare that the Application-Oriented Research Project (M1400-ACO08) entitled "Model checking methods for BIM: The state-of-the-art analysis" (Modellprüfungsmethoden für BIM: Eine Analyse des aktuellen Stands der Technik), which is being submitted in partial fulfilment of the requirements for Masters of Science in Advanced Computational and Civil Engineering Structural Studies at Technische Universität Dresden, is carried out by me. I confirm that the report has been written independently and all the sources used have been specified.

This work has not yet been submitted to another examination institution, either in Germany or outside of Germany, nor has it been published in the same or in a similar way.

Last name: Jos

First name: Vladislav

Matriculation number: 5059603

Dresden, 22.05.2023

Signature

ACKNOWLEDGEMENT

I would like to express my heartfelt gratitude to my supervisor, M.Sc. Prathap Valluru, whose contributions were invaluable to the completion of this project. His guidance, support, and mentorship were instrumental in navigating the complexities of the project, and his constant encouragement and patience kept me motivated throughout the process.

I would also like to extend my appreciation to the examiners of this project work, Prof. Dr.-Ing. habil. Karsten Menzel and Prof. Dr.-Ing. Raimar J. Scherer, for their valuable insights, guidance, and constructive feedback. Their expertise and support were invaluable to the successful completion of this project and have been critical to my development as a student and as a professional.

I ABSTRACT

Building Information Modeling (BIM) is an essential tool in the Architecture, Engineering, and Construction industry, and ensuring the quality of BIM models is crucial for successful project delivery. One critical aspect of BIM quality assurance is model checking, which involves verifying the accuracy and consistency of BIM models against specific requirements. However, the execution of BIM-based model checking (BMC) remains poorly defined, and available methods are not adequately classified to support practical implementation. To address this issue, this study focuses on conducting a state-of-the-art analysis of BIM-based model checking methods. The research aims to develop a comprehensive classification of model checking methods based on potential verification scenarios and to analyze available current tools that aid the implementation of these methods. A literature review is conducted to identify existing classifications of BIM-based model checking methods and gather up-to-date information on each of the approaches. The study proposes a classification scheme for BMC, categorizing the methods into three general classes based on the types of checking scenarios they are designed for: clash detection, model content checking, and code checking. Each of the presented types is discussed in detail, providing subcategorization and examples. The work also provides an analysis of available software tools for model checking, covering aspects such as program features, compatibility with BIM software, exchange formats, and cost. Additionally, the research demonstrates two model checking methods using a small-scale BIM model and one of the contemporary software tools. Overall, the results of this project work contribute to the understanding of BIM-based model checking methods and their importance in the Architecture, Engineering and Construction industry.

II TABLE OF CONTENTS

I	Abstract.....	VII
II	Table of Contents	IX
III	List of Figures	XI
IV	List of Tables	XIII
V	List of Abbreviations	XV
1	Introduction	1
1.1.	Building Information Modeling	2
1.2.	BIM-based Model Checking	3
2	Classification of BMC methods.....	7
2.1.	Clash detection	7
2.2.	Model content checking.....	11
2.3.	Code checking	13
3	Process of code checking	17
3.1.	Four-step algorithm.....	17
3.2.	Complexity and editing	20
4	Model checking tools	23
5	Demonstration of model checking methods.....	27
5.1.	Model for demonstration	27
5.2.	Clash detection. Demonstration of method	36
5.3.	Model content checking. Demonstration of method	43
6	Conclusion	45
7	References.....	i

III LIST OF FIGURES

Figure 1. Classification of BIM-based model checking methods.....	7
Figure 2. Classification of clash detection.....	8
Figure 3. Hard clash between the wall and the ventilation duct in Autodesk Navisworks Manage	9
Figure 4. Hard clash between the floor slab and the pipe in Autodesk Navisworks Manage	9
Figure 5. Soft clash between the tank and the surrounding walls in Autodesk Navisworks Manage (Herrera, 2021)	10
Figure 6. Soft clash between the two pipes in Autodesk Navisworks Manage (Herrera, 2021).....	10
Figure 7. Classification of model content checking	11
Figure 8. Checking the presence of components in spaces in Solibri Model Checker (Bolpagni, 2013)	12
Figure 9. Classification of code checking	13
Figure 10. Checking width of the corridor in Solibri Model Checker (Soliman-Junior et al., 2021).....	14
Figure 11. Checking distance between the two spaces in Solibri Model Checker (Bolpagni, 2013).....	15
Figure 12. Checking the value of sound transmission loss of the walls in Solibri Model Checker (Bolpagni, 2013)	15
Figure 13. Process of code checking. Four-step algorithm	17
Figure 14. Different classes of complexity	20
Figure 15. White-box and black-box approaches of editing	21
Figure 16. Beyer-Bau BIM model in Autodesk Revit	28
Figure 17. Small-scale part of the Beyer-Bau BIM model in Autodesk Revit	28
Figure 18. Plan of the 2nd floor of the Beyer-Bau building	29
Figure 19. WC room with MEP elements.....	29

Figure 20. MEP elements.....	29
Figure 21. Visual representation of two different models in separate files.....	30
Figure 22. IFC Options in Autodesk Revit	32
Figure 23. Setting IFC Export Classes in Autodesk Revit.....	32
Figure 24. Export to IFC in Autodesk Revit	33
Figure 25. IFC export settings in Autodesk Revit	33
Figure 26. Modifying setup of IFC export in Autodesk Revit. General settings.....	34
Figure 27. Modifying setup of IFC export in Autodesk Revit. Advanced settings	34
Figure 28. Visual check of the model after import in Solibri Model Checker	35
Figure 29. "Model Tree" window in the Solibri Model Checker interface	36
Figure 30. Choosing a discipline for uploaded files in Solibri Model Checker	36
Figure 31. Choosing a role in Solibri Model Checker	37
Figure 32 Uploading a ruleset from a directory in Solibri Model Checker	37
Figure 33 Information about selected ruleset or rule in Solibri Model Checker	38
Figure 34. Parameters of the rule in Solibri Model Checker	39
Figure 35. Results of clash detection checking in Solibri Model Checker	40
Figure 36. Grouping of clashes in Solibri Model Checker.....	41
Figure 37. Collision between water pipes and wall in Solibri Model Checker.....	42
Figure 38. Collision between ductworks/pipes and floor slab in Solibri Model Checker..	42
Figure 39. Results of model completion checking in Solibri Model Checker	43
Figure 40. Existing BIM model elements in Solibri Model Checker	44

IV LIST OF TABLES

Table 1. BMC classification proposed by Hjelseth	4
Table 2. BMC classification proposed by Succar	5
Table 3. List of software for BMC	23
Table 4. Analysis of model checking tools	25
Table 5. Model checking methods for demonstration	30

V LIST OF ABBREVIATIONS

BIM	Building Information Modeling
BMC	BIM-based Model Checking
AEC	Architecture, Engineering and Construction
QA	Quality Assurance
QC	Quality Control
PAIS	Process-Aware Information Systems
BPM	Business Process Management
NLP	Natural Language Processing
AI	Artificial Intelligence
IDM	Information Delivery Manual
MVD	Model View Definition
IDS	Information Delivery Specification
ISO	International Organization for Standardization
LOI	Level of Information
PAS	Publicly Available Specification
RASE	Requirement, Applicability, Selection, Exception
SWRL	Semantic Web Rule Language
IFC	Industry Foundation Classes
OWL	Ontology Web Language
SMC	Solibri Model Checker
MEP	Mechanical, Electrical, and Plumbing

1 INTRODUCTION

The construction industry has seen a major shift towards digitalization in recent years, with Building Information Modeling (BIM) at the forefront of this change. BIM is a methodology that facilitates digital collaboration among various stakeholders involved in the planning, execution, and operation of construction projects. The accuracy and quality of BIM models are crucial in ensuring that the final product meets the required specifications. Consequently, auditing BIM services plays a pivotal role in ensuring the quality of services provided and creating trust in the BIM methodology.

However, while the checking of BIM models is an integral part of the BIM methodology, the execution of this process remains poorly defined and largely manual. Moreover, it is important to differentiate between two types of verification when it comes to BIM models: geometric verification, which has been incorporated into various software solutions, and semantic verification. Therefore, there is a need for clearly formulated classification of available methods for checking BIM models.

This project work is devoted to the state-of-the-art analysis of model checking methods for BIM. The goal is to develop a comprehensive classification of model checking methods based on potential verification scenarios, considering the currently applied methods of BIM-based model checking. The project work does not consider in detail any methods that are under research and development.

The structure and the corresponding tasks of the project work, as provided in the task sheet, comprises several milestones. Firstly, the introduction will provide a concise definition of key concepts such as BIM technologies and BIM-based model checking (BMC). Subsequently, a comprehensive literature review will be conducted, with the primary aim of identifying existing classifications of BIM-based model checking methods and gathering up-to-date information on each of these methods. The next step will involve analyzing and organizing the collected information according to the structure outlined in the issued task sheet. Furthermore, certain model checking methods will be demonstrated using a small-scale BIM model as an illustrative example. In the final section, the results obtained from the conducted research will be summarized and presented.

The scope of the project work includes compiling the model checking methods and tools for BIM, analyzing the model checking methods, and demonstrating two model checking methods with a small-scale BIM model. The limitations of the work should be noted, with the project work focusing only on the currently applied methods of BMC.

1.1. BUILDING INFORMATION MODELING

Nowadays it is difficult to imagine the modern AEC industry (Architecture, Engineering and Construction) industry without the use of Building Information Modeling technologies. BIM has emerged as a fully-fledged, independent field that has shown impressive growth and development in recent years.

There are various definitions of the term BIM, but this project work will not delve into them extensively. The current work will provide only a concise definition that encapsulates the core concept: Building Information Modeling is a collaborative digital approach that involves creating, managing, and sharing detailed building information throughout the entire construction project lifecycle (Eastman et al., 2011). "Creating" in this case means the process of developing a detailed digital model of a building or structure, including its physical and functional characteristics. "Managing" refers to the process of organizing and maintaining the BIM data that tasks such as updating the BIM model as design changes are made, coordinating with other project team members to ensure the model is accurate and up-to-date, and ensuring that all necessary information is included in the model. "Sharing" refers to the process of making the BIM data available to all project team members and stakeholders who need access to it. This collaborative work includes sharing the BIM model itself as well as any related documentation or data, such as material specifications or construction schedules.

The benefits and opportunities offered by BIM have been analyzed in a large number of scientific papers, such as Azhar, 2011, Bryde et al., 2013, Volk et al., 2014, Ghaffarianhoseini et al., 2017. Improved efficiency, increased productivity, better safety and risk management, cost savings, and stronger stakeholder engagement are among the key benefits and opportunities that can be attained. To enable the realization of the aforementioned points one of the integral components of the BIM domain is applied – BIM-based model checking.

1.2. BIM-BASED MODEL CHECKING

In the most general definition, BIM-based model checking is a processing of information contained in a BIM model according to pre-established rules (Andrich et al., 2022; Aydın, 2022; Hjelseth et al., 2016). BMC plays a crucial role in ensuring the accuracy, integrity, and quality of a BIM model, and it offers several benefits, including:

- 1) Quality Assurance (QA) and Quality Control (QC);
- 2) Improved coordination and collaboration;
- 3) Better risk management;
- 4) Increased efficiency and cost saving.

Each of the points is discussed in more detail below.

QA and QC: BMC enables the identification of errors, discrepancies, and inconsistencies in a BIM model, helping to ensure that the model complies with project requirements, standards, and codes. It helps identify issues such as overlapping elements, incorrect dimensions, clashes between different building systems and others, which can be corrected early in the design process (Hjelseth et al., 2016).

Improved coordination and collaboration: BMC facilitates better coordination among project stakeholders by allowing them to identify and resolve clashes and conflicts between different building elements and systems in the BIM model. This leads to smoother coordination and collaboration among various project participants, resulting in improved project outcomes (Hjelseth et al., 2016).

Better risk management: BMC helps identify potential risks and issues in the BIM model, such as constructability issues, code violations, and design conflicts, allowing them to be addressed proactively. This minimizes risks associated with construction delays, cost overruns, and rework, resulting in better risk management and mitigation (Zou et al., 2015).

Increased efficiency and cost saving: BMC enhances the efficiency of the construction process by automating repetitive and time-consuming tasks, such as manual checking of clashes and errors in BIM models. This saves time and reduces the chances of human errors, leading to increased productivity and cost savings. Moreover, early detection and resolution of issues through BMC prevent costly rework and construction delays, resulting in cost savings and improved project timelines (Andrich et al., 2022; Hjelseth, 2016).

1.2.1. EXISTING BMC CLASSIFICATIONS IN LITERATURE

Checking BIM models is a crucial step in the BIM methodology, and there is a need for a clearly formulated classification of the available methods for conducting this process. As a result of literature review, it was possible to identify two of the broadest and most general classifications of methods for checking BIM models.

The first classification system is described by Hjelseth, 2016. In this research an ontological methodology was employed to construct a comprehensive framework for the classification of distinct BMC concepts. This study gives the representation of the following BMC concepts (Table 1):

- 1) Compliance checking solutions, which can be further categorized into "Validation checking" and "Model content checking";
- 2) Design solution checking, which comprises "Smart objects checking" and "Design option checking".

Table 1. BMC classification proposed by Hjelseth

Group	Type	Example
Compliance checking	Validation checking	clash detection, code compliance
	Model content checking	relevant data for sharing
Design solution checking	Smart object checking	dimensions of building components (objects) associated with the building (model)
	Design option checking	knowledge system for selecting relevant solutions

In this case, the disadvantages of this classification are that the second group "Design solution checking" reflects categories that are still under research and development and are not widely used in practice. At the same time, doubts arise about whether this category belongs to BIM-based model checking, since such concepts as "Smart object checking" and "Design option checking" may be more related to the field of process automation in BIM modeling.

The second classification is proposed by Succar, 2009 and developed by Hjelseth, 2015.

This concept describes the development of a taxonomy to classify the level of BMC compliance and content checking. This taxonomy is based on two criteria or taxa:

- 1) Requirement of information content in the BIM file;
- 2) Complexity of the rules or rulesets.

Table 2 provides an overview of the five levels and their corresponding descriptions. These levels are developed in accordance with the special relationships that are defined between the two described criteria. This relationships are presented in detail in a matrix in the paper published by Hjelseth, 2015

Table 2. BMC classification proposed by Succar

Level	Description
1	Clash detection checking
2	Adjusted model checking
3	Specific purpose checking
4	Integrated model checking
5	Pervasive model checking

The significant drawback of this model is the absence in the classification of such a fundamental direction for BMC as "clash detections". Moreover, the classification based on the proposed criteria is not entirely suitable for understanding the methods used for checking BIM models and for forming a clear structure.

In the process of reviewing the literature, it was not possible to identify the classifications of BIM-based model checking methods based on the principle of various model checking scenarios. While scientific sources referenced all types of checking methods, which are discussed in the core part of the present work (2 Classification of BMC methods), none of them were structured or represented in a single detailed scheme.

1.2.2. CHALLENGES AND OPPORTUNITIES OF BMC

This part presents a brief description of several works that reflect some new challenges and possible area of interest for further research in the BMC scope.

Considering the current state of scientific research, several prominent topics have emerged in recent years that are actively being studied and possess significant potential for academic investigation. These topics have also brought about challenges that are yet to be resolved, and warrant further investigation in order to fully comprehend their impact on their respective fields of study.

As previously mentioned in the introduction, model checking is a crucial process in ensuring that building designs are compliant with the certain codes and regulations. However, this process can be time-consuming and arduous, and there is a potential for mistakes and tampering with the outcomes throughout the procedure (Gao & Zhong, 2022). To address these issues, automated compliance checking technologies have been developed, which can reduce human errors and output drawing review reports automatically (Gao & Zhong, 2022; Sun & Kim, 2022).

Despite the benefits of automated compliance checking, the use of BIM-based Model Checking systems in building design practices remains limited, largely due to socio-technical challenges (Gade & Svidt, 2021). These challenges include cultural resistance, lack of transparency, and inflexibility (Gade et al., 2018; Gade et al., 2022; Gade & Svidt, 2021). Several studies have proposed solutions to improve the flexibility of BMC, such as using PAIS/BPM (Process-Aware Information Systems / Business Process Management) theories to set technical requirements for a prototype that integrates BIM-model information in a BPM environment (Gade et al., 2018), or developing a NLP-based (Natural Language Processing) semantic framework that implements rules-based automated compliance checking for BIM at the design stage (Zhou et al., 2022).

To further improve the efficiency and accuracy of compliance checking, some studies have proposed the integration of AI technology into the process. Sun & Kim, 2022 proposed the use of AI (Artificial Intelligence) technology to identify BIM objects and their observable characteristics, which could enhance the usability and functionality of automated code checking. Meanwhile, Shen et al., 2022 proposed an automated system that detects and mitigates potential safety hazards during the construction process by combining an ontology-based safety rule database with Natural Language Processing techniques.

The study of Guedes et al., 2021 have focused on identifying the essential characteristics that need to be inserted into BIM models of airport design projects for evaluation using code checking. The paper written by Ren & Zhang, 2019 aimed to develop a new approach for automatically checking information completeness of BIM models in order to support BIM usage in structural analysis in an interoperable manner.

2 CLASSIFICATION OF BMC METHODS

The methods for BIM-based model checking can be categorized into three general classes, depending on the types of checking scenarios they are designed for:

- 1) Clash detection (collision detection);
- 2) Model content checking;
- 3) Code checking (compliance checking, rule checking).

An illustrative diagram depicting the proposed scheme is provided in Figure 1.

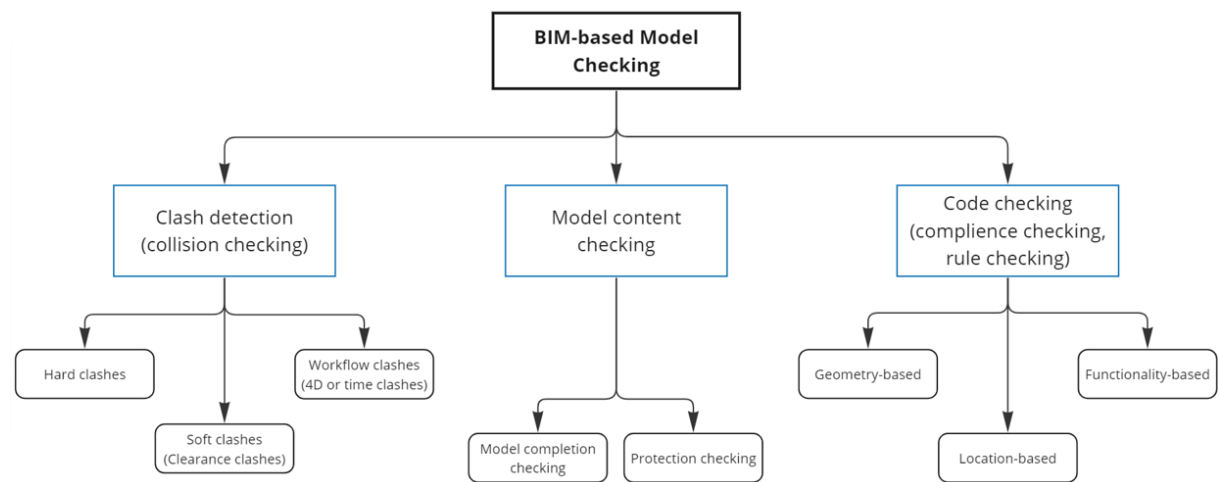


Figure 1. Classification of BIM-based model checking methods

2.1. CLASH DETECTION

Clash detection or collision checking is a process of identifying collisions or interferences between different building elements (Abdalhameed & Naimi, 2023; Akhmetzhanova et al., 2022).

There are generally three types of clashes (Figure 2):

- 1) Hard clashes;
- 2) Soft clashes;
- 3) Workflow clashes.

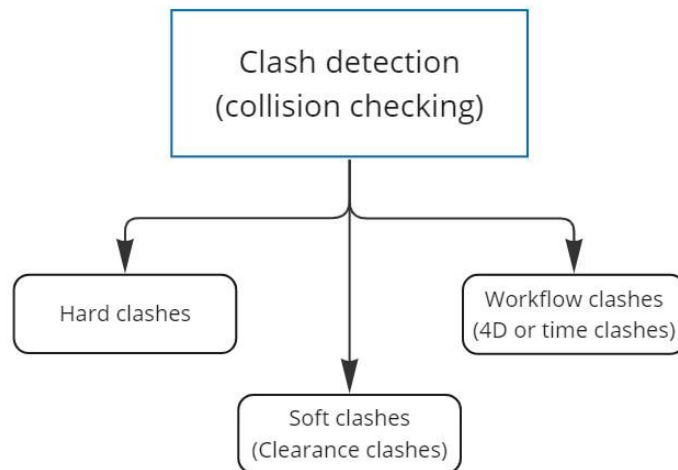


Figure 2. Classification of clash detection

2.1.1. HARD CLASHES

Hard clashes refer to physical collisions between building elements that occupy the same space or interfere with each other (Savitri et al., 2020). For example, it could be intersections between:

- 1) a wall and a ductwork (Figure 3);
- 2) a floor slab and a pipe (Figure 4);
- 3) a pipe and an electrical conduit.

2.1.2. SOFT CLASHES

Soft clashes, also known as clearance clashes indicate that an object has not been given sufficient geometric tolerances or if its buffer zone has been violated. In this scenario building elements come too close to each other, but do not physically intersect (Chidambaram, 2020). There are some examples that could be perceived as soft clashes:

- 1) lack of space for maintaining a tank in an engine room (Figure 5);
- 2) no safety distance between pipes (Figure 6);
- 3) short distance between a heater and floor level;

2.1.3. WORKFLOW CLASHES

Workflow clashes, also known as 4D or time clashes, are clashes that occur between different disciplines or trades involved in the construction process (Savitri et al., 2020). For example, it could be a clash between the structural and mechanical systems, or between the electrical and plumbing systems. Workflow clashes can cause coordination issues, delays, and rework during construction.

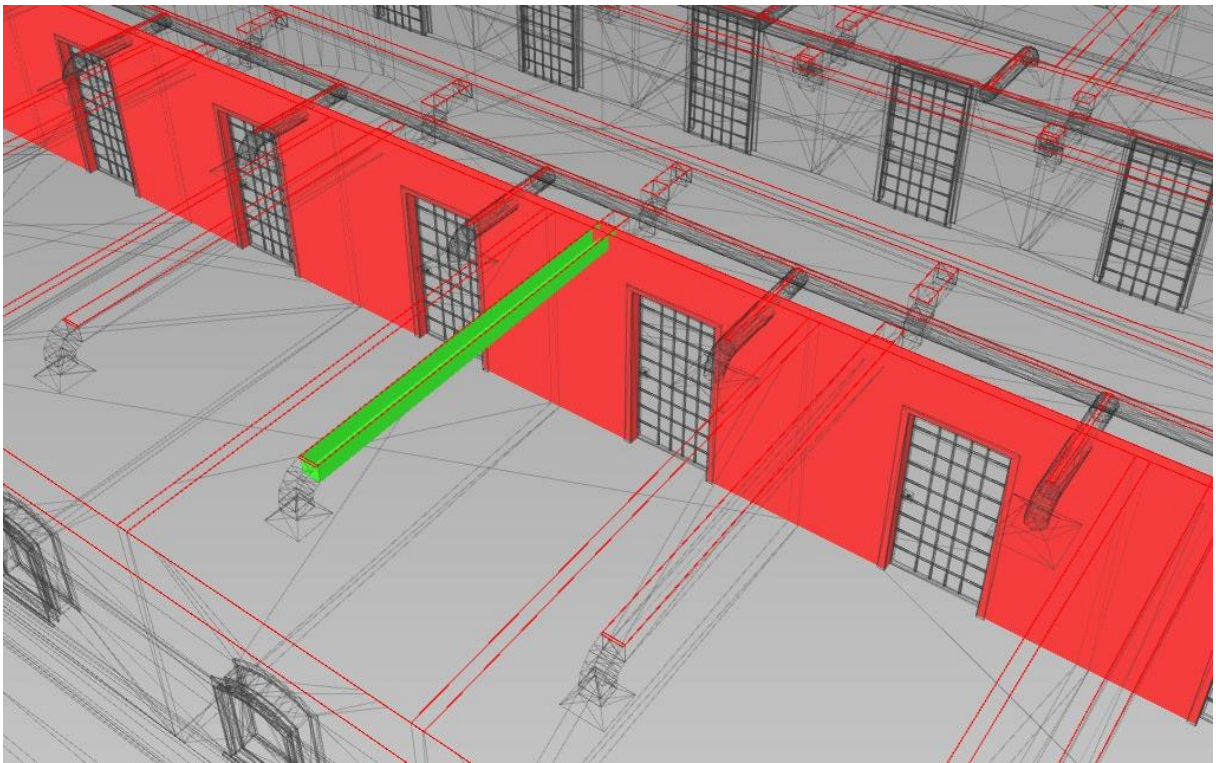


Figure 3. Hard clash between the wall and the ventilation duct in Autodesk Navisworks Manage

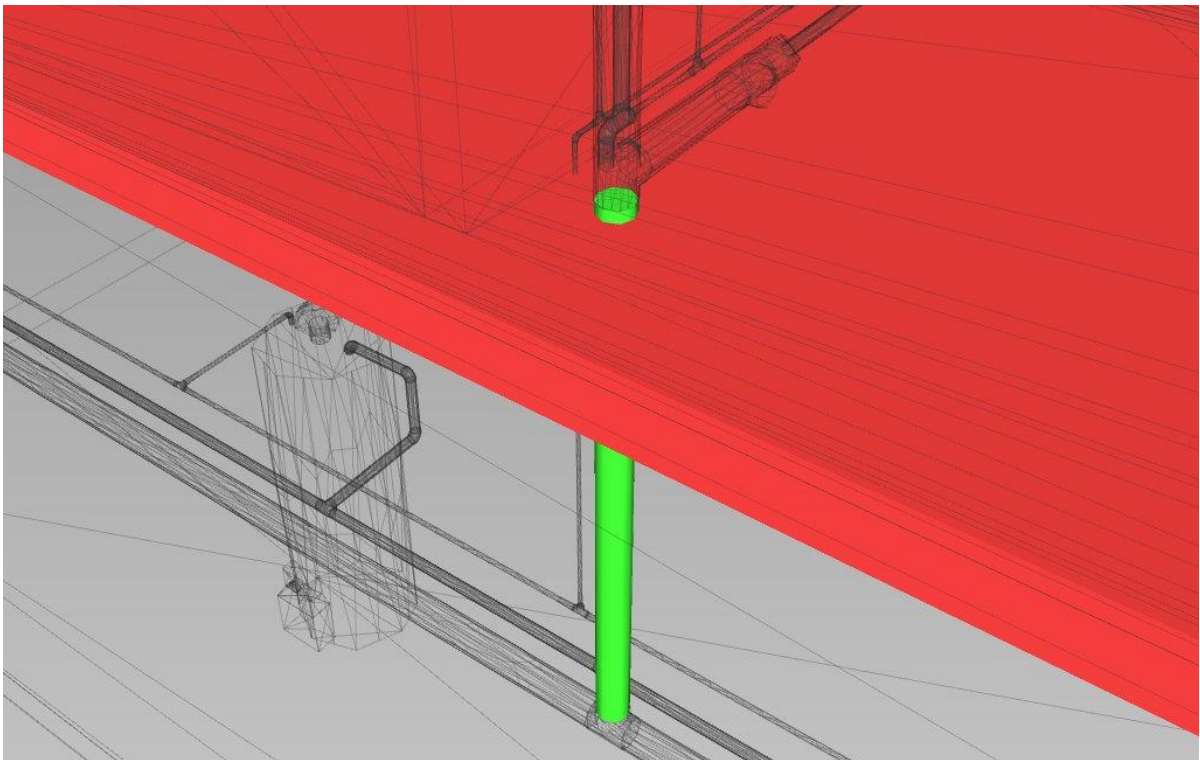


Figure 4. Hard clash between the floor slab and the pipe in Autodesk Navisworks Manage

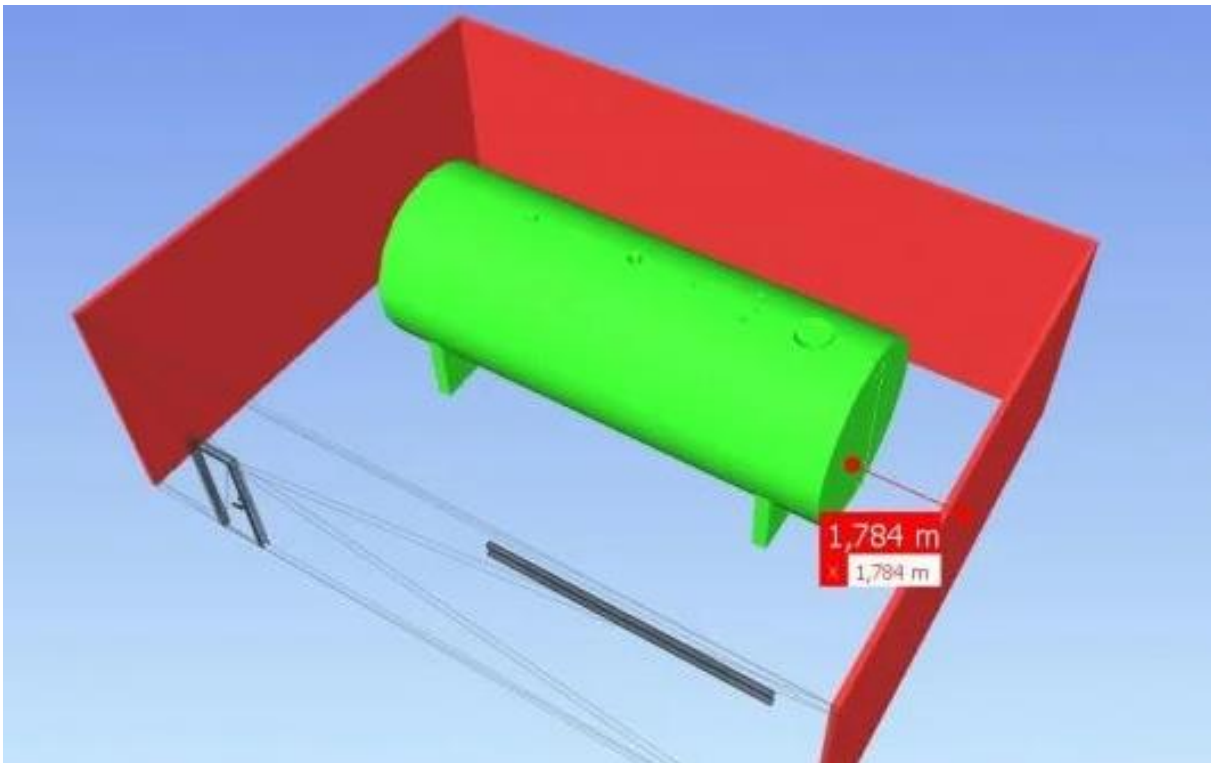


Figure 5. Soft clash between the tank and the surrounding walls in Autodesk Navisworks Manage (Herrera, 2021)

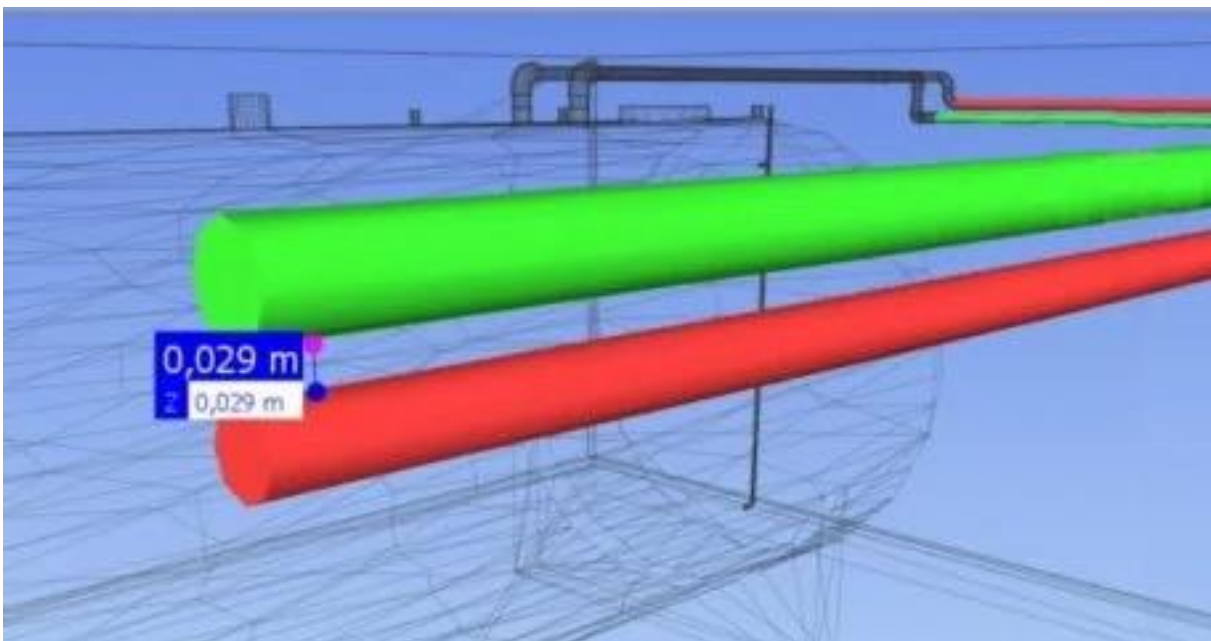


Figure 6. Soft clash between the two pipes in Autodesk Navisworks Manage (Herrera, 2021)

2.2. MODEL CONTENT CHECKING

Model content checking is a process of verifying the completeness of the data contained within a BIM model for a specific purpose. This process involves applying filters or rules to automatically identify relevant information, which can then be further analyzed in BMC software, spreadsheets, word processors, or databases.

Content checking can be particularly relevant during the handover of the BIM model or before performing calculations and other information-intensive tasks, such as code (compliance) checking. It can help to ensure that a BIM model contains the necessary information required by the project specifications, contracts, or other documents, and can also be used to check for the presence of too much information (Hjelseth, 2016).

The Figure 7 shows two main types of content checking: model completion checking (lack of information) and protection checking (excess of information).

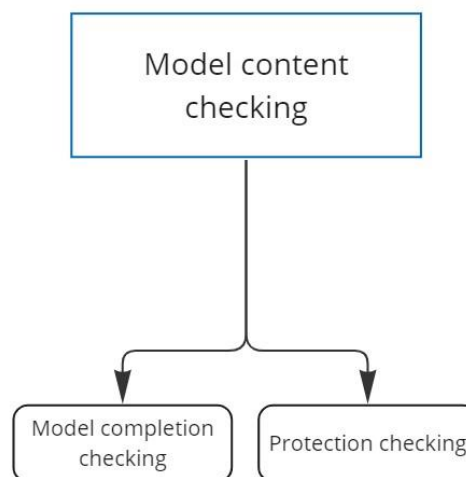


Figure 7. Classification of model content checking

2.2.1. MODEL COMPLETION CHECKING

Model completion checking focuses on verifying the compliance between the information present in a BIM model and a predefined list of requirements. This form of content checking ensures that the BIM model contains all the necessary information according to the project requirements and standards (Figure 8).

The requirement list for model completion checking can be derived from diverse sources such as BIM manuals, BIM guidance, buildingSMART Processes, Information Delivery Manuals (IDM) based on ISO 29481-1:2016 standard, Information Delivery Specifications (IDS), Data Drop or Level of Information (LOI) according to PAS 1192-2, or classification systems (Hjelseth, 2016).

Model completion checking can be integrated into the information exchange processes throughout the lifecycle of construction projects to support collaboration among stakeholders. It helps in maintaining client requirements throughout the project lifecycle. This ensures that the design process and its components go through various stages from proposal to demolishing while keeping track of the changes in the status of information. This helps in improved control of client demands and the level of their satisfaction.

2.2.2. PROTECTION CHECKING

Protection checking is a type of content checking that ensures that a BIM model does not contain any information that should not be accessible to others. This type of content checking is related to safety, security, commercial information, and intellectual property protection (Hjelseth, 2016).

Protection checking involves removing or securing sensitive information in a BIM model to prevent unauthorized access or disclosure. For instance, this can include information related to ventilation systems, electrical systems, access or security systems, commercial pricing, proprietary designs, and other sensitive data that should not be shared with external parties.

Protection checking is important to ensure that a BIM model is compliant with safety and security requirements, protects commercial information, and safeguards intellectual property. This type of content checking is typically done before transferring or exporting the BIM model for use by external stakeholders, ensuring that only relevant and approved information is shared while sensitive information is removed or secured.

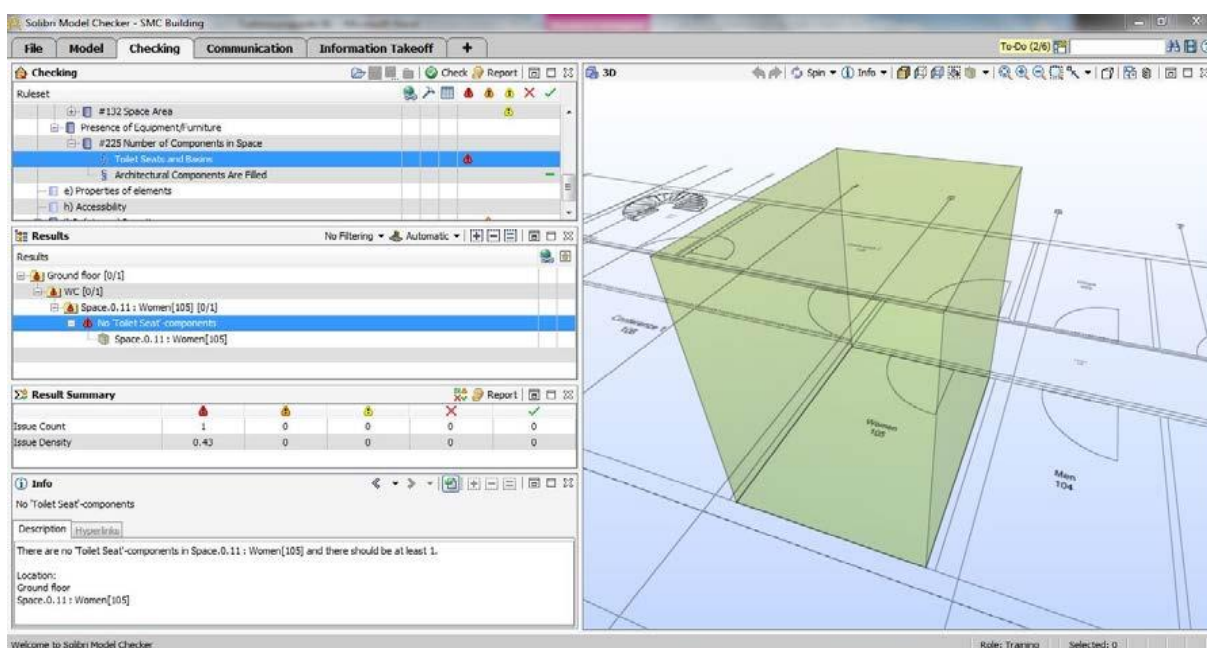


Figure 8. Checking the presence of components in spaces in Solibri Model Checker (Bolpagni, 2013)

2.3. CODE CHECKING

Code checking, compliance checking or rule checking refers to the process of validating a BIM model against predefined rules, regulations, standards, or guidelines to ensure that it complies with established requirements (Schwabe et al., 2019).

Code checking can be broadly categorized into three groups based on different aspects of the BIM model (Figure 9):

- 1) Geometry-based checking;
- 2) Location-based checking;
- 3) Functionality-based checking.

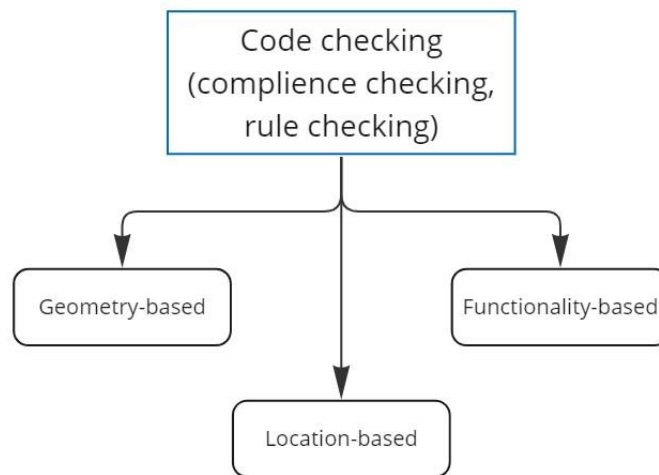


Figure 9. Classification of code checking

2.3.1. GEOMETRY-BASED CODE CHECKING

Geometry-based code checking verifies geometric aspects of a BIM model, ensuring that building components are modeled correctly in terms of their size and shape. In other words, it involves verifying if the geometry of building elements, such as walls, roofs, slabs, doors, windows, and other architectural and structural elements, comply with the required standards and design specifications.

For example, this type of checking verifies if:

- 1) a width of a corridor is \geq defined value (Figure 10);
- 2) a glazing area of a wall is \leq defined value;
- 3) a riser height of a staircase meets defined requirements.

2.3.2. LOCATION-BASED CODE CHECKING

Location-based code checking is about spatial relationships and positioning of building components within a BIM model. It checks if the components are correctly located and aligned with respect to each other and the building site.

For example, this type of checking verifies:

- 1) distances between specified objects (Figure 11);
- 2) accessibility of disabled parking spaces;
- 3) minimum height of handrails.

2.3.3. FUNCTIONALITY-BASED CODE CHECKING

Functionality-based code checking validates functional aspects of a BIM model, controlling that building components are correct in terms of their intended use and performance. It involves verifying if the functional requirements are met according to the established rules, regulations, and design standards.

For example, this type of checking verifies if:

- 1) sound transmission loss is sufficient for a specified object (Figure 12);
- 2) flow rate of ventilation diffuser is enough for a specified space;
- 3) number of parking spaces is enough.

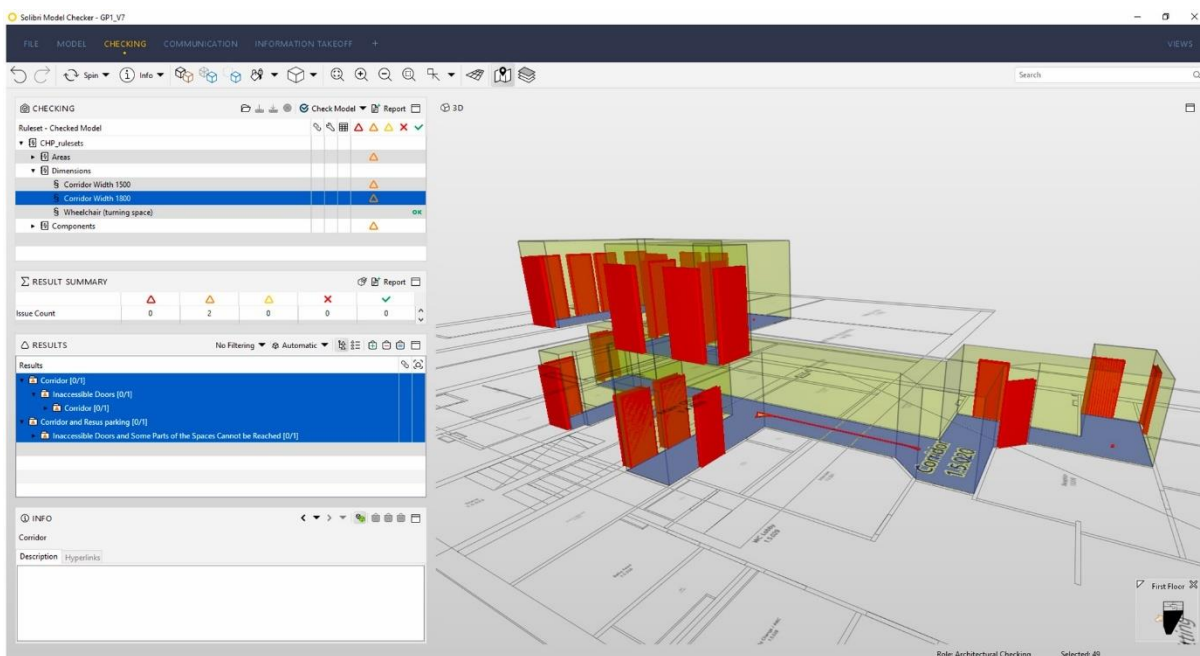


Figure 10. Checking width of the corridor in Solibri Model Checker (Soliman-Junior et al., 2021)

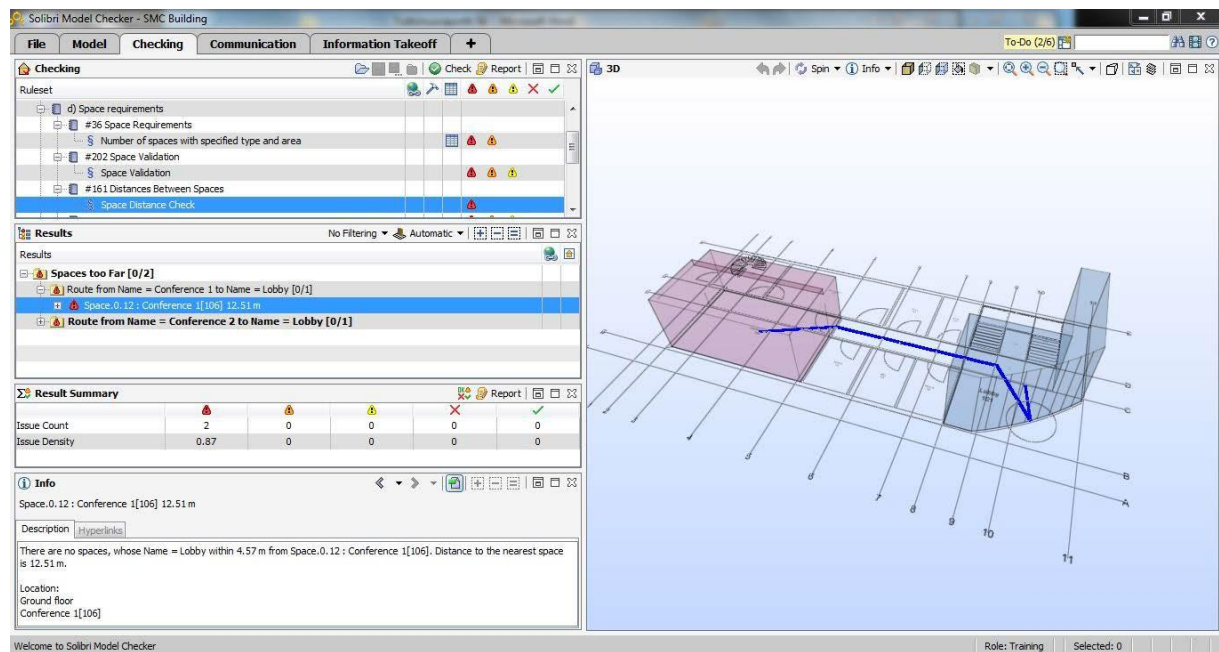


Figure 11. Checking distance between the two spaces in Solibri Model Checker (Bolpagni, 2013)

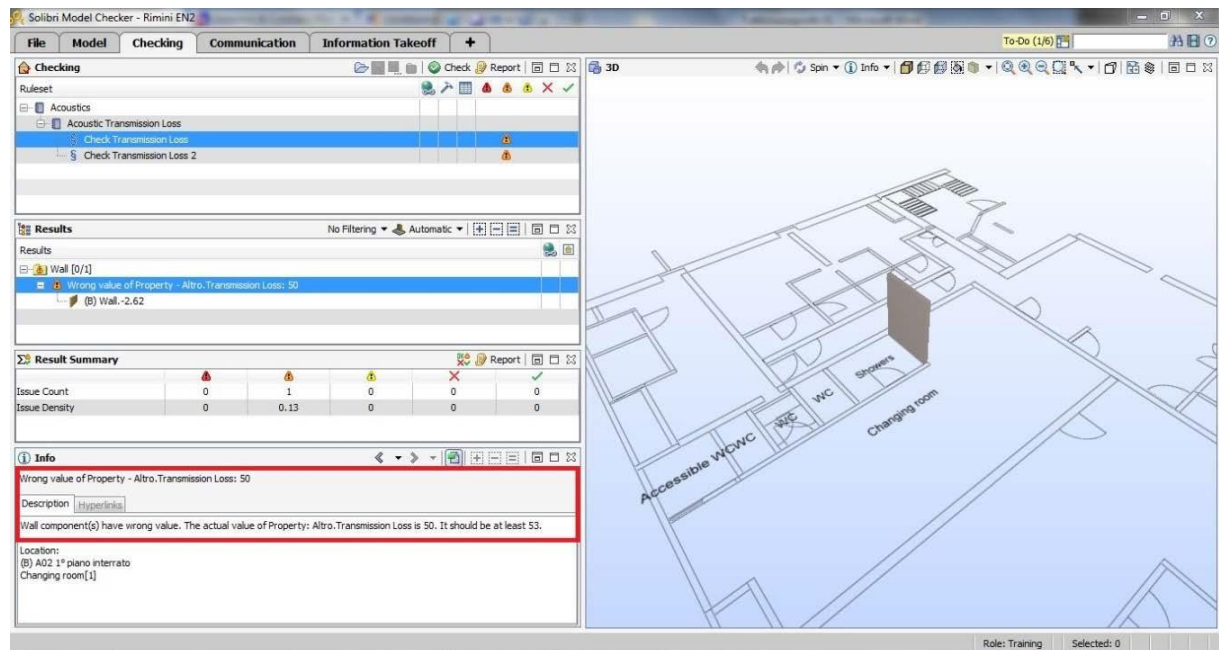


Figure 12. Checking the value of sound transmission loss of the walls in Solibri Model Checker (Bolpagni, 2013)

3 PROCESS OF CODE CHECKING

3.1. FOUR-STEP ALGORITHM

This chapter briefly describes the sequence of the code checking process. One of the most appropriate algorithms to describe this process is presented in (Eastman et al., 2009).

According to Eastman et al., 2009 the process of code checking mainly consists of 4 parts:

- 1) Rule interpretation;
- 2) Model preparation;
- 3) Rule execution;
- 4) Reporting.

The detailed schema containing the described steps is illustrated in Figure 13.

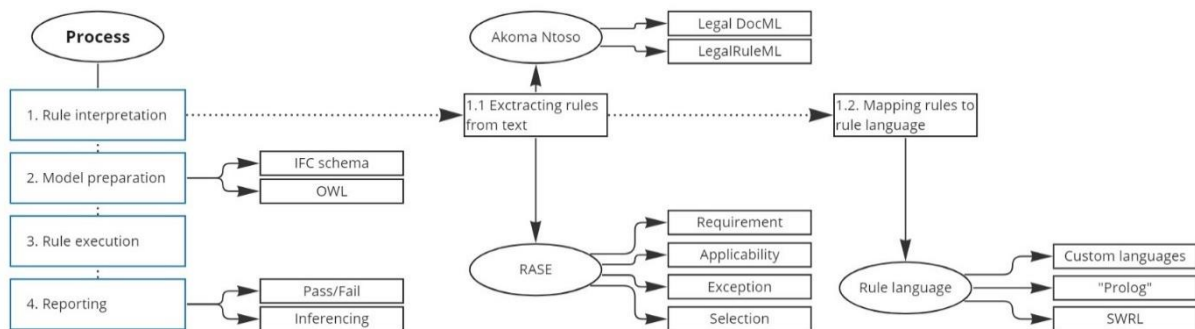


Figure 13. Process of code checking. Four-step algorithm

3.1.1. RULE INTERPRETATION

The first step is responsible for converting rules and requirements into a format that can be processed by software tools. Codes and guidelines are typically written in natural language and need to be translated into a machine-readable format for compliance checking (Di Giuda et al., 2021; Eastman et al., 2009; Ismail et al., 2017; Rehman et al., 2021).

In the initial phase, rules need to be extracted from the text and then they can be transferred to a defined rule language.

Extracting rules from text

First of all, rule content should be extracted from the rule source such as regulations, standards, guidelines or other documents.

One of the most representative examples is Akoma Ntoso Project, where open standard projects such as LegalDocML and LegalRuleML were applied to extract the specific rule content from the rule document (Schwabe et al., 2019). LegalDocML focuses on the syntax and structure of the document, whereas Legal-RuleML focuses on the semantics and logic of the rule. The structured rule content can be translated or transformed into different rule languages for further processing.

Other methods like RASE (Requirement, Applicability, Selection, and Exception) markup can also be used to classify information in the rule document for further processing (Hjelseth & Nisbet, 2011).

Mapping rules to rule language

After identifying the rule content, it is necessary to convert it into a language that can be understood and utilized by a rule engine (Sydora & Stroulia, 2019). Different rule engines may require different rule languages, such as Drools or Prolog. Semantic Web technologies like SWRL (Semantic Web Rule Language) can also be used for reasoning about ontology-based knowledge representations (Shen et al., 2022; Zhou et al., 2022). The XML (Extensible Markup Language) based language mvdXML was developed to define MVDs (Model View Definitions) with IFC (Industry Foundation Classes) and can be used for code checking (buildingSMART International). The implementation of mvdXML model checking is described in details in Weise et al., 2016.

It is worth noting that there is also a technology that is still under development – Information Delivery Specification (IDS), which has the potential to be a valuable tool for checking BIM models. IDS can be used for checking IFC files and served as a crucial component for correct delivery of information. This approach enables project-specific and use-case-specific requirements to be created for projects and asset portfolios, making data exchange work-flows more predictable and reliable (Berlo et al.; buildingSMART International).

3.1.2. MODEL PREPARATION

The building model data needs to be prepared in a way that is compatible with the code checking process (Andrich et al., 2022). This may involve converting the BIM model into a format that can be analyzed by the software tools, such as IFC, which is a standard file format for BIM data exchange (Laakso & Kiviniemi, 2012).

The building model data, usually represented using the IFC schema, needs to be transformed into a knowledge representation suitable for processing by the rule engine (Bazjanac, 2008; Succar, 2009). For example, if SWRL is used as the rule language, the knowledge needs to be represented in an ontology that uses the OWL (Ontology Web Language). Projects like ifcOWL have been developed for this purpose, but they are still in the early stages of research (Li et al., 2021). The concept of semantic enrichment, as introduced by Belsky et al., 2016, can also be used to add missing facts to the model subsequently.

To sum up, properly preparing the model data is crucial to ensure accurate and reliable code compliance checking results.

3.1.3. RULE EXECUTION

Once the rules are translated into a machine-readable format and the building model data is prepared, the code compliance checking process needs to be executed. This involves running the automated software tools to analyze the BIM model and compare it against the translated rules. The rule execution stage may involve complex algorithms and logic to accurately assess compliance with the applicable codes and regulations.

3.1.4. REPORTING

Once the rule checking process is completed, the outcomes are analyzed and reported to the end user, which can entail generating new knowledge, such as pass/fail results of the rules or additional semantic knowledge inferred by the rule engine (Eastman et al., 2009).

This inferencing process is a notable advantage of using rule engines instead of hard-coded approaches as it enables the system to deduce further insights. Additionally, the results of the code checking process must be conveyed in a meaningful manner through reports that highlight the compliance status of the BIM model, searching for any violations or issues, and offering recommendations for resolution. These reports should be user-friendly and easily understandable by stakeholders, including architects, engineers, and other members of the project team. It means that results of the checking process need to be reported in a meaningful way so that the responsible person can understand the detected problems and initiate appropriate actions for resolution (Schwabe et al., 2019).

3.2. COMPLEXITY AND EDITING

In this work, in addition to the main focus on making a classification based on possible checking scenarios, two important schemes regarding complexity of rules and editing of BMC are presented.

3.2.1. COMPLEXITY OF RULES

The Figure 14 depicts the different classes of rule complexity regarding BIM-based model checking methods.

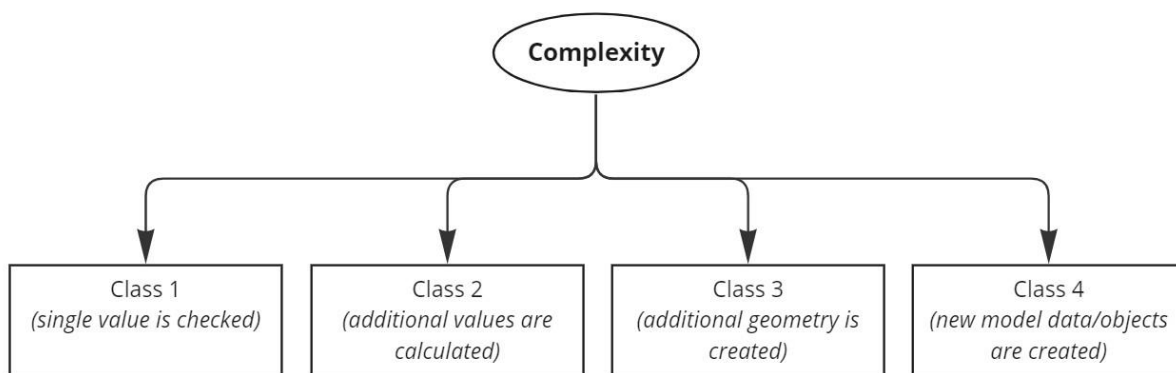


Figure 14. Different classes of complexity

According to Solihin & Eastman, 2015 there are 4 classes of rule complexity.

Class 1 rules require only a small amount of explicit data and involve checking attributes and entity references in the BIM dataset. These rules are relatively simple and often involve checking whether certain attributes are set correctly, such as the fire rating of a wall or the type of door required by building code.

Class 2 rules involve checking derived attribute values, which are based on a single value or a small set of values. These rules require the program to derive the data from the BIM model data and relationships, and may involve arithmetic or trigonometric calculations. Checking these rules may require implicit relationships, which are not explicitly captured in the BIM model.

Class 3 rules require an extension to the data structure and involve checking complex requirements, such as building code checking. These rules often require sophisticated 3D solid modeling libraries to perform complex geometric and spatial operations, and may involve derivation of topological graphs and the use of algorithms like shortest path.

Class 4 rules require a proof of solution and emphasize how the building model demonstrates compliance with prescribed criteria. These rules are often performance-based codes or similar rules that can have more than one acceptable answer. A knowledge-based facility is typically coded into the system to provide solutions, which are captured

and continuously updated as new knowledge arises. Checking these rules may involve temporarily and virtually inserting additional model data into the existing BIM model.

3.2.2. EDITING

The concept of editing can be broadly categorized into two main approaches: Black-Box and White-Box methods (Figure 15).

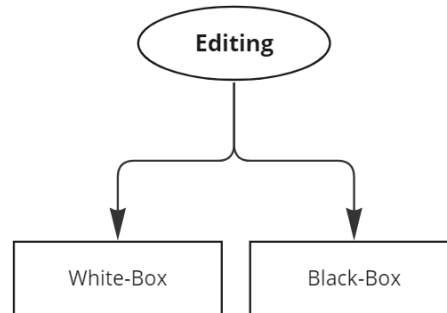


Figure 15. White-box and black-box approaches of editing

The Black-Box method involves defining hard-coded algorithms that execute the checking process without revealing the internal processing steps to the user. This results in a covert process where the user lacks visibility into the processing procedure, and modifications or extensions can only be made by the software vendor. Due to the closed nature and inaccessibility to the underlying data structures of the code checking system, this approach has the benefit of having a relatively low error rate.

The White-Box method, on the other hand, seeks to make the user aware of and understand the internal processing operations. This approach requires translating the rules based on a code representation system or language that allows for the description of objects, methods, and relationships. The user must be able to understand and track information at any moment as well as follow the progress of the checking operation. Therefore, the rules must be readable not only by the machine, but also by the user. Although this approach requires more effort in development and implementation compared to the Black-Box method, it offers major advantages in terms of transparency and user understanding.

This classification is described in more detail in the following papers: Di Giuda et al., 2022; Preidel & Borrmann, 2018.









4 MODEL CHECKING TOOLS

BIM-based model checking software is a relatively new technology that has emerged alongside the development of BIM in general, and openBIM using the IFC-format in particular. As a result, there are currently a growing number of software developers offering BMC software to the market. This section presents an overview of prevalent software and their respective functionalities.

Before proceeding to the core content of this section, which outlines the key features of model checking tools, it is important to highlight that such tools can be divided into two main categories based on their deployment options: cloud-based and local software.

Table 3 provides a list of several popular BIM-based model checking software, along with links to their official websites.

Table 3. List of software for BMC

No.	Software		Official website
1	Solibri Model Checker		https://www.solibri.com/
2	Autodesk Navisworks Manage		https://www.autodesk.com/products/navisworks/overview
3	Autodesk BIM Collaborate Pro		https://www.autodesk.com/products/bim-collaborate/overview
4	Bentley Projectwise Navigator		https://www.bentley.com/software/projectwise/
5	Trimble Connect (Tekla BIMsight)		https://connect.trimble.com/
6	dRofus – Nosyko		https://www.drofus.com/
7	BIMcollab Zoom		https://www.bimcollab.com/en
8	Verifi3d		https://verifi3d.xinaps.com/

Cloud-based software (e.g., Autodesk BIM Collaborate Pro, Trimble Connect, Verifi3D) refers to software that is hosted and run on remote servers, allowing users to access it through the internet. Local software (e.g., Solibri Model Checker, Autodesk Navisworks Manage), on the other hand, refers to software that is installed and run on a user local

computer or device. Both types of software have their benefits and drawbacks, and the choice often depends on the specific needs of the user. In order to combine the advantages of both categories some software in addition to local version also offers a cloud-based tool (e.g., BIMcollab Zoom and BIMcollab Cloud, dRofus and dRofus Web).

Table 4 gives the representation of the features and capabilities of the BIM-based model checking software. The Table 4 provides information on the various model checking methods supported by each software, as well as the exchange formats each software can handle, and their cost.

Based on the analysis of the Table 4, it can be concluded that Solibri Model Checker is the most comprehensive model checking software, supporting all three model checking methods. It also offers flexible licensing options, including monthly and annual licenses.

Autodesk Navisworks Manage and Autodesk BIM Collaborate Pro are also excellent choices, with Navisworks being particularly strong in clash detection, while BIM Collaborate Pro offers seamless collaboration and communication features. Despite being the most expensive option, the benefits of Autodesk Navisworks Manage are notable, with the software being highly stable and capable of supporting a wide range of file formats regarding BIM.

Bentley Projectwise Navigator is a good option for larger projects, offering robust project management capabilities in addition to clash detection features. Trimble Connect (Tekla BIMsight) is a cost-effective option that supports clash detection and basic code checking, making it a good choice for small to medium-sized projects.









dRofus – Nosyko is a specialized software solution that provides advanced capabilities for checking model content, rule compliance, and data management. Additionally, this software is available at no cost, making it an attractive option for organizations with budget constraints.

BIMcollab Zoom besides the availability of all three model checking methods offers easy and effective collaboration. This makes it a versatile tool that can be used for a wide range of purposes and particularly attractive for teams that need to work closely together to resolve issues in their building models.

Verifi3D is a dedicated clash detection software, which has a direct connection to Autodesk Revit. Verifi3D does not provide information on the cost of a license upfront, as it requires consultation with a representative.

To sum up, the choice of software depends on the specific needs of the user, such as project size, model complexity, and budget. By carefully evaluating the features and capabilities of each software, users can select the tool that best suits their needs, enabling them to create high-quality building models that meet all necessary codes and regulations.

Table 4. Analysis of model checking tools

			Solibri Model Checker	Autodesk Navisworks Manage	Autodesk BIM 360 Model Coordination	Bentley Projectwise Navigator	Trimble Connect (Tekla BIMsight)	dRofus – Nosyko	BIMcollab Zoom	Verifi3d
										
Model checking	Clash detection		✓	✓	✓	✓	✓		✓	✓
	Content checking		✓					✓	✓	
	Code checking	Modifying existing rules	✓	✓			✓	✓	✓	
		Separate rulelets	✓					✓	✓	
		Combining rules to new rulesets	✓					✓	✓	
		Developing new rulesets	✓					✓	✓	
Exchange formats	Direct link to Revit			✓	✓			✓	✓	✓
	Direct link to ArchiCAD		✓						✓	
	Import of BIM file		ifc, dwg	all	ifc, dwg, rvt, nwc	dgn, dwg, pdf	ifc	ifc	ifc, bcp, e57, pts	ifc, rvt
	Export of report		pdf, xls	pdf, xls, html, xml		pdf			pdf, xls, bcf	bcf
Cost	Individual monthly license, €		280	387	155			free	-	negotiable
	Individual annual license, €		1870	3112	1226	901	120		720	

5 DEMONSTRATION OF MODEL CHECKING METHODS

The final part of this work is devoted to a more practical aspect of the current topic – demonstration of the BIM-based model checking methods.

Specifically, this chapter considers two previously described methods: clash detection and model content checking. To illustrate the process of BIM-based model checking, one of the most functional model checking tools is applied – Solibri Model Checker. As a BIM model, the part of the Beyer-Bau model is used.

5.1. MODEL FOR DEMONSTRATION

5.1.1. PREPARATION OF MODEL IN AUTODESK REVIT

As it was mentioned earlier, in order to perform the model checking, the model must be prepared in an appropriate way. The demonstration model utilized in this study is a pre-designed BIM model of the Beyer-Bau building, which was developed in the Autodesk Revit software (Figure 16). The floor area of this object is almost 18000 m². In this regard, it was decided to use only part of the model in order to make the demonstration of methods more visual and clearer (Figure 17). This part of the model is located in the west wing of the building. The extracted part consists of two floors with identical room layouts on each.

According to the plan of the floor (Figure 18), one of the rooms is a WC room, where elements of engineering systems are located: ventilation ductworks, pipes of water supply and sewerage systems. Creating and adding these elements to the BIM model will help to simulate a scenario that frequently occurs in practice: the intersection of MEP (Mechanical, Electrical, and Plumbing) elements with structural elements.

Therefore, the next elements were created for further demonstration: ventilation ductworks, pipes for cold and hot water supply, sinks (Figure 19, Figure 20). It is important to note the following two points in the modeling process:

- 1) Openings for MEP elements in structural elements are deliberately not created in order to simulate a collision situation in the next steps;
- 2) To make the execution of verification process possible, the structural and MEP building elements are modeled in separate files and can be linked to each other if necessary (Figure 21).

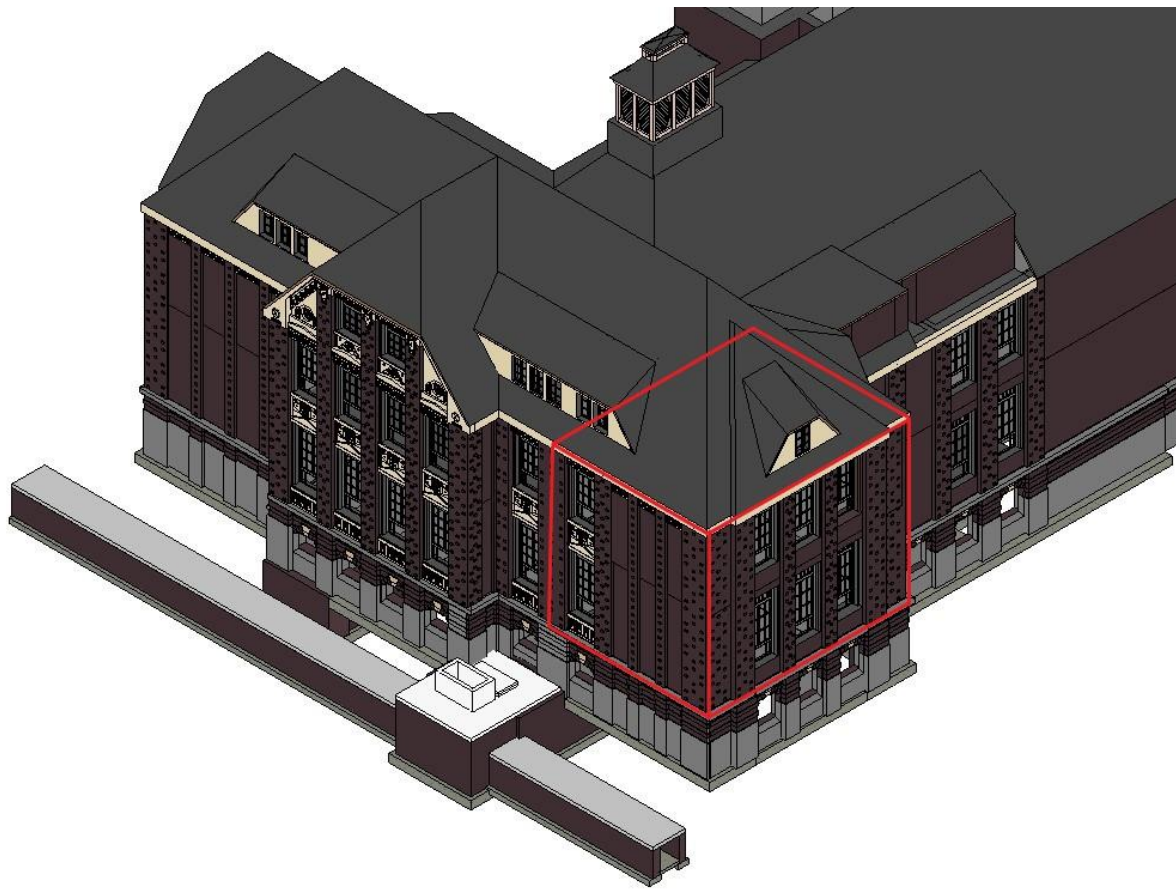


Figure 16. Beyer-Bau BIM model in Autodesk Revit

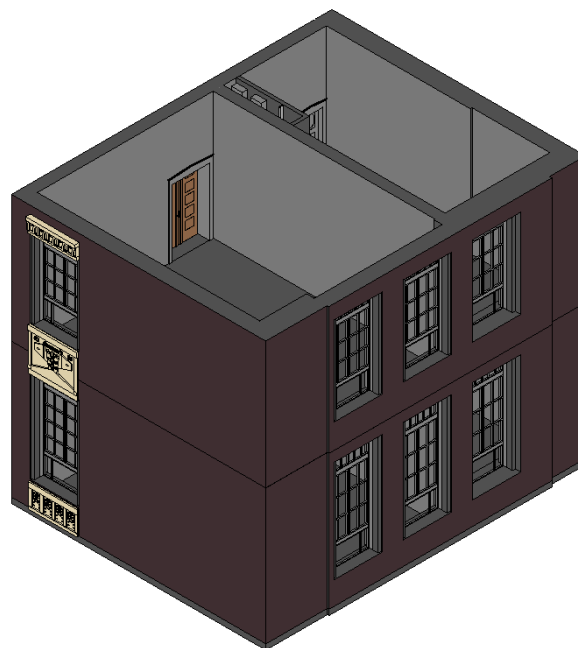


Figure 17. Small-scale part of the Beyer-Bau BIM model in Autodesk Revit

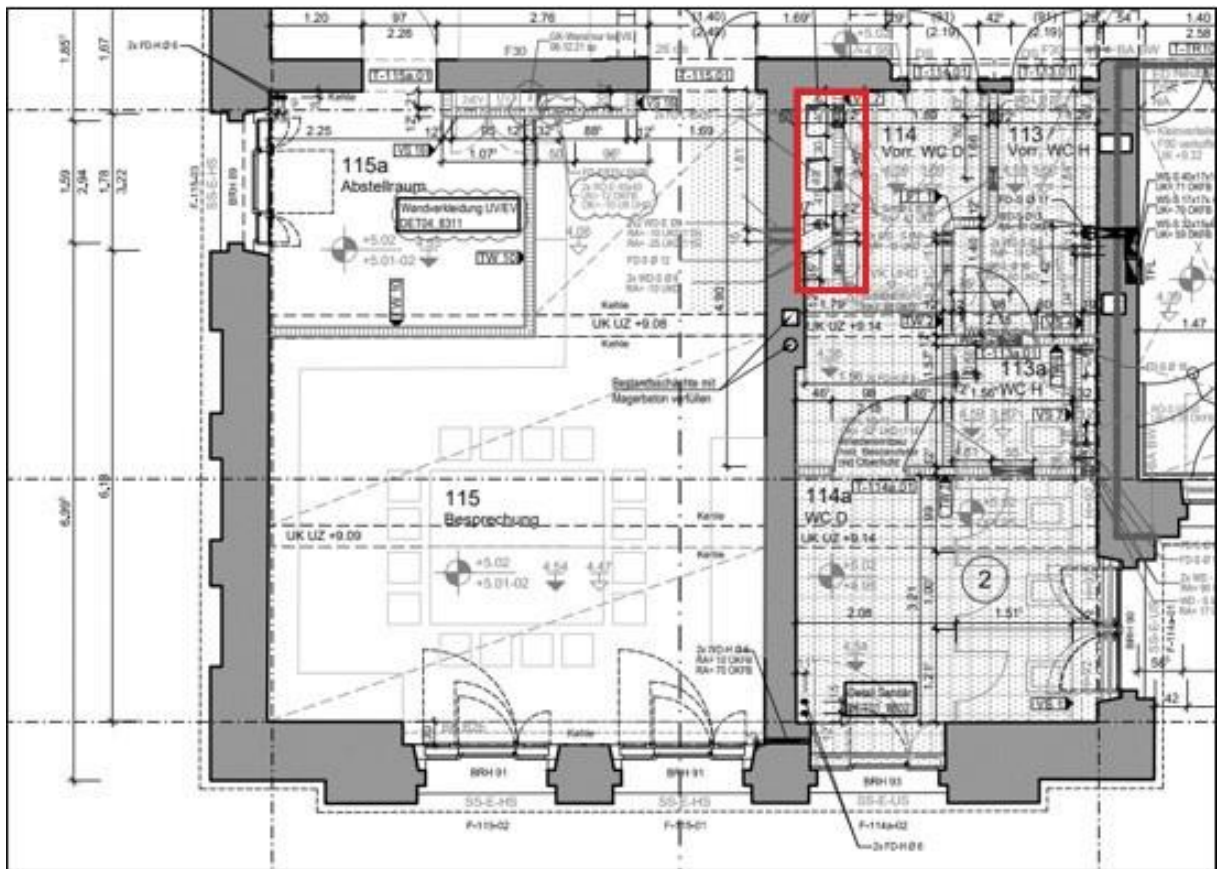


Figure 18. Plan of the 2nd floor of the Beyer-Bau building

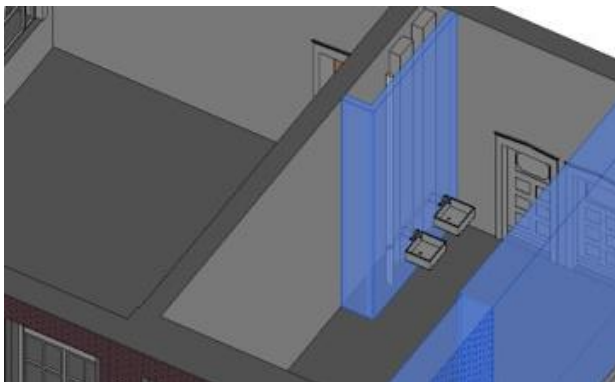


Figure 19. WC room with MEP elements

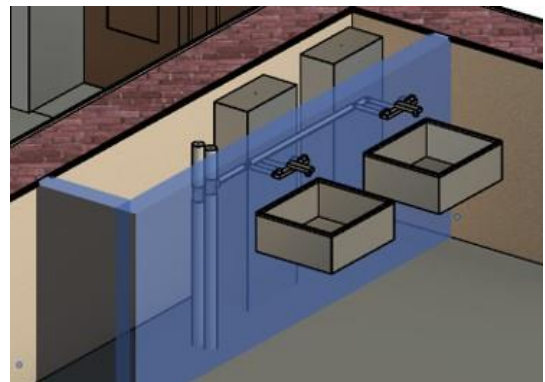


Figure 20. MEP elements

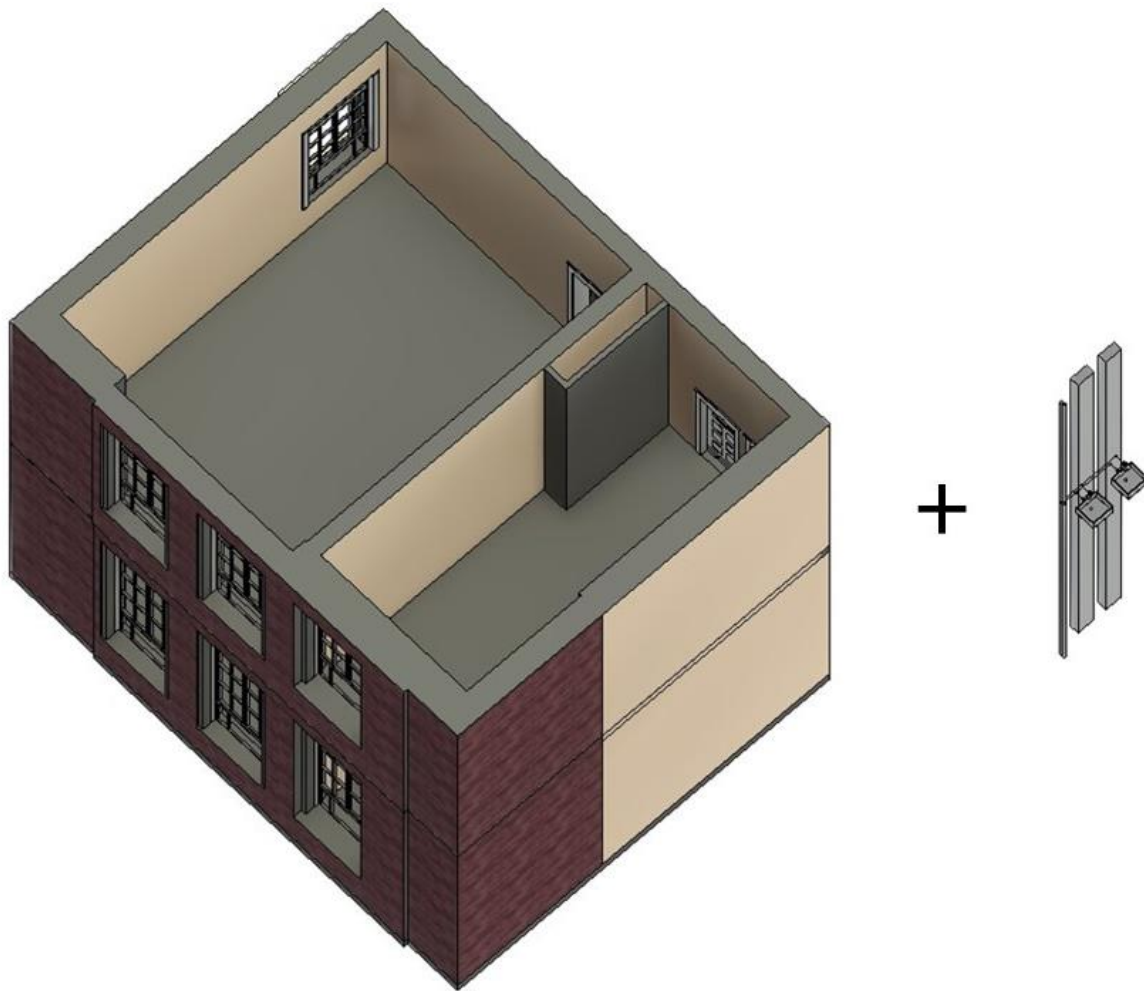


Figure 21. Visual representation of two different models in separate files

Table 5 summarizes the information on the model checking methods demonstrated in subsequent sections. For clash detection, one of the most commonly encountered scenarios – the case with hard clashes – is chosen. For model content verification, a model completion checking scenario is used.

Table 5. Model checking methods for demonstration

No.	Model checking method	Checking scenario	Software	BIM Model
1	Clash detection	Hard clashes	Solibri Model Checker	Beyer-Bau
2	Model content checking	Model completion checking		

5.1.2. EXPORT MODEL TO IFC

After completing the modeling process and preparing the corresponding files of the BIM model, the next step is to convert the model to the format required for model checking.

In this case, the model is planned to be checked in Solibri Model Checker, which has no direct connection with the Autodesk Revit environment (Table 4). Therefore, files with prepared models must be transformed from ".rvt" format to ".ifc" format. This feature of exporting files to the IFC environment is built into Autodesk Revit 2022 and therefore does not require any additional actions regarding using additional software or plugins, as it was earlier. However, at this stage it is crucial to make sure that the export process itself and its settings are correct.

One way to make sure that IFC file contains accurate and complete information before using it in model checking software is proper setting of IFC Options in Autodesk Revit (Figure 22). This function allows to assign IFC Class Name to every Revit Category (Figure 23). For instance, interior and exterior walls ("Walls/Interior" and "Walls/Exterior" Revit Categories) are mapped to "IfcWall", while all types of windows ("Windows" Revit Category) are mapped to "IfcWindow". Some of the Revit Categories, which are not exported to IFC have the "Not Exported" label in IFC Class Name column.

After setting up the IFC Export Classes and confirming that all the required object Categories in Revit are assigned the appropriate IFC Class Names, the important step is configuring the export process itself. After selecting the "Export IFC" command in Autodesk Revit (Figure 24), a configuration window is displayed (Figure 25), presenting several options to be customized with. The main parameter to be modified is "Current Selected Setup", where a range of pre-configured setups are available for selection, including such MVDs (Model View Definitions) as "IFC 2x3 Coordination View 2.0", "IFC 4 Reference View", "IFC 4 Design Transfer View", and others (Figure 25).

By selecting a specific MVD, the user can define which aspects of the BIM model should be included and how they should be structured. Selecting the appropriate MVD is an important step in ensuring that the BIM data is correctly defined and can be used effectively throughout the project. The basic and advanced settings for each of the pre-defined setup can be viewed in the "Modify Setup" window. There is also an option to create custom setups by selecting "In-Session Setup" and adjusting the available settings (Figure 26, Figure 27). This level of customization is valuable because it allows users to receive data according to the specific use case, ensuring that it performs optimally for their needs.

5 Demonstration of model checking methods

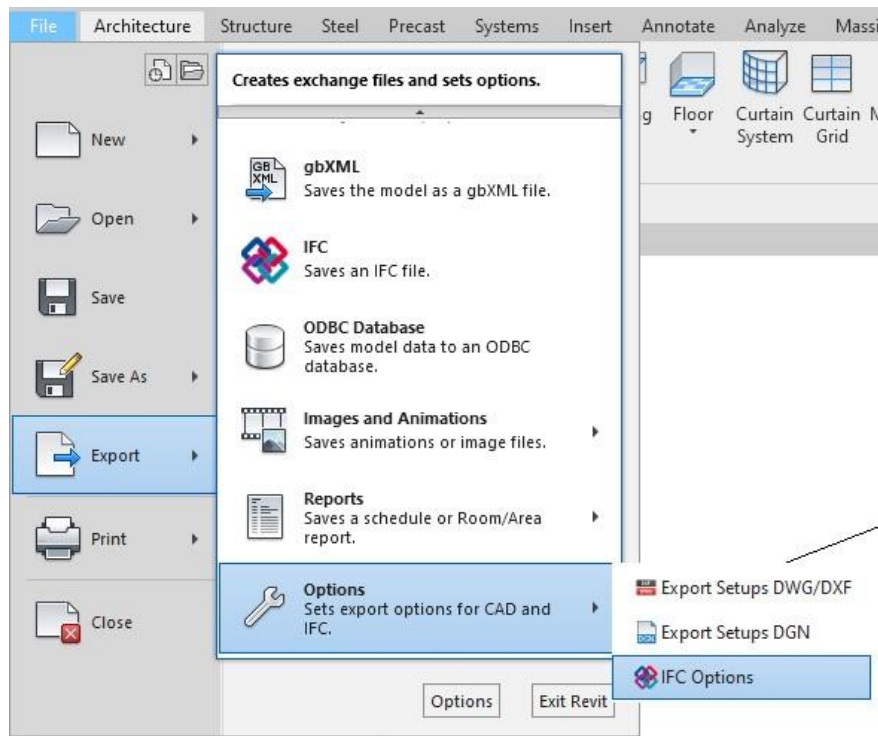


Figure 22. IFC Options in Autodesk Revit

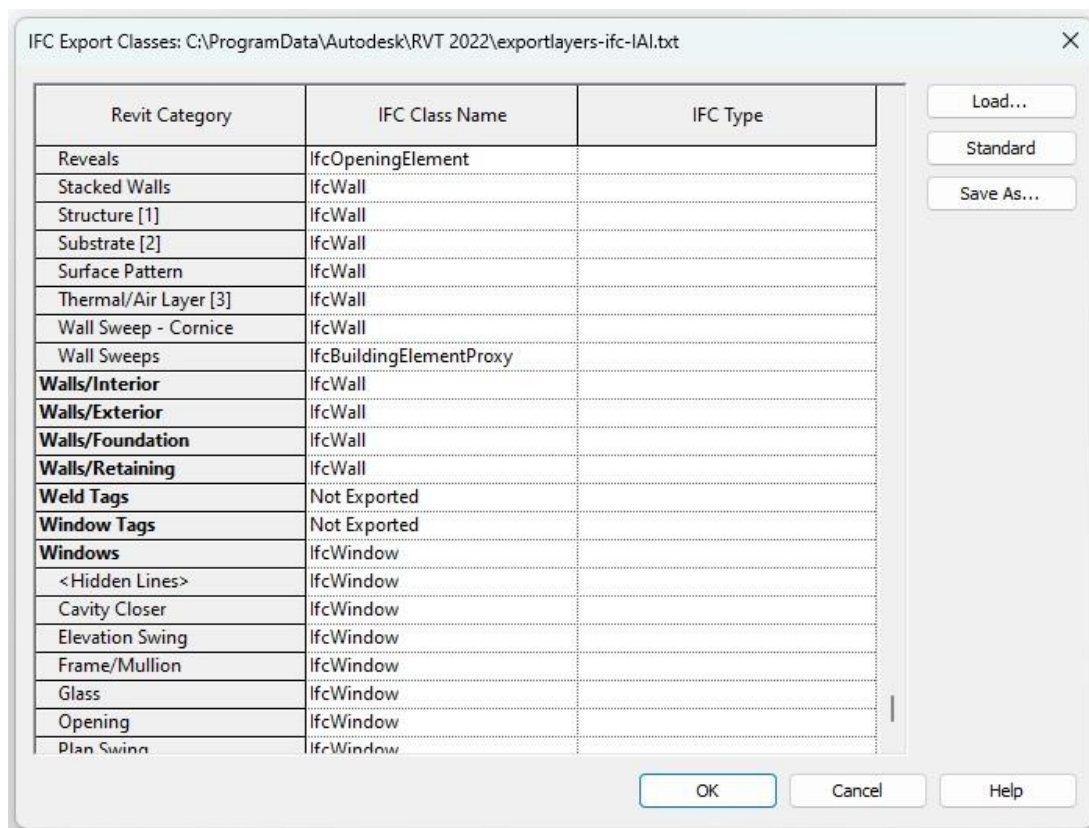


Figure 23. Setting IFC Export Classes in Autodesk Revit

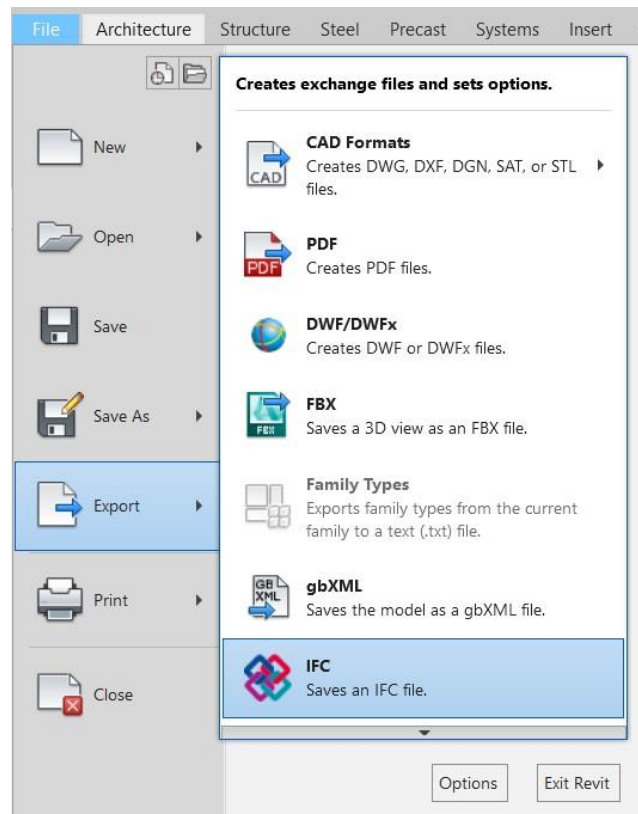


Figure 24. Export to IFC in Autodesk Revit

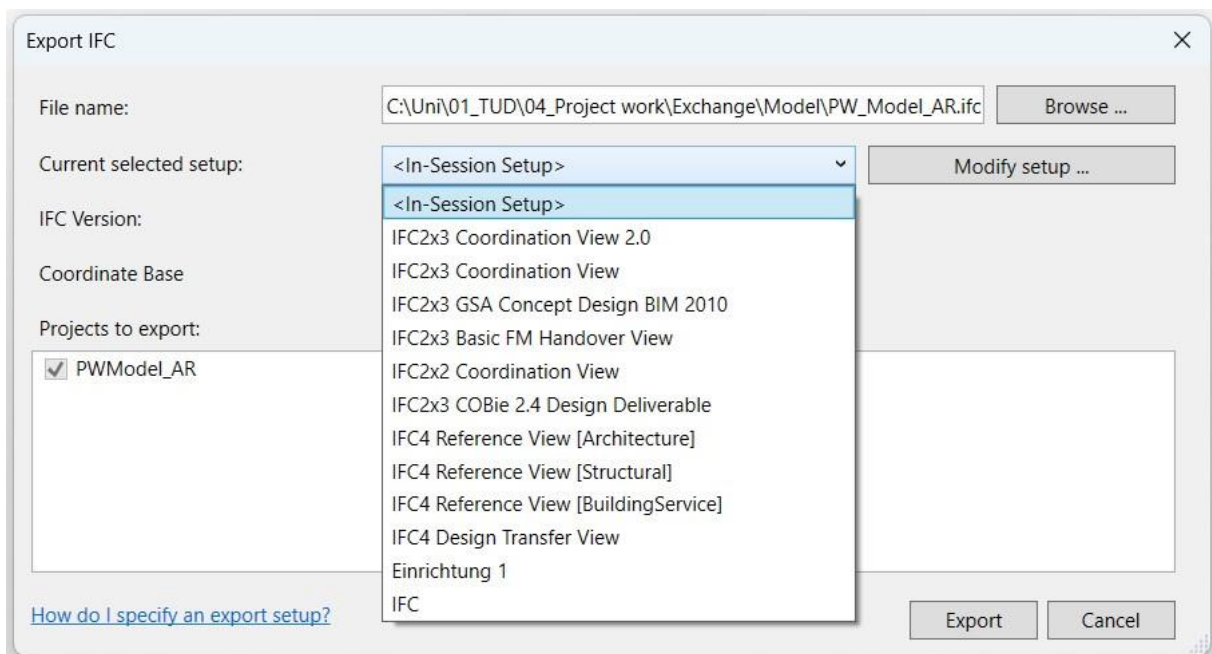


Figure 25. IFC export settings in Autodesk Revit

5 Demonstration of model checking methods

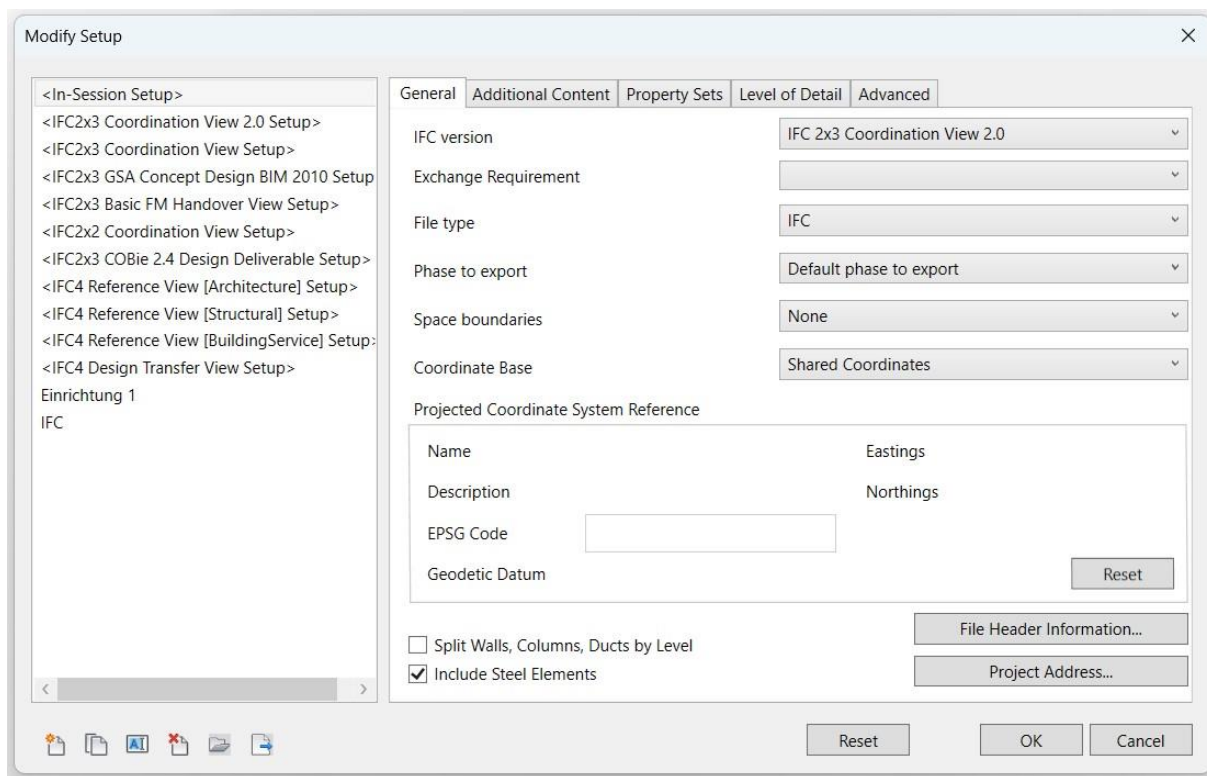


Figure 26. Modifying setup of IFC export in Autodesk Revit. General settings

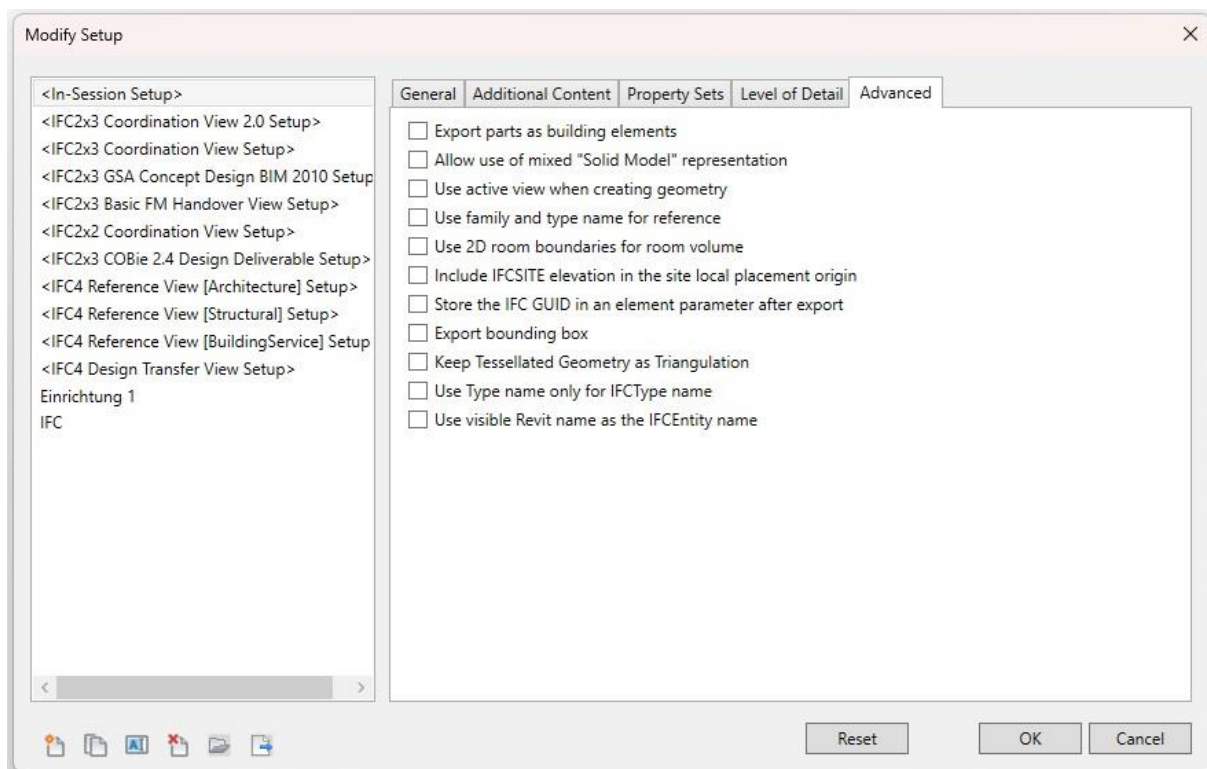


Figure 27. Modifying setup of IFC export in Autodesk Revit. Advanced settings

Finally, the last step in verifying the correctness of the exported file is to visually check the BIM model after opening the IFC file in the model checking software. After importing the IFC file into Solibri Model Checker (the import process is described in detail in the next chapter), first of all, it is possible to visually roughly evaluate the BIM model for the presence of main elements, and secondly, there is a way to check the composition of the model using "Model Tree" (Figure 28). Solibri Model Checker automatically organizes elements into groups depending on the levels (ground floor, first floor, second floor, etc.) and element type (doors, windows, walls, etc.). When an element is selected in the "Model Tree", it is highlighted in the 3D view of the model, while information about it and its properties are displayed in the "Info" window (Figure 28).

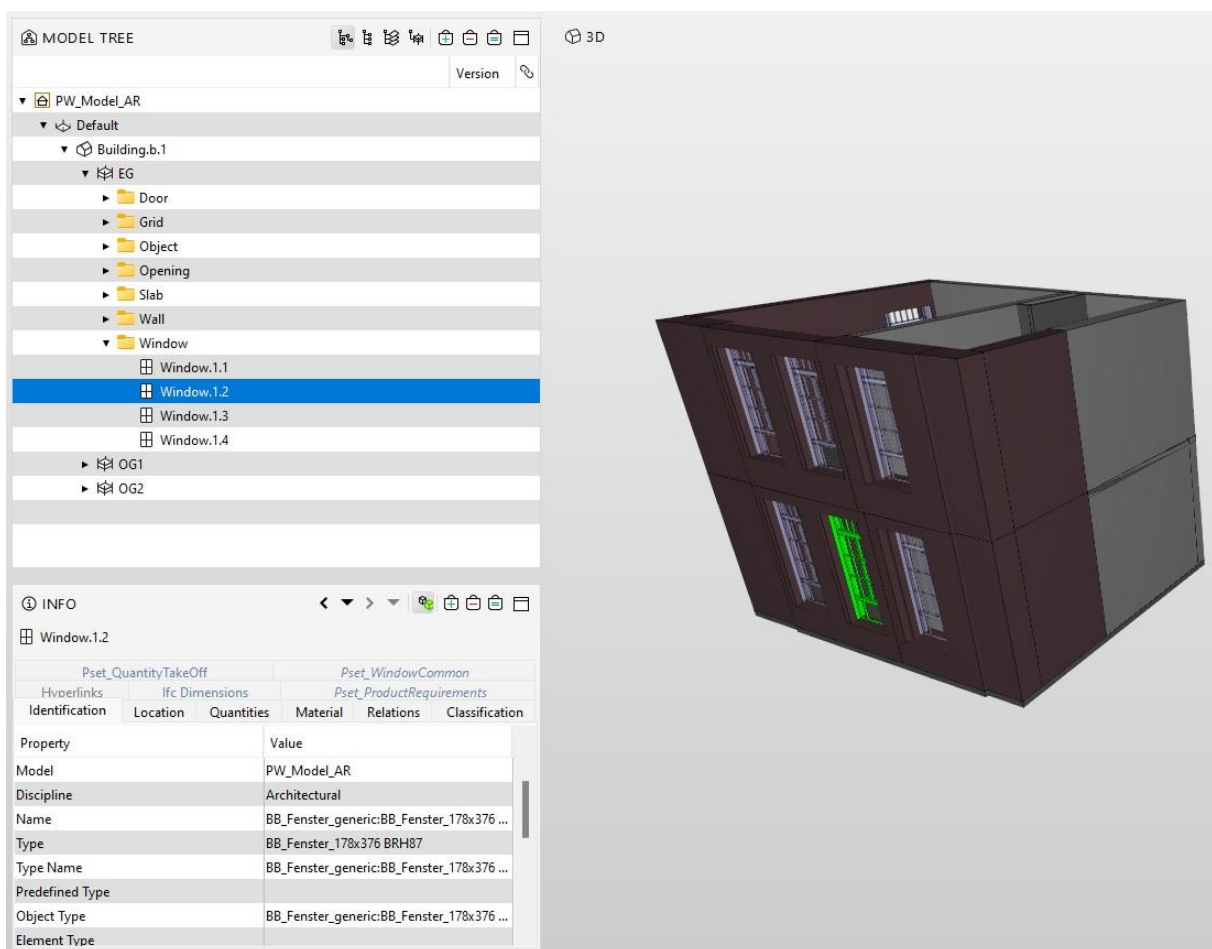


Figure 28. Visual check of the model after import in Solibri Model Checker

5.2. CLASH DETECTION. DEMONSTRATION OF METHOD

This subsection describes the main steps of the model checking process performed in Solibri Model Checker.

After the successful export of files (in this case, two files: with structural and architectural elements; with MEP elements), it becomes possible to upload and open them in Solibri Model Checker. Further, these files become visible in the "Model Tree" window (Figure 29), with the corresponding disciplines that were previously assigned in SMC, as shown in the Figure 30.



Figure 29. "Model Tree" window in the Solibri Model Checker interface

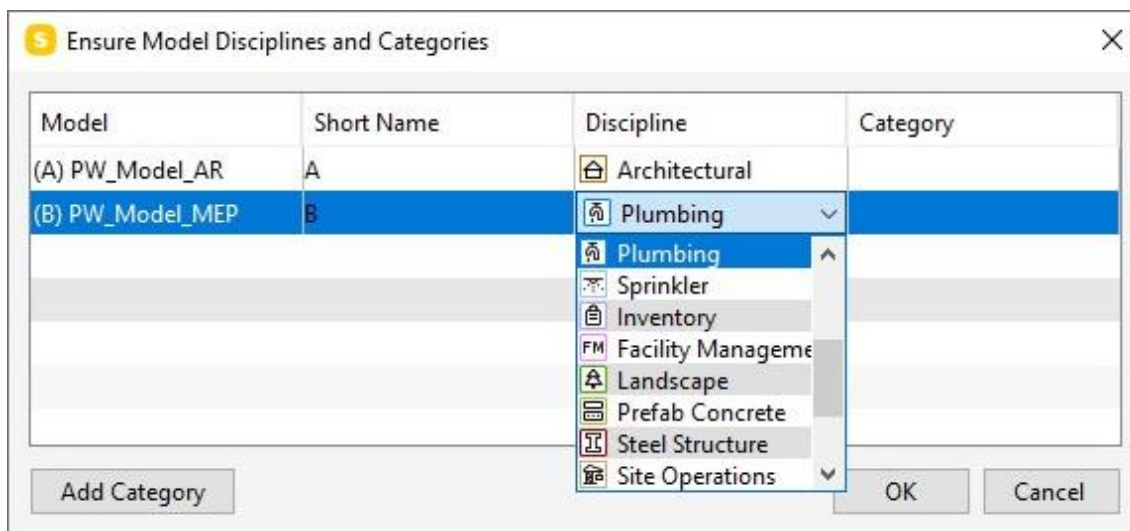


Figure 30. Choosing a discipline for uploaded files in Solibri Model Checker

The next important step is adding rules or rulesets that will be used to check the BIM model. When loading rules, Solibri Model Checker offers to select a role that determines the set of rules that will be offered as the default option (Figure 31).

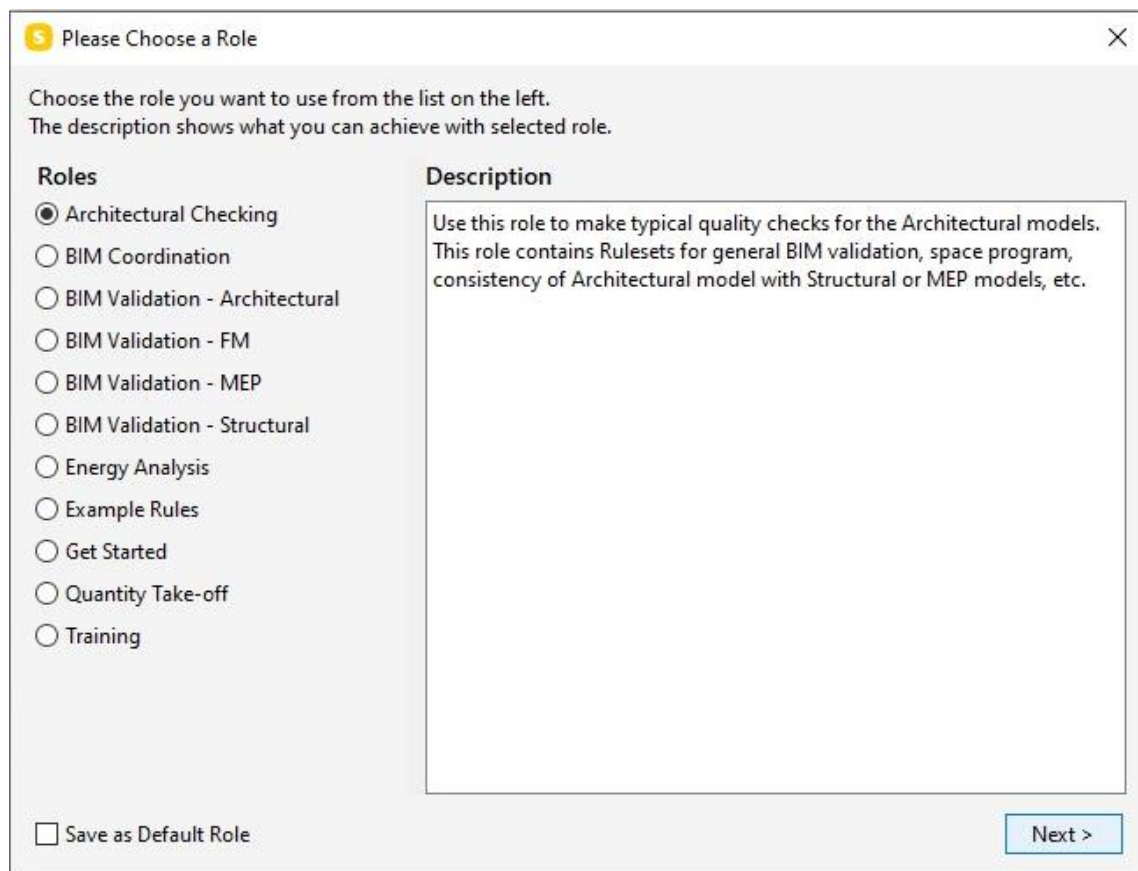


Figure 31. Choosing a role in Solibri Model Checker

However, further it is possible to load files with rules from a directory. In this case, according to the initially defined tasks, the most appropriate set of rules called "MEP models and Architectural model" was uploaded (Figure 32).

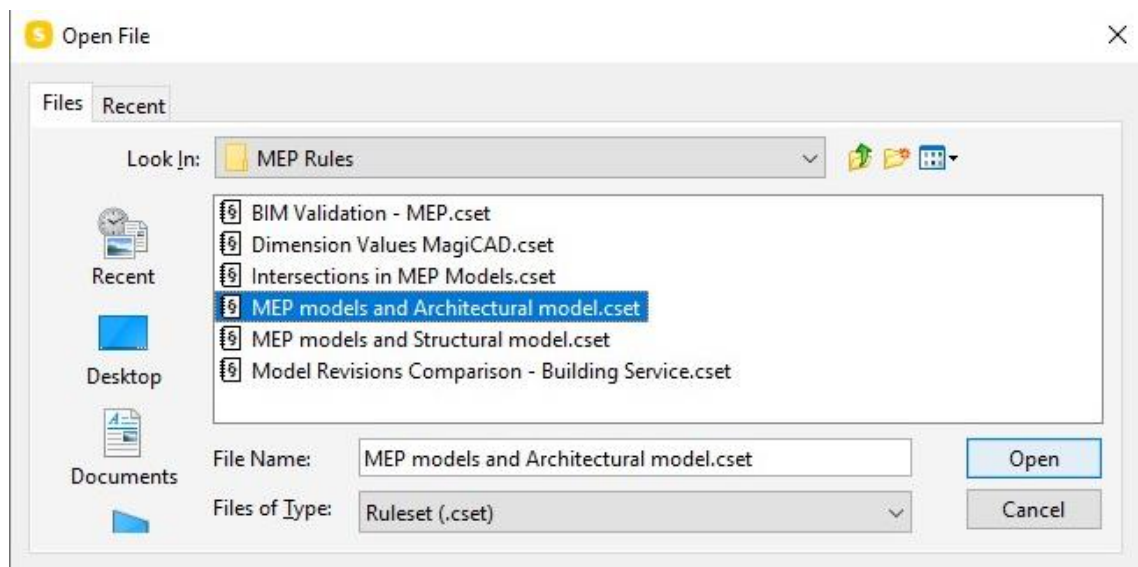


Figure 32 Uploading a ruleset from a directory in Solibri Model Checker

5 Demonstration of model checking methods

The Figure 33 shows one of the most significant windows in Solibri Model Checker - the "Checking" window. This window displays the rulesets and rules that were opened or loaded into the project. It is possible to change the parameters of the rules, as well as enable and disable them. By clicking the "Check Model" button, the checking process starts, and the results appear in the same window on the right side.

Moreover, the Figure 33 shows an additional "Info" window, which contains basic information about the ruleset or rule that is selected as active.

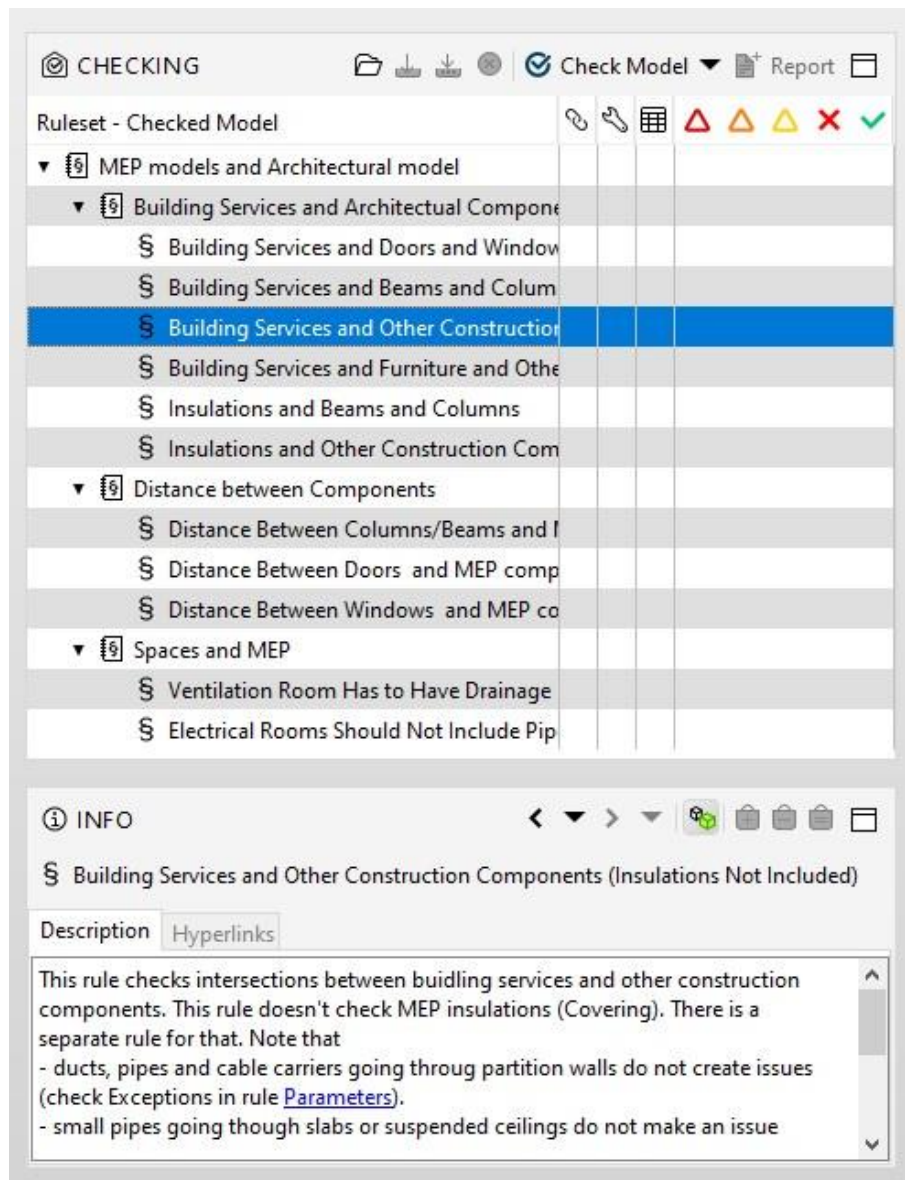


Figure 33 Information about selected ruleset or rule in Solibri Model Checker

The parameters of a rule in Solibri Model Checker can be edited in the "Parameter" window (Figure 34). These parameters can include different aspects. In the considered case of clash detection between "Building Services and other Construction Components" it is possible to adjust components that need to be checked. By modifying the parameters, users can adjust the rule to suit their specific project requirements and standards.

SMC allows for the adjustment of intersection tolerances (Figure 34), which is the degree to which two or more elements can overlap without being considered a clash or collision. This tolerance can be adjusted depending on the project requirements, the level of detail in the BIM model, and the precision of the design.

PARAMETERS

Severity Parameters

Components to Be Checked

Component 1

State	Component	Property	Operator	Value
Include	Any	Discipline	One Of	[Architectur...
Exclude	Furniture	Discipline	One Of	[Architectur...
Exclude	Light Fix...	Discipline	One Of	[Architectur...
Exclude	Object	Discipline	One Of	[Architectur...
Exclude	Flow Ter...	Discipline	One Of	[Architectur...
Exclude	Sanitary ...	Discipline	One Of	[Architectur...
Exclude	Space			

Component 2

State	Component	Property	Operator	Value
Include	Any	Discipline	One Of	[Air Condit...
Exclude	Covering			
Exclude	Space			
Exclude	Opening			
Exclude	Cable			
Exclude	Any	Decompose...	Is Not Empty	
Exclude	Any	MEP Comp...	One Of	[Flexible Pipe]

Include Intersections

☒ Duplicate

☒ Inside

☒ Overlapping

Intersection Tolerances

Horizontal: 10 mm

Vertical: 10 mm

Use Volume Tolerance: ☐

Volume Tolerance: 1 l

Ignore Intersections When Intersecting Components Are

☒ In the same system

☐ In the same layer and model

Figure 34. Parameters of the rule in Solibri Model Checker

5 Demonstration of model checking methods

The next step after completing the checking process is to analyze the results. Once the check is completed, the corresponding status appears next to each rule. The results can be accepted or rejected, and they can be assigned with the different levels of severity: low, moderate and critical.

The Figure 35 depicts the window "Results", where by selecting a rule, the results of the checking can be analyzed in detail.

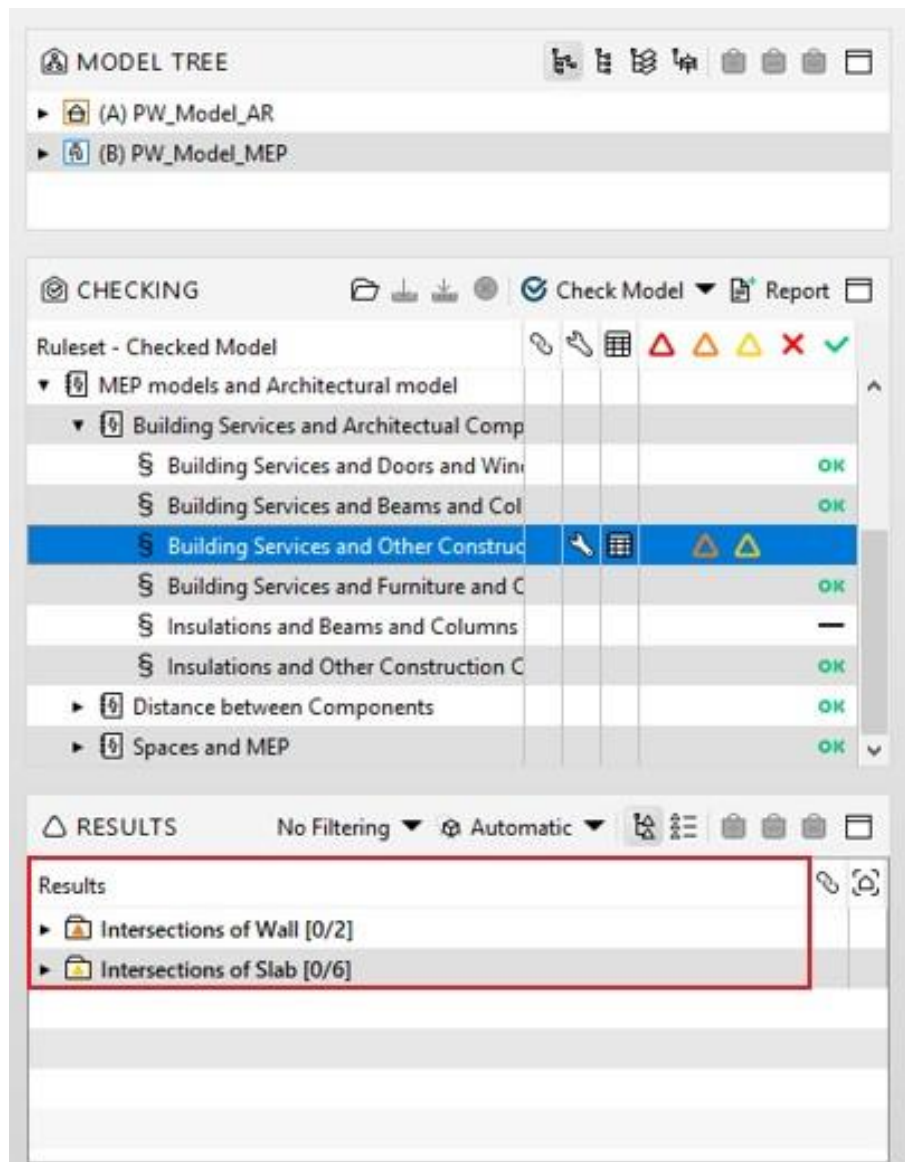


Figure 35. Results of clash detection checking in Solibri Model Checker

In the considered case – checking the model for collisions between architectural/structural elements and elements of MEP systems – 8 conflicts were identified in the model. These conflicts are presented in the "Results" window (Figure 35, Figure 36). It is shown that there are 2 intersections with a wall and 6 intersections with a slab.

In Solibri Model Checker, if two elements intersect each other multiple times, only one conflict will be shown in the "Results" window. In the considered example, the "Results" window shows 2 collisions between the wall and the pipes, despite the fact that there are 4 intersections (Figure 36). This happens because initially one pipe branches into two and crosses the wall in two places (Figure 20).

Therefore, the result looks like one element (pipe) has a collision with another element (wall), which is calculated as one collision, despite that in fact there are two of them. However, if there is a need to consider the conflict in detail, users can expand the tree of elements in the "Results" window and see all the places where different parts of one element (pipe) have intersections with another element (wall). This can provide more detailed information about the location and extent of the conflict, which can be useful in identifying and resolving the issue.

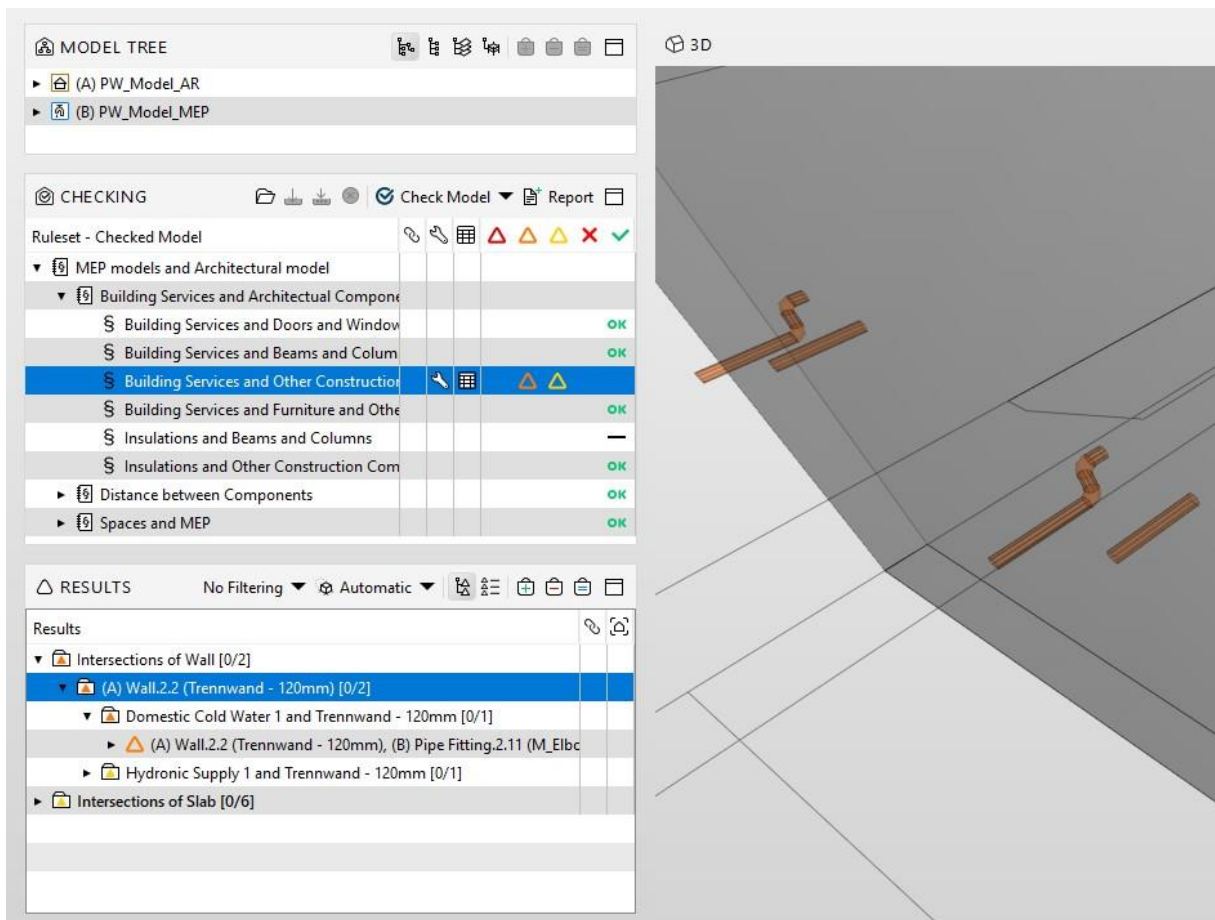


Figure 36. Grouping of clashes in Solibri Model Checker

Solibri Model Checker makes it possible to present the detected conflicts in a visualized form.

The Figure 37 demonstrates the first case – "Intersections of Wall". In this case there are collisions between the wall and two different pipes (this explains 2 conflicts that are depicted in "Results" window).

The Figure 38 shows the second case – "Intersections of Slab". There are collisions of the two pipes with the two floor slabs and collisions of two ductworks with one floor slabs (this explains 6 conflicts that are depicted in "Results" window).

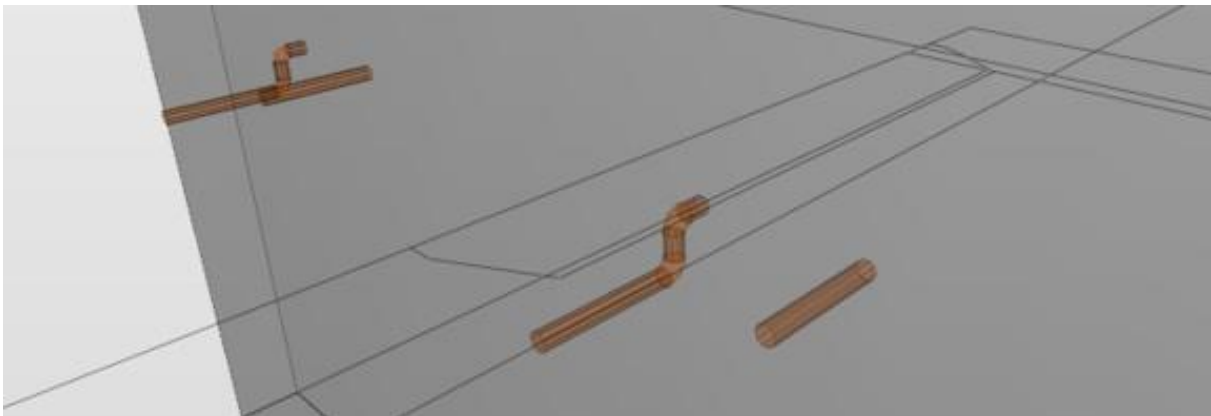


Figure 37. Collision between water pipes and wall in Solibri Model Checker

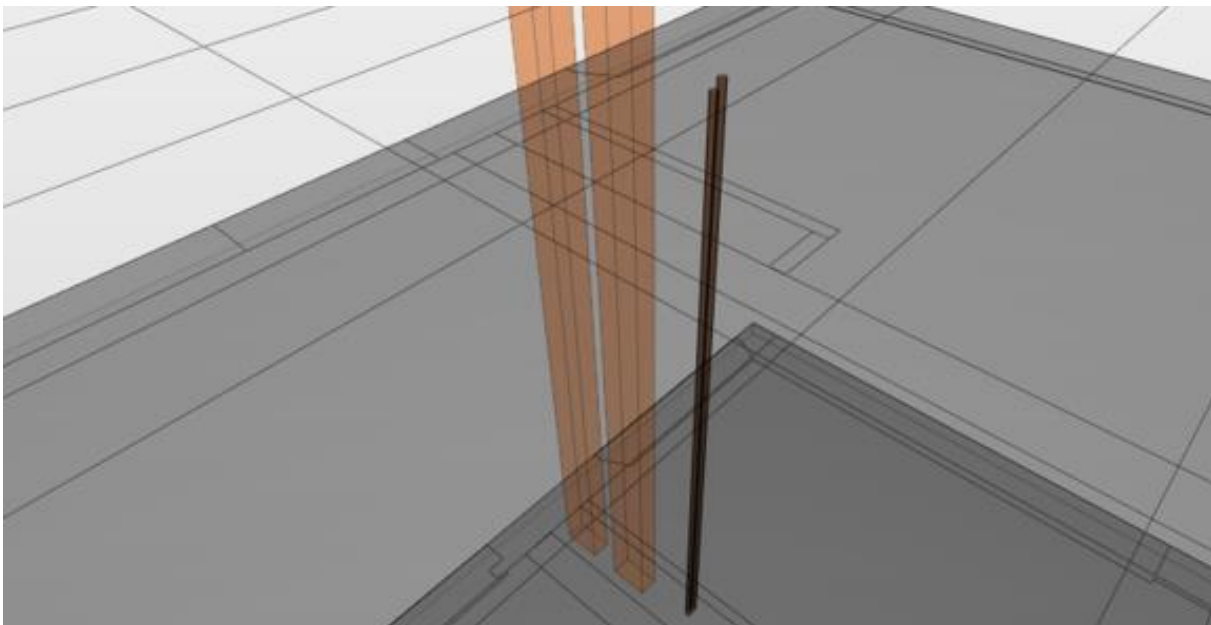


Figure 38. Collision between ductworks/pipes and floor slab in Solibri Model Checker

5.3. MODEL CONTENT CHECKING. DEMONSTRATION OF METHOD

This subsection briefly describes the model content checking method carried out in Solibri Model Checker. The sequence of steps in this case is exactly the same as described in the previous subsection regarding clash detection (5.2. Clash detection. Demonstration of method). The only difference is the rulesets and the rules themselves.

In order to demonstrate model completion checking, one of the standard set of rules "BIM Model Structure Validation" was selected (Figure 39). This set of rules checks for the presence of basic components in the model, such as: storeys, spaces, basic structural elements (walls, slabs, beams, etc.).

After checking the model according to the selected rulesets, several errors were identified indicating the absence of the following components: beams, columns, roofs, spaces and stairs (Figure 39).

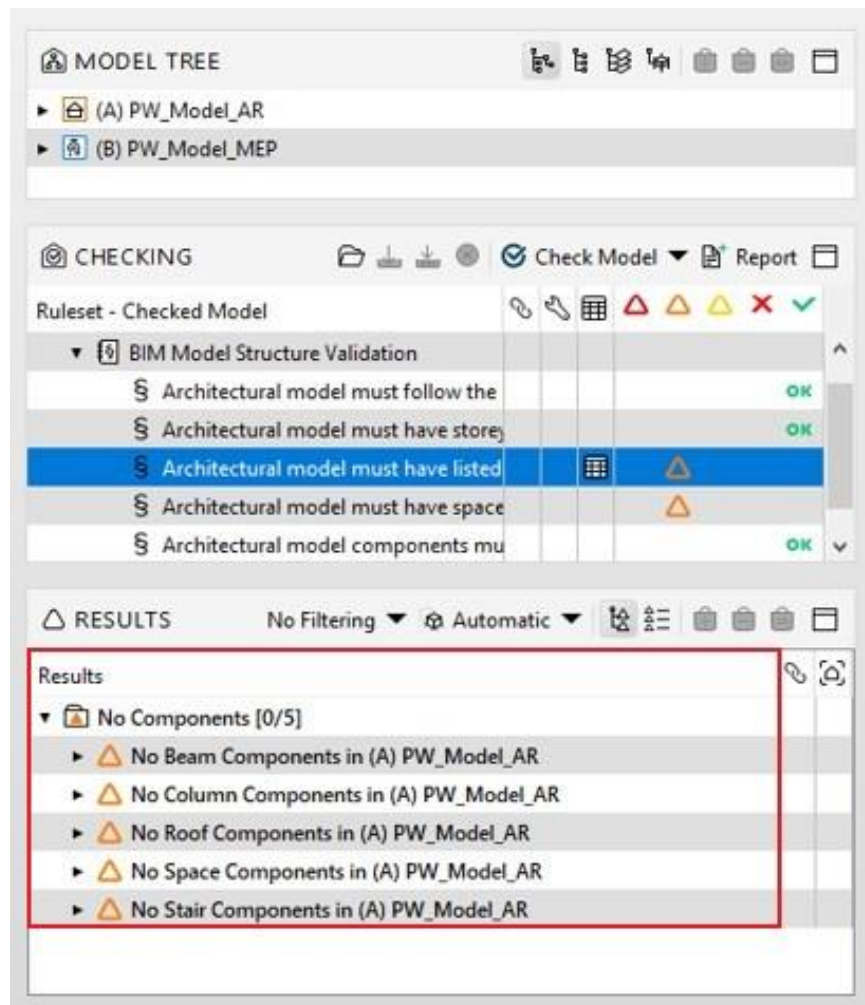


Figure 39. Results of model completion checking in Solibri Model Checker

The Figure 40 clearly demonstrates the presence of such elements as walls, slabs, windows and doors, and the absence of such elements as beams, columns, roofs, and others.

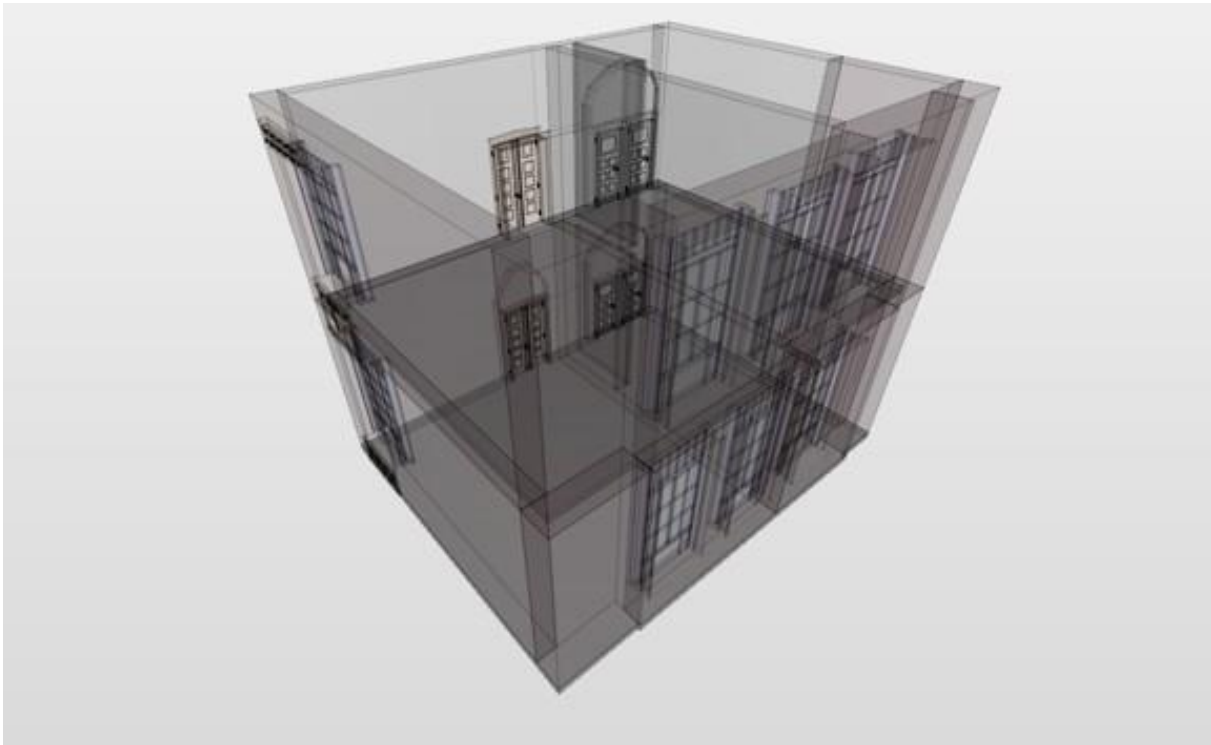


Figure 40. Existing BIM model elements in Solibri Model Checker

6 CONCLUSION

The present research aimed to conduct a state-of-the-art analysis of BIM-based model checking methods, with the main focus on compiling an intelligible classification of these methods based on possible verification scenarios.

As a result of the literature review, it was concluded that scientific sources present many different classifications of methods for checking BIM models. However, most of them consider either too wide area of research, including methods that are still under development and have not found applications in practice, or they are too narrowly focused, considering only one method in great detail. At present, there has not been an up-to-date, complete, and clearly formulated classification of available methods for checking BIM models, which would be based on possible checking scenarios and have practical examples for visual representation.

Therefore, in this work, the model checking methods for BIM were divided into three groups: clash detection, model content checking, and code checking. Each of the methods is in turn divided into subcategories, providing examples for each one. Based on the proposed structure of this classification of BIM-based model checking methods, an analysis of contemporary software was carried out. The analysis involved evaluating the functionality of the software, determining the exchange file formats, and estimating the cost of the licenses. Finally, the two of three methods presented were demonstrated on a small-scaled BIM model using the Solibri Model Checker software.

Overall, the proposed classification of BIM-based model checking methods provides a comprehensive framework for understanding the available methods and their practical applications in the construction industry.

Moreover, this paper also highlights the opportunities for further innovation and development regarding the considered topic, providing a foundation for future research.

For example, one of these opportunities concerns the expansion of the functionality of software for checking BIM models: developing an additional module that gives the possibility to select a normative document, scan it, extract rules, and add them to the software for further checks. Additionally, besides the presented methods for checking models, there are methods discussed earlier that are under research and development: smart object checking and design option checking.

The implementation of these opportunities could bring the technology of BIM-based model checking to a new level, make a great contribution to AEC sector, and lead to further development.

7 REFERENCES

- Abdalhameed, B. & Naimi, S. (2023). Based BIM techniques to clash detection for construction projects. *Periodicals of Engineering and Natural Sciences*, 11(1), 239–245. <https://doi.org/10.21533/pen.v11i1.3461>
- Akhmetzhanova, B., Nadeem, A., Hossain, M. A. & Kim, J. R. (2022). Clash Detection Using Building Information Modeling (BIM) Technology in the Republic of Kazakhstan. *Buildings*, 12. <https://doi.org/10.3390/buildings12020102>
- Andrich, W., Daniotti, B., Pavan, A. & Mirarchi, C. (2022). Check and Validation of Building Information Models in Detailed Design Phase: A Check Flow to Pave the Way for BIM Based Renovation and Construction Processes. *Buildings*, 12. <https://doi.org/10.3390/buildings12020154>
- Aydın, M. (2022). A Review of BIM-Based Automated Code Compliance Checking: A Meta-Analysis Research. *Automation and Control - Theories and Applications*. Vorab-Onlinepublikation. <https://doi.org/10.5772/intechopen.101690>
- Azhar, S. (2011). Building information modeling (BIM): Trends, benefits, risks, and challenges for the AEC industry. *Leadership and Management in Engineering*, 11(3), 241–252. [https://doi.org/10.1061/\(ASCE\)LM.1943-5630.0000127](https://doi.org/10.1061/(ASCE)LM.1943-5630.0000127)
- Bazjanac, V. (2008). IFC BIM-Based Methodology for Semi-Automated Building Energy Performance Simulation. *Proceedings of the 25th International Conference on Information Technology in Construction*.
- Belsky, M., Sacks, R [Rafael] & Brilakis, I. (2016). Semantic Enrichment for Building Information Modeling. *Computer-Aided Civil and Infrastructure Engineering*, 31, 261–274. <https://doi.org/10.1111/mice.12128>
- Berlo, L., Drogemuller, R. & Omrani, S. *What is Information Delivery Specification*. <https://www.buildingsmart.org/what-is-information-delivery-specification-ids/>
- Bolpagni, M. (2013). *The Implementation of BIM within the Public Procurement: A model-based approach for the construction industry* [Thesis for: Master of Science]. VTT Technical Research Centre of Finland, Espoo.
- Bryde, D., Broquetas, M. & Volm, J. M. (2013). The project benefits of building information modelling (BIM). *International Journal of Project Management*, 31(7), 971–980. <https://doi.org/10.1016/j.ijproman.2012.12.001>
- buildingSMART International. *Information Delivery Specification IDS: Machine readable Information Delivery Specification*. <https://technical.buildingsmart.org/projects/information-delivery-specification-ids/>
- buildingSMART International. *mvdXML*. <https://technical.buildingsmart.org/standards/ifc/mvd/mvdxml/>

- Chidambaram, S. (2020). The application of clash-detection processes in building information modelling for rebars. *Proceedings of the Institution of Civil Engineers: Smart Infrastructure and Construction*, 172(2), 53–69. <https://doi.org/10.1680/jsmic.20.00005>
- Di Giuda, G. M., Bonomi Boseggia, A. & Francesco, P. (2021). Development of a Rule-Based System for Automated BIM Code-Checking. *Proceedings of International Structural Engineering and Construction*, 8(1). [https://doi.org/10.14455/ISEC.2021.8\(1\).AAE-08](https://doi.org/10.14455/ISEC.2021.8(1).AAE-08)
- Di Giuda, G. M., Tagliabue, L. C., Locatelli, M., Pattini, G., Paleari, F., Campi, S. & Tucci, A. (2022). BIM Model Content Checking: Short And Long Term Strategies for Large Building Portfolio Owners. *EUBIM 2022 - BIM International Conference / 11º Encuentro de Usuarios BIM*, 140–150. <https://doi.org/10.4995/EUBIM2022.2022.15784>
- Eastman, C., Lee, J [Jae-min], Jeong, Y. & Lee, J [Jin-kook] (2009). Automatic rule-based checking of building designs. *Automation in Construction*, 18, 1011–1033. <https://doi.org/10.1016/j.autcon.2009.07.002>
- Eastman, C., Teicholz, P., Sacks, R [R.] & Liston, K. (2011). *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*. John Wiley & Sons.
- Gade, P. N., Hansen, R. & Svidt, K. (2018). BIM-based model checking in a business process management environment. *European Conference on Product and Process Modelling (ECPPM 2018)*, 12, 305–312. <https://doi.org/10.1201/9780429506215-38>
- Gade, P. N., Lauritzen, D. H., Andersen, M. & Hjelseth, E. (2022). How Practice Is Represented in BIM-Based Model Checking Research – A Literature Review and Reflections. *Conference: European Conference on Product and Process Modeling*.
- Gade, P. N. & Svidt, K. (2021). Exploration of practitioner experiences of flexibility and transparency to improve BIM-based model checking systems. *Journal of Information Technology in Construction*, 26, 1041–1060. <https://doi.org/10.36680/j.itcon.2021.055>
- Gao, H. & Zhong, B. (2022). A blockchain-based framework for supporting BIM-based building code compliance checking workflow. *IOP Conference Series: Materials Science and Engineering*, 1218. <https://doi.org/10.1088/1757-899X/1218/1/012016>
- Ghaffarianhoseini, A., Tookey, J., Naismith, N., Azhar, S., Efimova, O. & Raahemifar, K. (2017). Building Information Modelling (BIM) uptake: Clear benefits, understanding its implementation, risks and challenges. *Renewable and Sustainable Energy Reviews*, 75, 1046–1053. <https://doi.org/10.1016/j.rser.2016.11.083>
- Guedes, Í., Andrade, M. & Carvalho, A. (2021). BIM-Based Airport Design Project Standardization (IFC) for Use of Code Checking. *Formal Methods in Architecture, Advances in Science, Technology & Innovation*, 37–47. https://doi.org/10.1007/978-3-030-57509-0_4
- Herrera, M. (2021). *Soft Clashes en Navisworks*. <https://msistudio.com/soft-clashes-en-navisworks/>

- Hjelseth, E. (2015). BIM-based Model Checking (BMC). *Building Information Modeling: Applications and Practices*, 33–61. <https://doi.org/10.1061/9780784413982.ch02>
- Hjelseth, E. (2016). Classification of BIM-based Model checking concepts. *Journal of Information Technology in Construction (ITcon)*, 21, 354–369.
- Hjelseth, E., Lassen, A. K. & Dimyadi, J. (2016). Development of BIM-based model checking solutions – ongoing research and practitioners' demand. *Proceedings of the 33rd CIB W78 Conference 2016*.
- Hjelseth, E. & Nisbet, N [Nick] (2011). Capturing normative constraints by use of the semantic mark-up RASE methodology. *Proceedings of CIB W78-W102 Conference*.
- Ismail, A. S., Ali, K. N. & Iahad, N. A. (2017). A Review on BIM-based automated code compliance checking system. *International Conference on Research and Innovation in Information Systems (ICRIIS)*, 1–6. <https://doi.org/10.1109/ICRIIS.2017.8002486>
- Laakso, M. & Kiviniemi, A [A.] (2012). The IFC standard - A review of history, development, and standardization. *Electronic Journal of Information Technology in Construction*, 17, 134–161. <https://www.scopus.com/inward/record.uri?eid=2-s2.0-84862009953&partnerID=40&md5=565a783972837f84e26e3d84ecdad107>
- Li, Y., Zhao, Q., Liu, Y., Hei, X. & Li, Z. (2021). Semiautomatic Generation of Code Ontology Using ifcOWL in Compliance Checking. *Advances in Civil Engineering*, 2021. <https://doi.org/10.1155/2021/8861625>
- Preidel, C. & Borrmann, A. (2018). BIM-based Code Compliance Checking. *Building Information Modeling - Technology Foundations*, 367–381. https://doi.org/10.1007/978-3-319-92862-3_22
- Rehman, S. U., Lee, S., Choi, J. & Kim, I. (2021). Analysis of Development of Open BIM-based Automated Rule Checking System in KBIM Project. *13th Asian Forum on Graphic Science*.
- Ren, R. & Zhang, J [Jiansong] (2019). Model information checking to support interoperable BIM usage in structural analysis. *ASCE International Conference on Computing in Civil Engineering*, 361–368. <https://doi.org/10.1061/9780784482421.046>
- Savitri, D. M., Juliastuti & Pramudya, A. A. (2020). Clash detection analysis with BIM-based software on midrise building construction project. *IOP Conference Series: Earth and Environmental Science*, 426(1). <https://doi.org/10.1088/1755-1315/426/1/012002>
- Schwabe, K., Teizer, J. & König, M. (2019). Applying rule-based model-checking to construction site layout planning tasks. *Automation in Construction*, 97, 205–219. <https://doi.org/10.1016/j.autcon.2018.10.012>
- Shen, Q., Wu, S., Deng, Y., Deng, H. & Cheng, J. C. P. (2022). BIM-Based Dynamic Construction Safety Rule Checking Using Ontology and Natural Language Processing. *Buildings*, 12. <https://doi.org/10.3390/buildings12050564>
- Solihin, W. & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction*, 53, 69–82. <https://doi.org/10.1016/j.autcon.2015.03.003>

7 References

- Soliman-Junior, J., Tzortzopoulos, P., Baldauf, J. P., Pedo, B., Kagioglou, M., Formoso, C. T. & Humphreys, J. (2021). Automated compliance checking in healthcare building design. *Automation in Construction*, 129. <https://doi.org/10.1016/j.autcon.2021.103822>
- Succar, B. (2009). Building information modelling framework: A research and delivery foundation for industry stakeholders. *Automation in Construction*, 18, 357–375. <https://doi.org/10.1016/j.autcon.2008.10.003>
- Sun, H. & Kim, I. (2022). Applying AI technology to recognize BIM objects and visible properties for achieving automated code compliance checking. *Journal of Civil Engineering and Management*, 28(6), 497–508. <https://doi.org/10.3846/jcem.2022.16994>
- Sydora, C. & Stroulia, E. (2019). Towards Rule-Based Model Checking of Building Information Models. *36th International Symposium on Automation and Robotics in Construction (ISARC 2019)*, 1327–1333. <https://doi.org/10.22260/ISARC2019/0178>
- Volk, R., Stengel, J. & Schultmann, F. (2014). Building Information Modeling (BIM) for existing buildings - Literature review and future needs. *Automation in Construction*, 38, 109–127. <https://doi.org/10.1016/j.autcon.2013.10.023>
- Weise, M., Liebich, T., Nisbet, N [Nicholas] & Benghi, C. (2016). IFC model checking based on mvdXML 1.1. *11th European Conference on Product & Process Modeling*.
- Zhou, Y., She, J., Huang, Y., Li, L., Zhang, L. & Zhang, J [Jiashu] (2022). A Design for Safety (DFS) Semantic Framework Development Based on Natural Language Processing (NLP) for Automated Compliance Checking Using BIM: The Case of China. *Buildings*, 12. <https://doi.org/10.3390/buildings12060780>
- Zou, Y., Kiviniemi, A [Arto] & Jones, S. W. (2015). BIM-based Risk Management: Challenges and Opportunities. *Proceedings of the 32nd CIB W78 Conference*, 847–855.