Univerza *v Ljubljani*
Fakulteta *za gradbeništvo*
in geodezijo

MOHAMED OMAR

AUTOMATED BIM-BASED MODEL CHECKING WORKFLOWS WITH EXCHANGE INFORMATION REQUIREMENTS

AVTOMATIZIRANI DELOTOKI ZA PREVERJANJE SKLADNOSTI MODELOV BIM Z INFORMACIJSKIMI ZAHTEVAMI

BIM A+ European Master in Building Information Modelling

Master thesis No.:

Supervisor:
Assist. Prof. Tomo Cerovšek, Ph.D.

Ljubljana, 2023

Co-funded by the Erasmus+ Programme of the European Union

ERRATA

Page	Line	Error	Correction

»This page is intentionally blank«

BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK: **004.946:69(078.7)(043.3)**

Avtor: **Mohamed Omar**

Mentor: **Ass. Prof. Tomo Cerovšek**

Somentor: **Ekaterina Moskvina**

Naslov: **Avtomatizirani delotoki za preverjanje skladnosti modelov BIM z informacijskimi zahtevami**

dokumenta: **Magistrsko Delo**

Obseg in oprema: **110 str., 79 sl., 12 pregl., 3 pril**

Ključne besede: **BIM, openBIM, skladnost, preverjanje, BMC, IDS**

Izvleček:

Uspeh gradbenih projektov je zelo odvisen od kakovosti projektnih informacij. Upravljanje projektnih informacij, vključno z zahtevami za informacije, je temelj za učinkovito projektno sodelovanje. Preverjanje informacijskih zahtev, identifikacija in odpravljanje neskladij v kompleksnih projektih BIM je dolgotrajen in drag proces. Zato je avtomatizacija postopka preverjanja modela na osnovi BIM (BMC angl. BIM-based model checking) nepogrešljiva za večjo učinkovitost. Aplikativni raziskovalni pristop in kvalitativne metodologije različnih uporab BIM v kontekstu zahtev za projektne informacije so vodile do petstopenjskega BMC: razlaga pravil, priprava modela, izvajanje pravil, poročanje o rezultatih preverjanja in avtomatizirano ali polavtomatsko izvajanje kode za reševanje informacijskih težav.

Za BMC predlagamo tri delotoke V delotoku A se celoten BMC izvaja v avtorskem okolju, kar povečuje učinkovitost z manj izmenjavami modelov BIM. Delotok B se izvaja v samostojni programski opremi ali specializiranih orodjih BMC z razlago in izvdbo pravil. Delotok C razširjeno uporablja openBIM za BMC in interpretaciji pravil z IDS. V študiji primera so bili implementirani delotoki BMC in stopnja informacijskih potreb vhodni podatek za razlage pravil, ki opredeljuje informacijske zahteve za popise količin in oceno stroškov. Študija podaja kritično oceno priložnosti in omejitve delovnih tokov BMC ter podaja poglobljeno razpravo o potencialu in učinkih na upravljanje projektnih informacij.

»This page is intentionally blank«

BIBLIOGRAPHIC– DOKUMENTALISTIC INFORMATION AND ABSTRACT

UDC: 004.946:69(078.7)(043.3)

Author: Mohamed Omar

Supervisor: Ass. Prof. Tomo Cerovšek

Co-supervisors: Ekaterina Moskvina (Company Tutor)

Title: Automated BIM-based Model Checking Workflows with Exchange Information Requirements

Document type: Master Thesis

Scope and tools: 110 p., 79 fig., 12 tab., 3 app.

Keywords: BIM, openBIM, compliance, checking, BMC, IDS

Abstract:

The success of building projects largely depends on the quality of project information. Therefore, project information management, including information requirements, establishes the foundation for effective project collaboration. Checking information requirements, identification, and resolution of information issues in complex BIM projects is a very time-consuming and costly process. Thus, automating the BIM-based Model Checking (BMC) process is indispensable for resource savings and efficiency. Through an applied research approach and qualitative methodology, the study investigated various BIM uses within the context of information requirements. Subsequently, the research introduced a five-stage BMC process, which involves adapting the traditional BMC process with an additional stage: Rule interpretation, Model preparation, Rules execution, Reporting check results, and Automated or Semi-automated code execution for resolving the information management issues.

To implement the BMC process, three distinct workflows are proposed. In Workflow A, the entire BMC process is conducted within a designated repository environment, enhancing efficiency by minimising the need for BIM model exchanges. Workflow B is conducted in standalone software or specialised BMC tools for BMC core stages of rule interpretation and execution. Workflow C extensively employs openBIM throughout the entire BMC process, mainly rule interpretation using IDS. Throughout the case study, the predefined BMC workflows were implemented. The Level of Information Need framework was employed as input for the rule interpretation stage, defining information requirements for QTO and cost estimation. The study critically evaluated the opportunities and limitations of BMC workflows, discussing the potential and implications for project information management.

»This page is intentionally blank«

ACKNOWLEDGEMENTS

I express my utmost praise to Allah the Almighty, the Most Gracious, and the Most Merciful for bestowing His blessings upon me throughout my academic journey and enabling me to complete this thesis. I want to express my gratitude to several individuals who have played a pivotal role in completing this work.

First and foremost, I would like to express my sincerest appreciation to my supervisor, Tomo Cerovšek, for his invaluable guidance, unwavering support, and insightful feedback throughout this work. His expertise has been instrumental in shaping the direction of this work. I convey my heartfelt thanks to my company tutor, Ekaterina Moskvina, whose expertise has illuminated the work and provided me with invaluable insights. Furthermore, I especially thank Dr. José Granja, from the University of Minho, Portugal, for his support and insightful feedback.

My family and lifelong friends deserve a special mention for their encouragement, patience, and understanding. Their unconditional support has been my driving force, and I am truly blessed to have them by my side.

I extend my heartfelt gratitude to the BIM A+ family, a group that has profoundly impacted my personal and professional journey. The camaraderie and shared passion within this family, as we fondly refer to it, have been an endless source of inspiration and joy.

Finally, my acknowledgement is adorned with profound gratitude and utmost respect for the BIM A+ program consortium and staff. Their unwavering support and tireless efforts in shaping this program to be as impactful and valuable as possible have not gone unnoticed. Their dedication has undoubtedly enriched my experience, and I am grateful for their contributions.

»This page is intentionally blank«

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ACRONYMS AND ABBREVIATIONS

AECO	Architecture, Engineering, Construction and Operation
AIM	Asset Information Model
AIR	Asset Information Requirements
BCF	BIM Collaboration Format
BIM	Building Information Modelling
BMC	BIM-based Model Checking
BOM	Bill of Materials
BOQ	Bill of Quantities
CCC	Compliance Code Checking
CORENET	Construction and Real Estate Network
EDM	Express Data Manager
EIR	Exchange Information Requirement
IDM	Information Delivery Manual
IDS	Information Delivery Specification
IFC	Industry Foundation Classes
ISO	International Organization for Standardization
LOD	Level of Development / Level of Detail
MVD	Model View Definition
OIR	Organisational Information Requirement
PIM	Project Information Model
PIR	Project Information Requirement
QTO	Quantity Take-Off
RASE	Requirement, Applicability, Selection, and Exception
regex(es)	Regular Expression(s)
RIBA	Royal Institute of British Architects
SMC	Solibri Model Checker
SWRL	Semantic Web Rule Language
VPL	Visual Programming Language
WBS	Work Breakdown Structure

LIST OF APPENDICES

»This page is intentionally blank«

1 INTRODUCTION

Building Information Modelling (BIM) has revolutionised the Architecture, Engineering, Construction, and Operation (AECO) industry, offering a collaborative and data-centric approach to building design, construction, and facility management. This transformative technology has brought remarkable benefits, improving efficiency, coordination, and decision-making throughout the project lifecycle. However, ensuring the quality of BIM models, including adherence to design standards and information requirements, remains a significant challenge. Often, achieving the desired level of quality necessitates incorporating hard-coding design standards into various design software.

Consequently, utilising BIM industry-standard practices enhances interoperability, enabling seamless information integration for both human stakeholders and technological systems. This process empowers end-users to extract valuable insights ([ISO 19650-1], 2018). The cycle of information exchange is a dynamic cycle between the information provider and the information receiver (specifier), as illustrated in Figure 1. Thus, the adoption of uniform information structures across various sectors fosters consistency. It also encourages controlled repetition and predictability, ultimately resulting in tangible efficiency gains for enterprises. To facilitate comprehension of deliverables and the establishment of automated validation rules, comprehensive information requirements must be carefully implemented in the design and development of BIM models (Fenves et al., 1995).

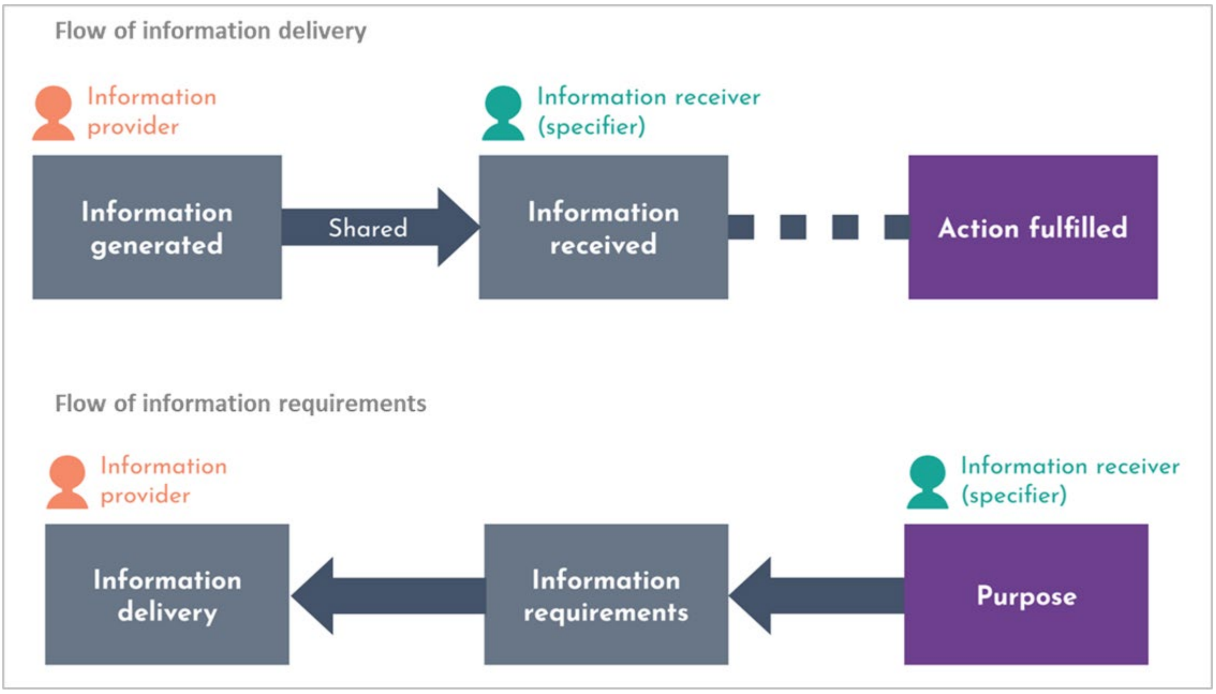


Figure 1: Information exchange flows
Resources: (UK BIM Framework, 2020)

Furthermore, the reliance on traditional drawing-based information exchange is compounding the challenges, particularly in the design phase. According to Liu et al. (2013, as cited in Jubierre, 2015), this technology cannot represent the relationships between the information elements they display. As a result, designers and engineers must manually update the geometric model to reflect changes in the project, a time-consuming and error-prone process.

1.1 Research Importance and Focus

Manual checking processes rely on 2D Drawings and textual documents (Preidel & Borrmann, 2016). As a result, these processes are causing delays and cost increases. Checking Models is not limited to the construction phase but is a continuous process through the project's Lifecycle (Amor & Dimyadi, 2021).

Consequently, researchers have been developing model-based rule-checking systems to assist in the AECO processes. These systems aim to automate checking compliance with rules and regulations, improving efficiency and reducing errors. Using rule-based algorithms in model-based checking processes is promising because it captures the logic behind the checking routines (Beach et al., 2015; Dimyadi et al., 2016; Schwabe et al., 2016, 2019).

It is argued that Checking Processes are costly. Meanwhile, the unsolved errors may affect the project budget. For example, in a large-scale housing project in London, Modifications of Steep and narrow ramps cost £800,000 for the design and construction phases (Ding et al., 2006). According to Ingvaldsen (1994, 2001, as cited in Hjelseth, 2016), as much as 40% of defects may be related to blunders in the design process. Lopez & Love (2012) concluded that unsolved design errors may increase the project cost throughout the life cycle.

In the construction phase, it was shown that unsolved design errors can increase the project's total cost by 14.21% relative to the project contract value. The errors direct cost represents 6.85 % while the indirect cost represents 7.36% (Hjelseth, 2016). However, the Automation process of the BIM-based model checking is still not adopted enough. The only fully automated solution was in Singapore, where an automated process was implemented but discontinued (Beach et al., 2020).

In summary, the process of BIM-based Model Checking (BMC) is a sequence procedure affecting the whole project outcome. The study focuses on Automating the BMC process and the information requirement needed across the project lifecycle.

1.2 Research Aim and Objectives

The research project aims to implement the BMC process and information requirement framework practically. The research will be guided by the following objectives:

A- Explore different BIM uses, including code validation, quantity take-off, cost estimation, and clash detection, within the context of information requirements for the BMC process.

B- Adapt the BMC process and develop automated workflows for the BMC, showcasing the implementation of each workflow. These workflows should encompass both black-box and white-box approaches.

C- Define typical scenarios for information requirements that may arise during the BMC process, whether geometrical or alphanumerical.

D- Implement the information requirements framework (Level of Information Need) and scenarios with the proposed BMC workflows to practically assess these workflows and identify their limitations and opportunities.

1.3 Partnership for the Thesis

The thesis was developed in collaboration with Protim Ržišnik Perc d.o.o., A leading Slovenian consulting and design company. The company provides its clients with consulting services, represents them, and leads their investment projects in the design phase through the pre-construction and construction management phases. The company specialise in Integrated Design, Project management, Construction management, Master Planning, Consulting Services, Expert Opinions, Geodesy, Energy Consulting, Valuations and Legal Support, Facility Management, and BIM.

The partnership included necessary support from the Protim Ržišnik company team with the required information to conduct the case study and other related information regarding BIM Processes within the company.

1.4 Thesis Structure

The thesis is structured in seven main chapters besides references and appendixes as follows:

<u>Chapter 1 - Introduction:</u> The chapter introduces the research background and its importance, aim and objectives with an outline of the thesis structure and timeline.

<u>Chapter 2 - Literature Review:</u> The chapter provides an in-depth exploration of the research's Literature review topics of BIM-based Model Checking (BMC), Information Requirements, and openBIM Standards.

Chapter 3 - Methodology: The chapter outlines the applied research approach and qualitative methodology adopted in this study. It provides insights into the chosen applied research approach, its type, data collection methods, and the software/tools used.

Chapter 4 - BMC Workflows and Scenarios: The chapter discusses the extensive application of BMC throughout the entire project lifecycle, focusing on information requirement relevance across BIM uses and exploring real-world scenarios that demonstrate the practical utilisation of BMC. Additionally, this chapter introduces the proposed BMC workflows as a main objective of the thesis.

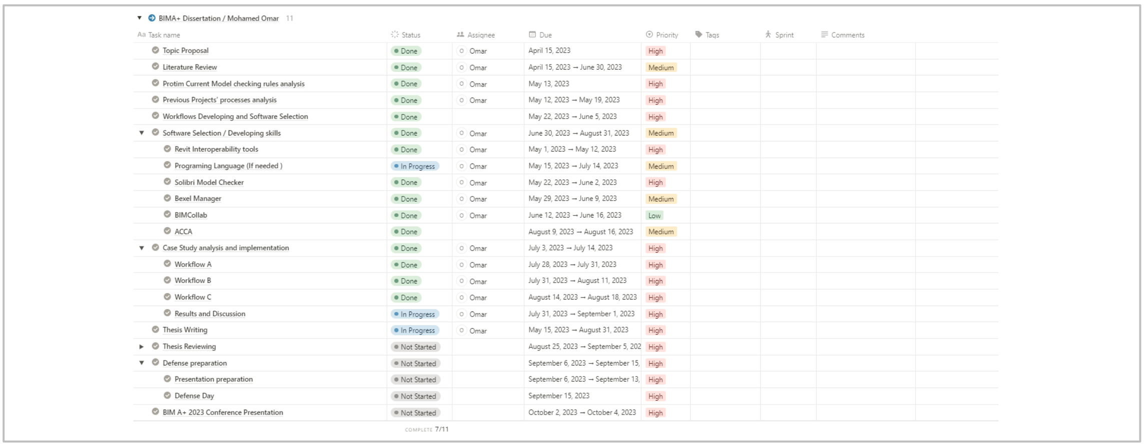
Chapter 5 - Case study: The chapter provides a practical implementation involving the case study project provided by the thesis company partner. The implementation includes defining the Level of Information Need framework for specific BIM use of QTO and Cost estimation. Each of BMC's proposed workflows is implemented in-depth using a variety of BIM software and tools.

Chapter 6 - Discussion: The chapter includes an overview of the implemented information requirement framework with an analysis of the BMC workflow implementation results, limitations, opportunities, and Software review and comparison.

Chapter 7 - Conclusion: This chapter provides a comprehensive conclusion, including theoretical and practical contributions, key findings, and recommendations for future research about the information requirements and BMC.

1.5 Thesis Timeline

The research work was distributed over the entire thesis period, as shown in Figure 2. The timeline includes the preparation of research content and the main software learning timeline. Meetings were conducted on a bi-weekly basis with the supervisor and either bi-weekly or monthly with the company mentor.



BIMA+ Dissertation / Mohamed Omar 11

Task name	Status	Assignee	Due	Priority	Tags	Sprint	Comments
Topic Proposal	Done	Omar	April 15, 2023	High			
Literature Review	Done	Omar	April 15, 2023 → June 30, 2023	Medium			
Protim Current Model checking rules analysis	Done	Omar	May 13, 2023	High			
Previous Projects' processes analysis	Done	Omar	May 12, 2023 → May 19, 2023	High			
Workflows Developing and Software Selection	Done		May 22, 2023 → June 5, 2023	High			
Software Selection / Developing skills	Done	Omar	June 30, 2023 → August 31, 2023	Medium			
Revit Interoperability tools	Done	Omar	May 1, 2023 → May 12, 2023	High			
Programing Language (If needed)	In Progress	Omar	May 15, 2023 → July 14, 2023	Medium			
Solibri Model Checker	Done	Omar	May 22, 2023 → June 2, 2023	High			
Bexel Manager	Done	Omar	May 29, 2023 → June 9, 2023	Medium			
BIMCollab	Done	Omar	June 12, 2023 → June 16, 2023	Low			
ACCA	Done		August 9, 2023 → August 16, 2023	Medium			
Case Study analysis and implementation	Done	Omar	July 3, 2023 → July 14, 2023	High			
Workflow A	Done	Omar	July 28, 2023 → July 31, 2023	High			
Workflow B	Done	Omar	July 31, 2023 → August 11, 2023	High			
Workflow C	Done	Omar	August 14, 2023 → August 18, 2023	High			
Results and Discussion	In Progress	Omar	July 31, 2023 → September 1, 2023	High			
Thesis Writing	In Progress	Omar	May 15, 2023 → August 31, 2023	High			
Thesis Reviewing	Not Started		August 25, 2023 → September 5, 202	High			
Defense preparation	Not Started		September 6, 2023 → September 15,	High			
Presentation preparation	Not Started		September 6, 2023 → September 13,	High			
Defense Day	Not Started		September 15, 2023	High			
BIM A+ 2023 Conference Presentation	Not Started		October 2, 2023 → October 4, 2023	High			

COMPLETE 7/11

Figure 2: Thesis timeline

2 LITERATURE REVIEW

The research focuses on the domains of BIM-based model checking (BMC) and its implementation, opportunities, and limitations in the context of information requirements, as illustrated in Figure 3. The BMC implementation can be achieved through the black-box and open-box approaches (it will be explained in-depth in the following sections). In this chapter, a literature review is conducted with a detailed overview of BMC and information requirements with an overview of openBIM Standards that is used later in subsequent chapters.

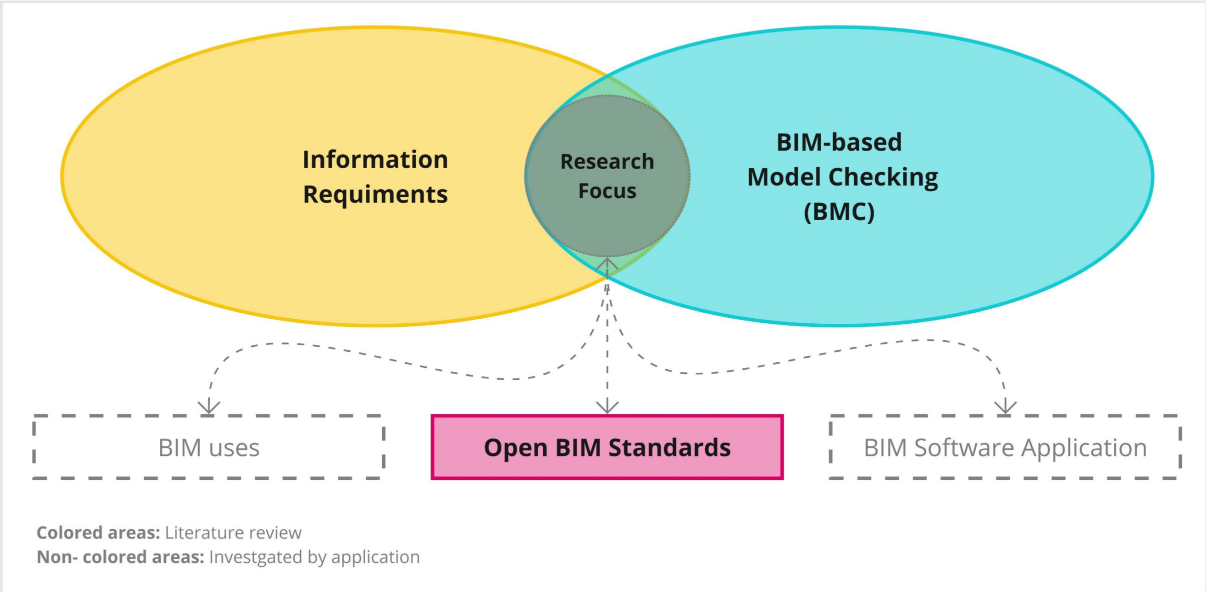


Figure 3: Research literature review areas

2.1 BIM-based Model Checking (BMC)

Model checking is a subsequent procedure, and, in many cases, it is limited to model quality and collision checking (Schwabe et al., 2019). Consequently, the designs must be thoroughly examined and revised if any issues or conflicts arise due to predefined checks. McGraw-Hill (2014, as cited in Hjelseth, 2016) expected Model Checking to boost BIM uses, especially in industrial sectors. Guedes & Andrade (2019) demonstrate how model checking can standardise the evaluation of designs, improve efficiency, and enhance accuracy.

BMC, as a specific terminology for BIM-based model checking (Hjelseth, 2016), has become an integral part of the BIM process, playing a crucial role in ensuring the accuracy, compliance, and quality of building models. In recent years, BMC has evolved beyond the conventional notion of clash detection, encompassing a broader range of checking processes such as validation, compliance, and design solution checking.

By leveraging advanced technology and methodologies, BMC enables professionals in the AECO industry to verify adherence to building codes, meet client demands, optimise energy performance, and adhere to industry standards. The following sub-sections explore the terminologies, concepts, applications, and successful case studies of BMC.

2.1.1 BMC Synonyms Terminologies

While defining BMC Processes as clash detection is theoretically not wrong, it is very narrow and limited. Instead, Clash detection and Duplication checking could be considered the common examples of BMC that rely on Boolean expressions (Hjelseth, 2009, 2015, 2016).

Thus, other checking processes such as client demand fulfilment, energy performance, public demands, and standards can be conducted. That is why BMC has many other terminologies with the same purpose while not mentioned in the ISO Concept Database. Those terminologies are Clash detection, Model checking, Validation checking, Compliance checking, Code compliance checking, Quality checking and many more (Hjelseth, 2016). For specific uses, some terminologies and abbreviations could be used. For insurance, Compliance code checking (CCC) describes the compliance process of codes (Aydın, 2022).

Meanwhile, it is derived that the dominating actors of BMC software use different terminologies. For instance, Solibri Model Checker (SMC) uses Compliance Control, Design Review, Analysis and Code Checking terminologies. Autodesk Navisworks uses Model Review, Interference Checking and Clash Detection. Tekla BIMsight defines clashes with the Check term. In the research, the abbreviation "BMC" was employed to encapsulate the comprehensive procedure encompassing BIM-based Model Checking, encompassing all the terminologies mentioned above.

2.1.2 BMC Concepts

Generally, one common BMC concept is checking the model against specific clauses in building codes. However, some approaches focus on checking specific object types of the model to check related rules and behaviour of the object in its environment (Ding et al., 2006).

Hjelseth (2016) classified the main concepts of BMC, as shown in Figure 4. Venn Diagrams [1] were used to express how Model-checking rules processes interact with different kinds of information. BMC has five main predefined characteristics: Structure of information, Information requirement, Structure of rule, Logic in rule, and Outcome.

[1] Venn Diagram: an illustration that uses circles to show the relationships among things or finite groups of things. Circles that overlap have a commonality, while circles that do not overlap do not share those traits. Venn diagrams help to represent the similarities and differences between two concepts visually.

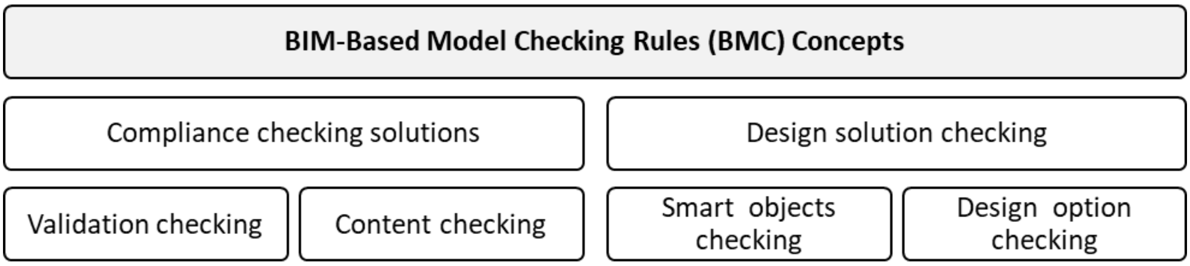


Figure 4: BIM-based Model Checking (BMC) concepts overview
Resource: Data was derived from (Hjelseth, 2016)

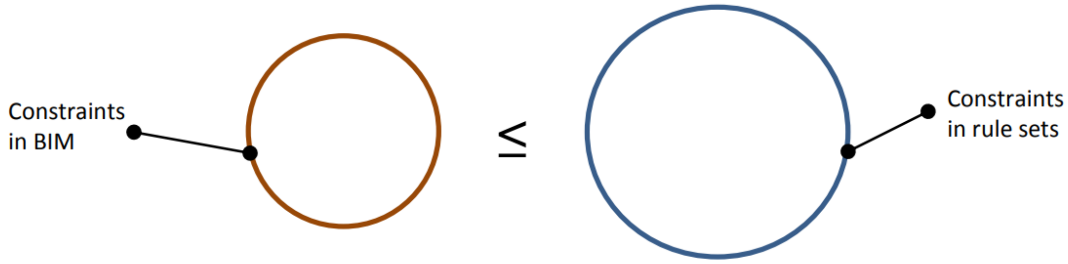
Table 1: BIM-based Model Checking (BMC) concepts summary
Resource: Graphs and Data were extracted from (Hjelseth, 2016)

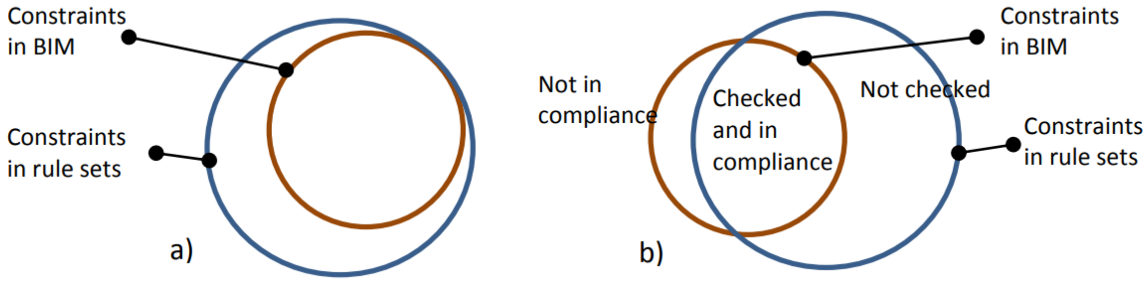
Concept type 1: Validation checking

The main concept is to compare specific constraints in an information model against pre-designed rulesets. The results will be:

a) BIM model constraints meet the rule sets' constraints.

b) Constraints in the BIM model meet some of the constraints in rule sets and contradict with some rules. Also, some rules are not to be checked due to the lack of information in the BIM model.

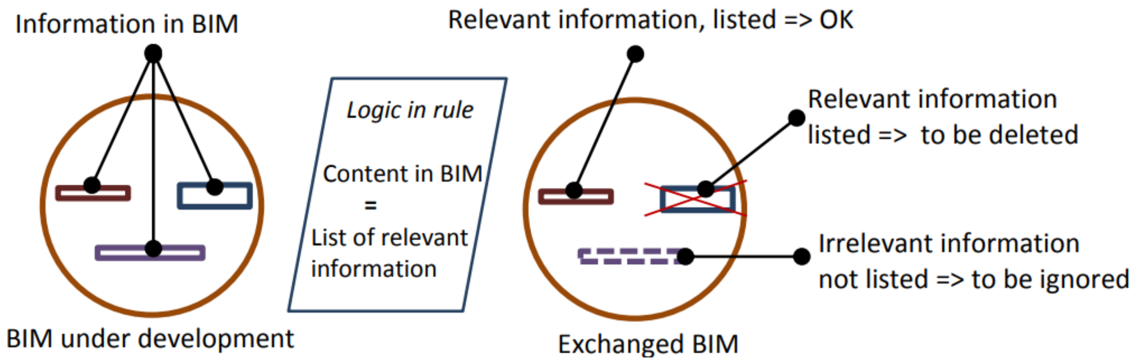




Concept type 2: Model content checking

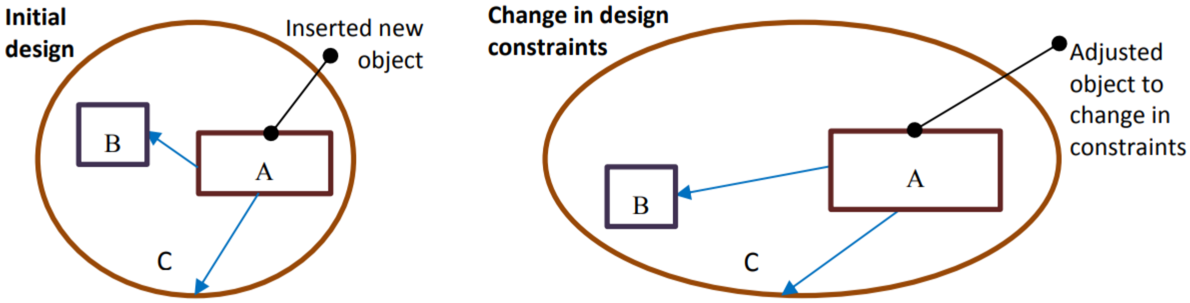
The main concept is to compare a predefined list of information according to project needs. The checking will lead to one of two results: Identified or not identified.

This concept can also be used to check unnecessary information as part of safety, commercial information and/or intellectual property.



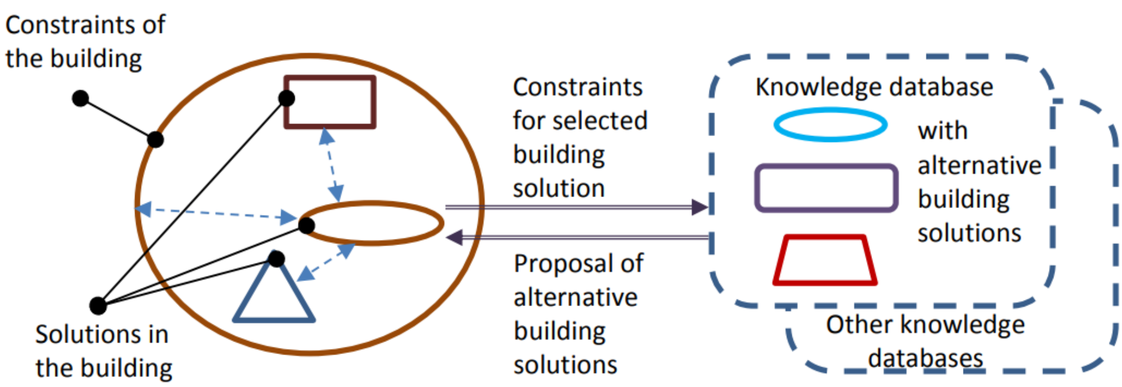
Concept type 3: Smart object checking

The concept relies on defining rules and algorithms in the object itself. The object will adapt to its environment and observe. The object adaptation is not only limited to the geometrical behaviour but also interacts with non-geometrical properties such as material properties specification.



Concept type 4: Design option checking

This concept creates rules that identify predefined design situations and suggest optimal solutions based on the knowledge base [1]. Nevertheless, this concept is not mature yet in the AECO Industry, and dedicated software solutions for such concepts are unknown.



[1] knowledge database: A knowledge base is a set of sentences, each sentence given in a knowledge representation language, with interfaces to tell new sentences and to ask questions about what is known, where either of these interfaces might use inference. It is a technology used to store complex structured data used by a computer system.

2.1.3 BMC Processes and Functionalities

Rule-based checking systems are large applications and require significant software utilities to provide the functionality. Eastman et al. (2009) expected that the model-checking system would go far beyond the code checking to be a standard tool used throughout the building lifecycle. These systems allow standardisation in the Model evaluation processes, ensuring compliance with building codes and regulations to promote consistency and quality in the project life cycle (Guedes & Andrade, 2019).

Moreover, rule checking increases the capacity of design project analysts, enabling them to assess designs more efficiently and accurately. The reusability of rules is another benefit of BIM-based rule-checking models. Users can create a library of rules that can be applied to future projects.

Eastman et al. (2009) observed four classes of capabilities for implementing a functional rule-based checking and reporting system: Rule interpretation, Building Model preparation, Rule execution, and Reporting checking results, as illustrated in Figure 5. It can be summarised as follows:

A- The rule Interpretation phase defines the rules for building design in a human language format, such as written text, tables, or equations. These rules must be translated into a machine-processable format that the rule-checking software can understand. One common approach is to use first-order predicate logic as an intermediate language for mapping the rules from natural language to a processable form.

B- In the Model Preparation phase, the building model is prepared, ensuring it contains all the information required for rule checking. This information can be explicitly provided by the designer or derived by the computer through analysis or simulation. The model must be syntactically checked to ensure it carries the properties, names, and objects needed for the rule-checking task.

C- The rule Execution involves applying the prepared building model to the defined rules. This is where the actual rule checking takes place. The rule-checking software matches the capabilities and information available in the building model with the functions defined by the rules. The software assesses whether the model complies with the rules and identifies any violations or non-compliant conditions.

D- Reporting checking results is the final step in the rule-checking process. It involves generating reports that summarise the results of the rule checking. The reports include information about the design conditions that pass the rules and those that fail. It is essential to provide a detailed audit trail that validates the completeness of the check. The reporting should also distinguish between different instances of objects within the building model to accurately identify violations.

However, Schwabe et al. (2019) described the previous approach as traditional and explained that problems and issues should be solved manually depending on the application.

Another modern and advanced approach was suggested by Solihin et al. (2017) as part of the automation checking rules process (Schwabe et al., 2019). It consists of five steps as follows:

A- Distilling the rule content from the written text.

B- Transcribing the rule content into a rule language

C- Transforming the existing model data into the required knowledge representation

D- Executing the computable rules within a rule engine

E- Generating new knowledge from reasoning about existing knowledge via the expressed rules.

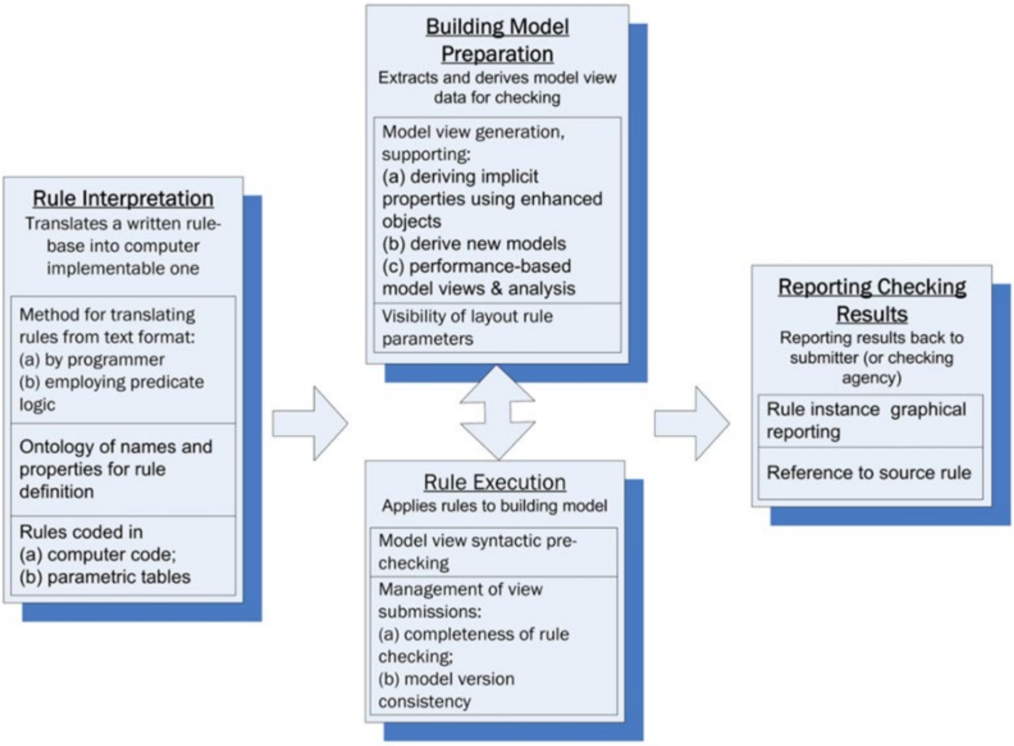


Figure 5: Rule-based platforms functionalities
Resource: (Eastman et al., 2009)

Nevertheless, The RASE (Requirement, Applicability, Selection, and Exception) method is a commonly used BMC approach. RASE is a structured approach to capture and classify information in rule documents. It involves categorising the content of the rules into four classes: requirement, applicability, selection, and exception. This classification helps organise and understand the logic behind the checking routines in the rule document.

The RASE method can capture and classify information in various domains, including the Architecture, Engineering and Construction (AEC) industry. The structured information obtained using the RASE method can then be mapped into existing rule languages for further processing (Hjelseth & Nisbet, 2011; Schwabe et al., 2016, 2019; Solihin & Eastman, 2015).

According to Zhang et al. (2013), a rule-based checking system can be implemented in two ways: through a design-based software application/plug-in or using an Industry Foundation Class (IFC)-based model viewer or checker. The software application allows architects, engineers, and designers to validate their building models during the design process. At the same time, the IFC-based approach focuses on data exchange and integration within the construction industry.

Moreover, data interoperability between different BIM platforms is a significant challenge in both approaches. Beach et al. (2006) adapted an automated code-checking system, which IFC as a standard model to the checking tool developed. This workflow was used in the design Consecutive stages of sketch design and detailed design and Documentation.

Beach et al. (2015) presented a rule-based semantic approach for automated regulatory compliance in the construction sector. The approach involves adding metadata to regulation documents, creating a regulation ontology, mapping regulations to data formats, and executing compliance checking using SWRL [1] rules.

Furthermore, Dimyadi et al. (2016) discussed the integration of the BIM Rule Language into a computer-aided compliance audit framework for extracting information from BIM and Regulatory Knowledge Models. This approach provides querying capabilities using an SQL-like syntax, including complex and nested expressions and performs logic checks for rule compliance. It also provides a lightweight geometry engine with standardised checking algorithms, such as intersection checks and shortest path finding, further simplifying the rule-checking process.

Additionally, Hjelseth (2016) discussed the importance of creating separate services that focus on the model checking rules and are independent of software implementation. At the same time (Eastman et al., 2009) expected that the model-checking system would go far beyond code checking to be a standard tool used throughout the building lifecycle.

2.1.4 BMC Applications and Challenges

Using a visual programming language (VPL) for defining pre-processing procedures in code compliance checking offers several advantages (Preidel et al., 2018; Preidel & Borrmann, 2015). Firstly, VPL allows for formulating checking and verification routines in a domain-specific and object-oriented manner. This means that the language is specifically designed to handle the requirements and complexities of code compliance checking, making it easier to express and implement the necessary procedures.

[1] Semantic Web Rule Language (SWRL) is a proposed language for the Semantic Web that can be used to express rules as well as logic, combining OWL DL or OWL Lite with a subset of the Rule Markup.

Secondly, it is strongly typed, which enforces strict information typing and validation. This helps to ensure the accuracy and reliability of the pre-processing procedures, reducing the risk of information duplication or inconsistencies.

Eastman et al. (2009) and Preidel & Borrmann (2016) discussed the challenges of verifiability and correctness of rules. In most known approaches, software developers are responsible for the computational translation of rules. Thus, any modification to the implemented code can be only conducted by software experts, which can limit the whole checking process.

Preidel & Borrmann (2016) discussed that the current challenge for the checking process is that most available tools are black-box [1] tools that usually use rule templates based on hard-coded rules and are inflexible. This inflexibility raises the need for a "white-box" workflow as a transparent approach to understand how the checking process is happening and the ability to customise the whole process. For instance, the system CORENET with the tool of e-PlanCheck is an example of such a black-box implementation. It is used in Singapore for checking the compliance of building digital models against the building codes, and the code is based on the FORNAX Library, developed by a private company. So, the checking processes are not visible. The e-PlanCheck system uses the IFC model and Express Data Manager TM (EDM) [2].

Solihin et al. (2017) stated that relying on the Raw IFC Raw data may lead to poor performance. Pauwels et al. (2011) discussed the limitations of using the IFC format in BIM systems in the AECO industry and proposed using semantic web technology, such as RDF graphs and ontologies, to overcome these limitations and enhance the functionality of BIM systems. A test case for acoustic performance to demonstrate the improvements of semantic web technology was presented and compared to traditional approaches.

Also, the IFC model is used by SMC as the information source, mainly focusing on 'design-spell-check'. However, SMC is restricted in its application to deal with a set of objects (Ding et al., 2006). Solihin et al. (2017) discussed the limitations of the current software applications. Except for SMC, Geometry and spatial relations are not covered. Currently, it is not applicable to check geometrical modelling requirements. Model View Definition (MVD) is limited only to check the existence of specific entities and/or attributes (Schwabe et al., 2019).

---

[1] Blackbox: In science, computing, and engineering, a black box is a system that can be viewed in terms of its inputs and outputs (or transfer characteristics) without any knowledge of its internal workings. Its implementation is "opaque" (black).

[2] Express Data Manager (EDM): is a software integration platform that supports interoperability of models defined by IFCs. It provides object-based rule bases and is an ideal platform for encoding and linking building codes with building models.

2.2 Information Requirements

Before undertaking a project, whether in the delivery or operational phase, it is imperative to contemplate the specification of information and physical assets. Information management ensures the appropriate delivery of information to its designated recipient, precisely when needed, to fulfil a particular objective. In this context, Information requirements encompass structured and unstructured information and define the inputs for the entire information management system (UK BIM Framework, 2020).

Information requirements can be likened to a skeletal framework, replete with various holes of varying shapes and sizes ecosystem (UK BIM Framework, 2020), as illustrated in Figure 6. These apertures distinctly articulate the precise demands for the necessary information to occupy them accurately. Consequently, information providers exchange information deliverables with the information receiver, who serves as the specifier, effectively filling these gaps in the framework.

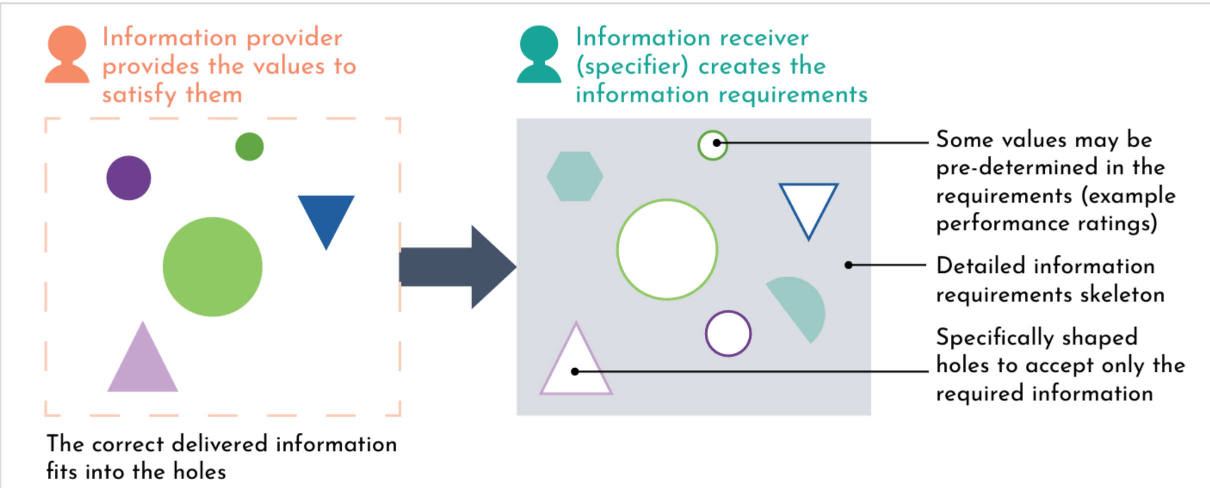


Figure 6: Information requirements skeleton
Resources: (UK BIM Framework, 2020)

2.2.1 Information Requirements Principles and Definitions

According to ([ISO 19650-1], 2018; [ISO 19650-2], 2018), Information requirements involve several resources, namely Organisational Information Requirements (OIR), Asset Information Requirements (AIR), and Project Information Requirements (PIR). These resources are the foundation for establishing appointment-level information requirements, commonly called exchange information requirements (EIR). Table 2 contains the definitions of previous terminologies and information model terminologies of the Asset Information Model (AIM) and Project Information Model (PIM). Figure 7 illustrates the different correlations between different types of information requirements and information models.

Table 2: Definitions of information requirements
Resource: ([ISO 19650-1], 2018)

	Abb.	Description
1	OIR	*"OIR explain the information needed to answer or inform high-level strategic objectives within the appointing party."*
2	AIR	*"AIR set out managerial, commercial, and technical aspects of producing asset information. The managerial and commercial aspects should include the information standard and the production methods and procedures to be implemented by the delivery team."*
3	PIR	*"PIR explains the information needed to answer or inform high-level strategic objectives within the appointing party in relation to built asset projects. PIR are identified from both the project management process and asset management process."*
4	EIR	*"EIR set out managerial, commercial, and technical aspects of producing project information. The managerial and commercial aspects should include the information standard and the production methods and procedures to be implemented by the delivery team."*
5	AIM	*"AIM supports the strategic and day-to-day asset management processes established by the appointing party. It can also provide information at the start of the project delivery process."*
6	PIM	*"PIM supports the delivery of the project and contributes to support asset management activities. PIM should also be stored to provide a long-term archive of the project and for auditing purposes."*

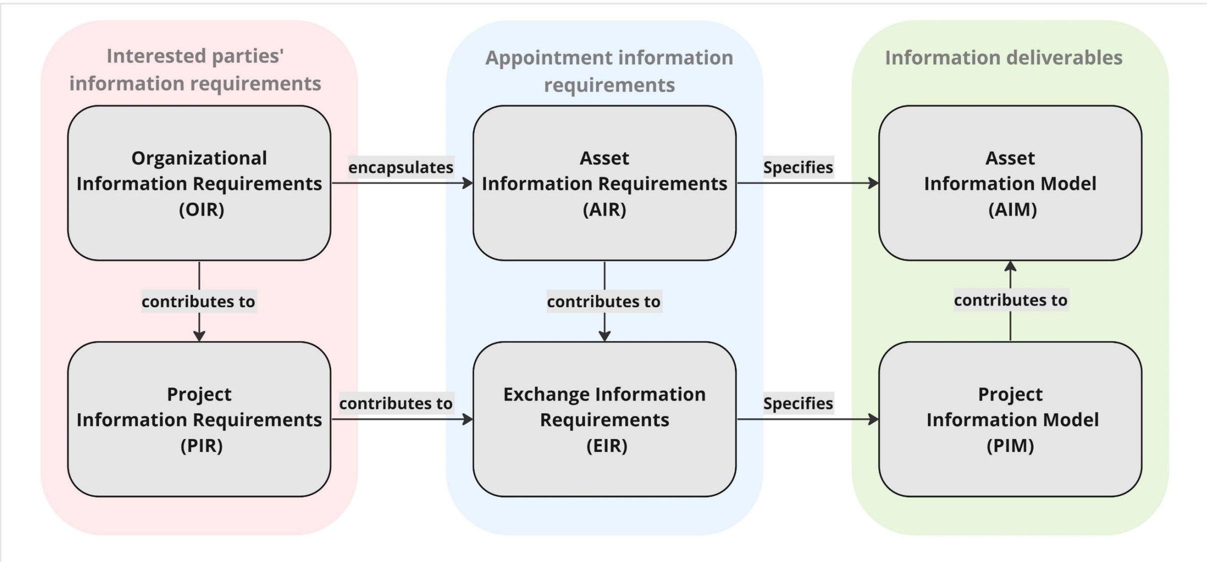


Figure 7: Hierarchy of information requirements
Resource: Data derived from ([ISO 19650-1], 2018)

It is advisable to structure information requirements consistently whenever feasible to enhance comprehension during information delivery and facilitate the establishment of automated checking rules. Employing structured tools like databases and spreadsheets can be instrumental in achieving this goal. A suggested method involves creating a master set of information requirements using rationalised purposes and the "Level of Information Need" [1] framework. Subsequently, these requirements can be filtered based on specific use cases and appointments. This approach streamlines the creation of appointment-specific information requirements while simultaneously eliminating redundancy and preventing information gaps in the delivery process.

2.2.2 Level of Information Need Framework

The level of Information Need is a framework that determines the scope and detail of information to be exchanged. It includes geometrical information, alphanumerical information, and documentation, as explained in Table 3. A combination of these types of information can fulfil this Level of Information Need. However, there can be inconsistencies and overlaps, which can create problems within the information model. To address this, a clear hierarchy of information containers should be established. The Level of Information Need can be predefined to the whole company process or project specific depending on the BIM use. If certain aspects are not relevant, they can be marked as "not applicable or N/A".

Also, the Level of Information Need specifies the required presence of geometrical, alphanumerical, and/or documentary information to meet specific purposes at information delivery milestones. Combining the Level of Information Need aspects is essential to meet all identified purposes.

Table 3: Definition of Level of Information Need and its sub-divisions
Resource: (*BSI Standards Publication Building Information Modelling-Level of Information Need*, 2020)

Division	Sub-division	Description
Geometrical Information	Detail	An aspect of geometrical information describes the complexity of the object's geometry compared to the real-world object. This is a continuum ranging from simplified to detailed. Also, the Level of Development framework (LOD) can be used to define the level of Details needed.
	Dimensionality	Several spatial dimensions characterise the object. Dimensionality can be zero-dimensional - 0D (location point), one-dimensional - 1D (e.g., line, curve, path), two-dimensional - 2D (e.g., surface, face) or three-dimensional - 3D (e.g., body, volume).
	Location	The position and orientation of an object. Location can be absolute, against a reference point, or relative, against another object.

[1] "Level of Information Need" should not be abbreviated, as stated in section 6.5.1 of ISO19650 Concepts and Principles Guidance.

Continued: Table 3: Definition of Level of Information Need and its sub-division

	Appearance	The visual representation of an object. This is a continuum ranging from symbolic to realistic compared to the real world. More refined Appearance can contain more shading attributes (e.g., diffuse colouring, transparency, reflectance, emissivity)
	Parametric behaviour	Whether or not the shape, position and orientation are created to remain dependent on other information associated with the object or the context into which the object is placed, allowing complete or partial reconfiguration.
Alphanumerical Information	Identification	Used to position an object within a breakdown structure. (e.g., Name, type name, classification, codification, reference structuring, index, numbering, etc.)
	Information content	All required properties: Properties can be grouped to facilitate the management of the alphanumerical information.
Documentation	Set of documents	The documentation for an object or set of objects to support processes, decisions, approvals, and verification of information deliverables should be specified as a set of required documents.

2.3 openBIM Standards

In 1994, Autodesk initiated its endeavours by inviting a consortium of companies to collaborate on developing C++ classes that would facilitate integrated application development. Originally known as the "Industry Alliance for Interoperability," it later transformed into the "International Alliance for Interoperability" in 1997 and ultimately underwent a name change to "buildingSMART" in 2005. Figure 4 demonstrates that buildingSMART standards encompass diverse process and information capabilities tailored to the built environment industry. The primary objective of buildingSMART is to create and uphold open international standards for BIM (openBIM). These standards are designed to enhance collaboration and interoperability within the industry. (buildingSMART, n.d.-a)

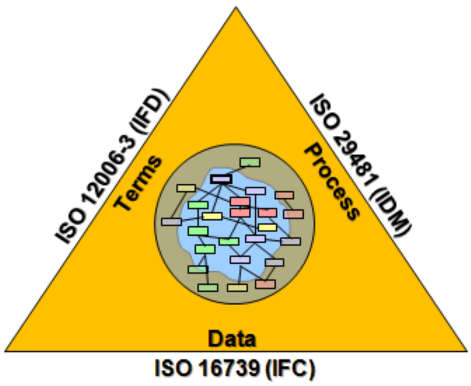


Figure 8: openBIM structure
Resource: (Laakso & Kiviniemi, 2012)

buildingSMART standards encompass a wide range of unique processes and information capabilities specific to the built environment industry. These include the Industry Foundation Classes (IFC), Information Delivery Manual (IDM), BIM Collaboration Format (BCF), and Information Delivery Specification (IDS). The standards facilitate interoperability and effective information exchange between software applications, define information requirements and deliverables, and provide structured formats for collaboration and issue resolution.

By promoting seamless information exchange, these standards enhance collaboration and efficiency in building projects. Below is a summary derived from the official website of buildingSMART International (buildingSMART, n.d.-a, n.d.-b, n.d.-c, n.d.-d). The research later uses the OpenBIM formats as a part of implemented workflows for Model-Checking.

2.3.1 Industry Foundation Classes (IFC)

IFC, short for "Industry Foundation Classes," is a standardised digital description of the built environment, encompassing structures like buildings and civil infrastructure. It adheres to an open, international standard (ISO 16739-1:2018), designed to be vendor-neutral and compatible with various hardware devices, software platforms, and interfaces for diverse applications. The specification of the IFC schema serves as the principal technical output of buildingSMART International, aligning with their objective of promoting openBIM®.

IFC is a standard format for sharing information between parties involved in specific business transactions. For instance, an architect may share a model of a new facility design with an owner, who can then send the building model to a contractor to request a bid. Subsequently, the contractor may provide the owner with an as-built model, including detailed information about installed equipment and manufacturer specifications. Additionally, IFC can be utilised for archiving project information incrementally throughout the design, procurement, and construction phases or as a comprehensive collection of "as-built" data for long-term preservation and operational purposes.

2.3.2 Information Delivery Manual (IDM)

The built asset industry requires effective communication among various organisations to work efficiently. This becomes more critical when using digital tools, as these tools have limited tolerance for interpreting digital data. buildingSMART developed the ISO 29481-1:2010 standard, "Building information modelling – Information delivery manual – Part 1: Methodology and format." These standards capture and specify processes and information flow throughout a facility's lifecycle.

The methodology can be used to document existing or new processes and describe the associated information that needs to be exchanged between parties. It is important to note that software support is crucial for implementing an information delivery manual.

The standard is accepted as an ISO standard, and efforts are underway to add more specific documentation of exchange scenarios and define stages in the communication process. While some IDM projects have led to successful specifications tested in real projects, challenges exist in areas lacking structured processes, necessitating the agreement on processes and exchange requirements. In such cases, IDM development must be followed by software development to achieve the expected results. Therefore, a clear strategy for implementing the IDM in software solutions is essential.

2.3.3 BIM Collaboration Format (BCF)

BIM Collaboration Format (BCF) is a communication format used in the built asset industry to facilitate collaboration and information exchange among BIM applications. It allows the sharing model-based issues using IFC models previously shared by project collaborators. BCF can be implemented through file exchange or a web service like a dedicated BCF server.

The format transfers XML data containing contextualised information about issues, referencing views, coordinates, and BIM elements. Solibri, Tekla, and iabi developed BCF to leverage open communication technology for IFC-based workflows. It has become an openBIM standard by buildingSMART International. BCF offers benefits in the design, procurement, construction, and operations phases, enabling activities like quality assurance, design coordination, procurement coordination, installation tracking, and handover documentation. By utilising open standards and bypassing proprietary formats, BCF enhances communication and improves workflows in the BIM industry.

2.3.4 Information Delivery Specification (IDS)

An Information Delivery Specification (IDS) is a machine-readable document that outlines the requirements for exchanging model-based information. It specifies how objects, classifications, properties, values, and units should be delivered and exchanged. This includes using Industry Foundation Classes (IFC), Domain Extensions, and additional classifications and properties, which can be stored in the bSDD or elsewhere. The IDS is used to determine the Level of Information Need.

It ensures validation of IFC for clients, modellers, and software tools, enabling automated analyses. The IDS is a core component that can be used as a contractual agreement to deliver accurate information. It allows for creating customised requirements based on specific projects and asset portfolios. The IDS provides a solution for reliable and predictable workflows when exchanging data.

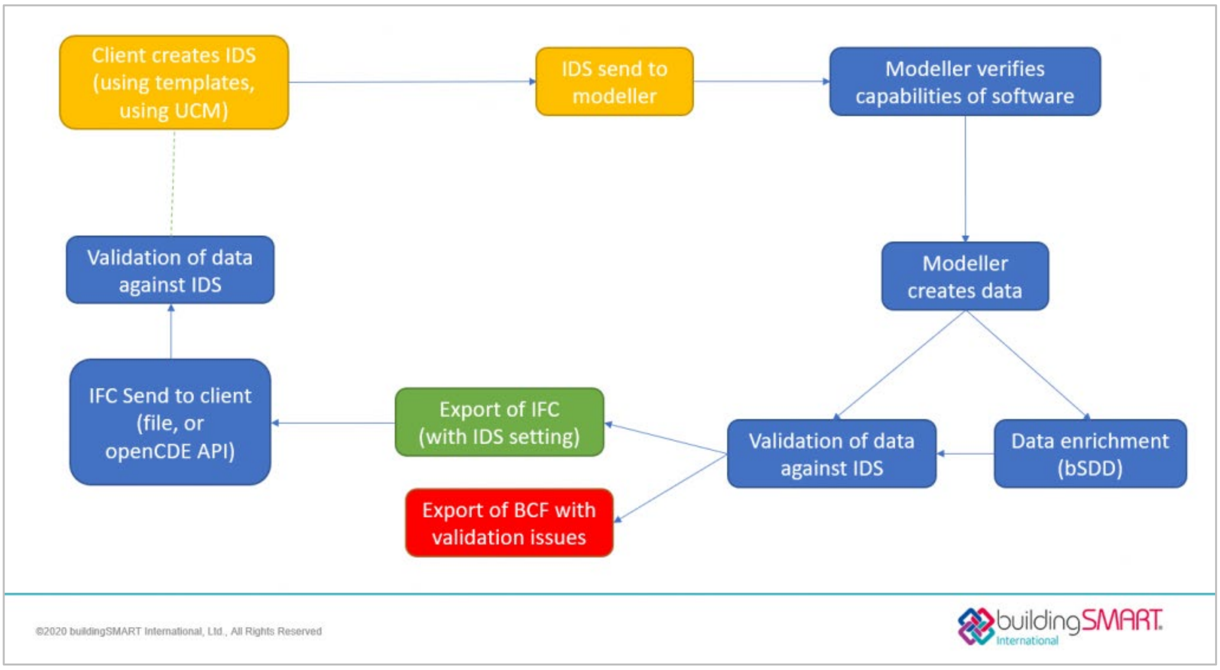


Figure 9: buildingSMART IDS workflow
Resource: (buildingSMART, n.d.-d)

3 METHODOLOGY

This chapter describes how the research has been conducted, allowing to assess the research's validity. The chapter explores the research approach and type, the methods used to collect data, the research constraints, and the tools used.

3.1 Research Approach and Type

In this research, the applied research approach is employed with a qualitative methodology to analyse different workflows derived from previous BMC case studies and projects in the AECO industry. The primary emphasis of this investigation is on identifying and checking the information requirements for different BIM uses during the project lifecycle.

As described in Figure 10, The research involves applying the knowledge or frameworks of BMC to real-world scenarios. The research focuses on developing an automation process for BMC, with a particular implementation on an actual project. This project serves as a platform to assess and analyse the proposed workflows, ultimately aiming to delineate the limitations of each workflow.

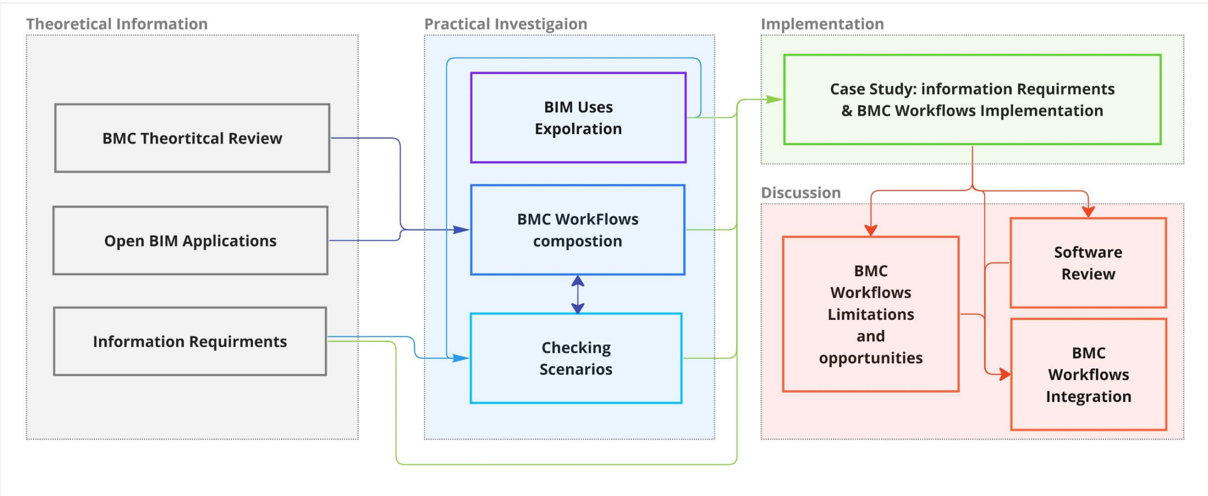


Figure 10: Research applied approach

3.2 Data Collection Methods

Using qualitative methods in this research facilitates in-depth understating of BMC implementation and realistic settings to develop optimised and automated BMC workflows. The methods that were used in the research data collection were as follows:

<u>A- Direct observations:</u>

The research incorporates the principles and applications of BMC, OpenBIM Standards and Information requirements as expounded in the extant literature review, Pg.5.

B- Documents and records:

A substantial portion of information utilised in this research is derived from international BIM standards, including ISO 19650-1:2018, ISO 19650-2:2018, and BS EN 17412-1:2020, along with documentation from buildingSMART and other BIM software resources.

C- Case Study:

In collaboration with Protim Ržišnik, this research comprehensively analyses the entire spectrum of case study data, encompassing the BIM model, extant quality control protocols, naming conventions, and company standards. Alongside project documentation, a set of monthly meetings during the whole research project were conducted with the company BIM Group manager to align the defined information requirements of the project with the company and to discuss and develop the case study implementation process.

3.3 Used Software/ Tools

As one of the research goals, a set of different software/Tools was used and explored to implement the BMC proposed workflows and determine the limitations of each software/tool. As indicated in Table 4, a set of software/tools was used.

Table 4: Software/Tools used in the research

Software/Tool	**Use**
Autodesk Revit	Employed as BIM modelling environment
Dynamo	VPL tool to implement the automated part of information modifications
Revit Lookup Add-in	Used to analyse Revit elements Information
Autodesk Interoperability Tools	Utilised to implement the core process of BMC – Workflow A
Diroots Add-In	for information enrichment and editing
Solibri Model Checker (SMC)	Used to implement the core process of BMC – Workflow B
Bexel Manager	Used to implement the core process of BMC – Workflow B
BIM Vision	Used as an IFC viewer
Blender	Used to implement part of BMC Process – Workflow C
usBIM.IDS Editor	Used to implement part of BMC Process – Workflow C
usBIM.IDS Validator	Used to implement part of BMC Process – Workflow C

4 BMC WORKFLOWS AND SCENARIOS

This chapter covers the comprehensive application of BMC throughout a project's lifecycle. First, it focuses on exploring various BIM uses, aiming to analyse the information exchange flow for each use. Consequently, The BMC traditional process is adapted, proposing three distinct BMC workflows. Before practically showcasing the implementation of proposed workflows, A set of typical scenarios of Information requirements checking is defined, shedding light on the practical utilisation of BMC within authentic, real-world settings, thus providing valuable insights into its tangible benefits and applications. Moreover, within the context of this chapter, one of these scenarios is dissected and applied through the proposed workflows, explaining the application of each workflow.

4.1 Investigated BIM Uses

BMC finds extensive applications across the AECO industry involving the methodical assessment and validation of BIM models to ensure accuracy, consistency, and compliance with design criteria, codes, regulations, and project requirements. Some key BIM uses include clash detection, code compliance, quantity take-off, cost estimation, scheduling, risk assessment, quality control, energy analysis and sustainability, legal and regulatory compliance, and design validation. The research investigates highlighted BIM uses, as presented in Table 5, to scrutinise the information requirements that could be inferred from each specific use/ application.

Table 5: Investigated BIM uses
Resource: Definitions were extracted from (Succar, 2019)

Code	Model use	BIM Dictionary Definition
4040	Clash Detection	*“A Model Use representing the use of 3D Models to coordinate different disciplines (e.g., structural and mechanical) and to identify/resolve possible clashes between virtual elements prior to actual construction or fabrication.”*
4050	Code Checking & Validation	*“A Model Use representing the process of inspecting a file, document, or BIM model for compliance against predefined specifications or established design, performance, or safety codes. Also, refer to Model Validation.”*
4070	Cost Estimation	*“A Model Use representing how 3D models are used to generate feasibility studies and compare different budgetary options.”*
4130	Quantity Take-off	*“A Model Use representing how 3D models are used to calculate the quantity of Furniture, Fixtures and Equipment or building materials for the purpose of generating Cost Estimates.”*

4.1.1 BIM-based Code Validation

Ensuring the BIM models adhere to relevant building codes, regulations, and standards. Jovanovic et al. (2022) explained the complexity of the Building Code Compliance Checking and its consumption to the time. Any errors found during construction will significantly affect the cost and time.

So, the imperative to delineate precise information requirements becomes evident within the code validation process. Any shortfall in information integrity results in a compromised code validation process. For example, focusing on fire regulations, Ismail et al. (2023) identified three types of clauses in fire safety regulations: declarative clauses, informative clauses, and remaining clauses. Declarative clauses provide specific requirements or instructions, while informative clauses contain ambiguous terminology or provide design options. The remaining clauses include explanations of terminology or general provisions (Ismail et al., 2023).

As illustrated in Figure 11, a rule is used from the Slovenian minimum technical requirements for the construction of residential buildings and apartments (*Website of the Legal Information System of Slovenia*, n.d.). The Rule states: *"Article 13 (room width): Rooms intended for living and sleeping are not narrower than 1.90 meters and, in the case of lighting only on the shorter side, not narrower than half of their length."*

In this case, the information requirements that shall be needed in the model include the alphanumerical information of room number and room name, while the rule validation will be through accessing the geometry of the walls representing the boundary room, using the available VPL inside the repository or externally using a checking software. Exporting the same model to IFC, the checking process can be implemented using different software and tools that will be explained in the following section.

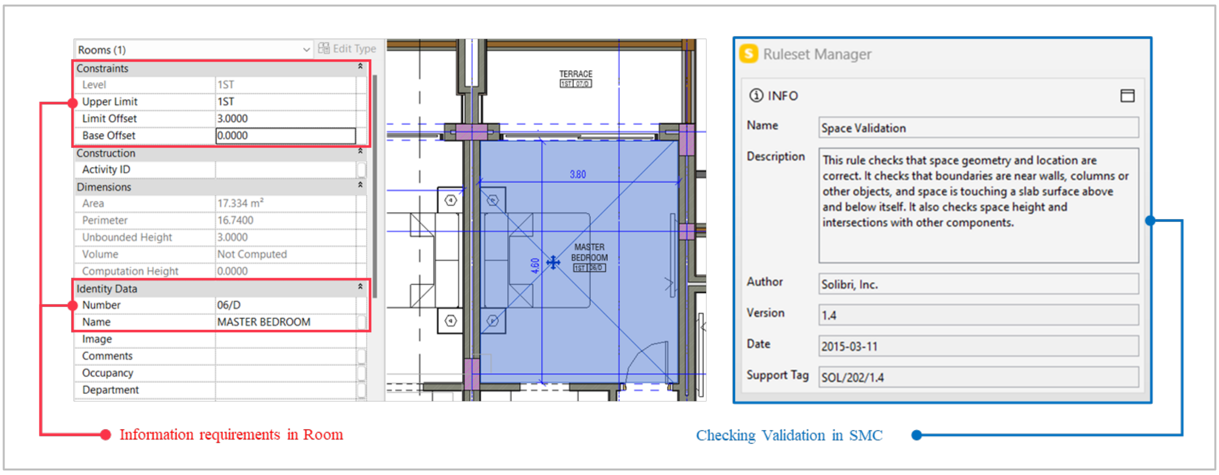


Figure 11: Code validation – Basic rule validation

4.1.2 BIM-based Quantity Take-off (QTO) Exploration

BIM-based Quantity Take-off (QTO) methods present an advancement over conventional methods reliant upon manual quantification from 2D drawings. These BIM-based methods and techniques introduce automation into the quantification process, enhancing its reliability and accuracy (Liu et al., 2022).

In this sub-section, a simplified staircase was modelled to explore the BIM-based quantities calculation methods. The same geometrical were exported to IFC and expected to have the exact base quantities. A review of used software was made to compare the quantities. During the lifecycle of the project, a set of quantities are needed. Following the Royal Institute of British Architects (RIBA) work plan, a set of quantities for the stair element are selected, as described in Table 6.

Table 6: Selected quantities for stair element

Phase	Required Quantity	Value Type	Notes
Strategic Definition	-	-	
Preparation Brief	-	-	
Conceptual Phase	Count of Landings	Integer	
	Count of flights	Integer	
Design Development	Stair Gross Volume	Double	
	Stair Net Volume	Double	
Technical Design	Number of Risers	Integer	
	Number of Treads	Integer	
	Riser Height	Double	
	Tread Length	Double	
Construction	Landing Thickness	Double	
	Landing Area	Double	
	Landing Volume	Double	
	Stair Flight Net Volume	Double	Used to quantify the materials needed
	Outer Surface Area	Double	Used to quantify the formwork needed
Handover and closing	-	-	
In use	-	-	

Reviewing other needed quantities, it is concluded that some required quantities can be derived from `IfcStairflight`, `IfcSlab` and `IfcRailing` and `IfcStair` Entities. Needed Information can be abstracted from Propertysets of `Pset_StairCommon`, `Pset_StairflightCommon`, `Qto_StairflightBaseQuantities` and `Qto_SlabBaseQuantities`, as shown in Figure 12. However, Some Quantities could be derived from the geometrical model and calculated. Some software, such as SMC and Bexel Manager, recalculate the quantities based on the geometrical model. However, by comparing the quantities, some differences were reported.

For instance, SMC calculated a volume of 3.05 m3, while Bexel Manager calculated a volume of 2.918 m3, as shown in Figure 13. SMC calculates the total volume, including the railing Volume, while Bexel Manager calculates the volumes for the stair and the railing Separately. Also, the surface area was calculated by Bexel Manager, while only the Bottom Area was calculated by SMC.

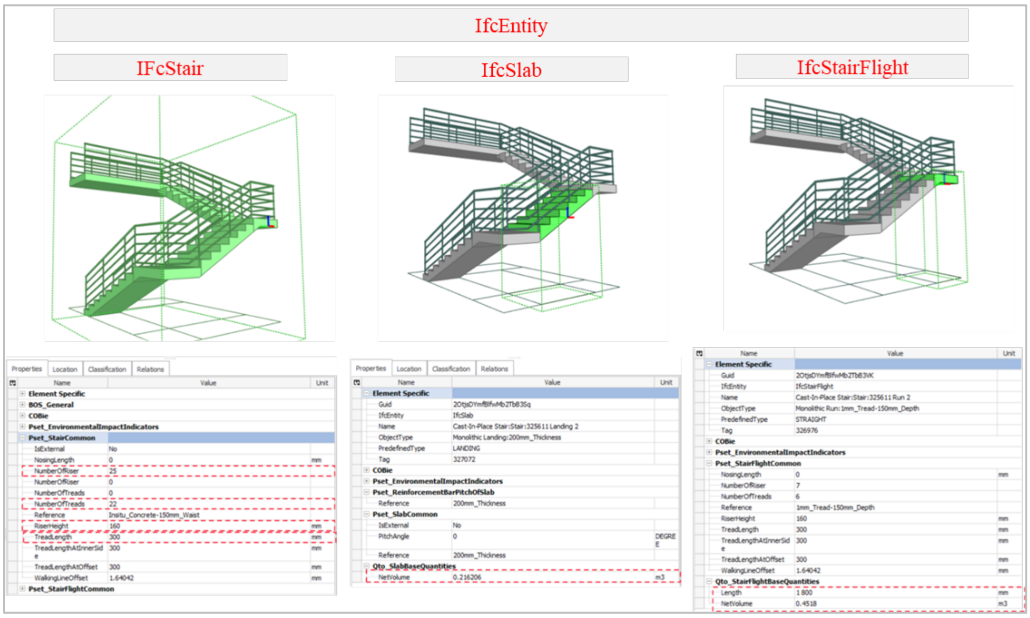


Figure 12: Stair element quantities

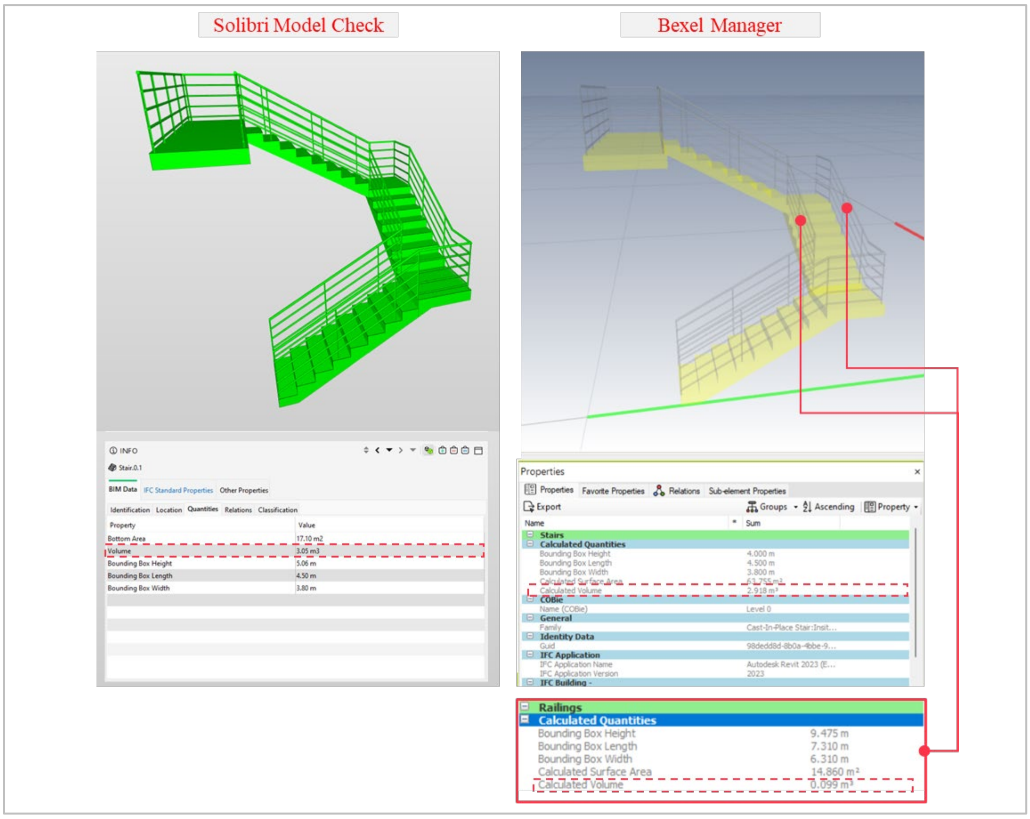


Figure 13: Comparison of stair calculated quantities in SMC & Bexel Manager

Another example is the architectural curtain wall quantification of curtain wall as demonstrated in Table 7. a set of properties was defined as follows:

Table 7: Selected quantities for the curtain system

Phase	Required Quantity	Value Type	Notes
Strategic Definition	-	-	
Preparation Brief	-	-	
Conceptual Phase	Curtain Wall Area	Double	
Design Development	Count of Glass Panels	Double	
	Glass Panel Area	Double	
Technical Design	Mullions Surface Area	Integer	
Construction	Mullions Cross Section Area	Double	
	Mullions Length	Double	
Handover and closing	-	-	
In use	-	-	

As mentioned, some basic quantities can be derived for the `Qto Pset`, as shown in Figure 14. For the calculated quantities from the geometrical model, each software creates its own calculated quantities for the overall Curtain Wall with an Area of 32.40m2. In contrast, Bexel Manager calculated the Surface Area for each Panel with a total Surface Area of 63.187 m2 and Mullions with a total Surface Area of 13.251 m2. Other Calculated quantities can be derived as shown in Figure 15.

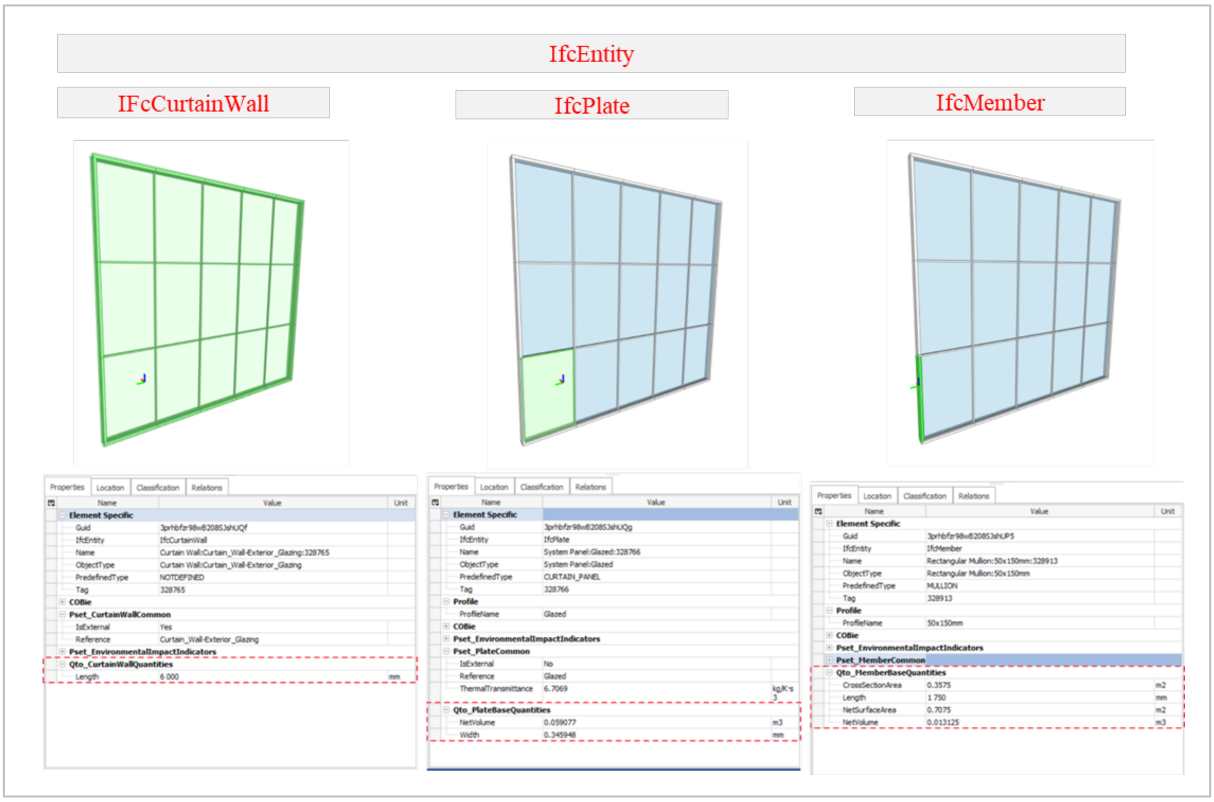


Figure 14: Curtain system quantities

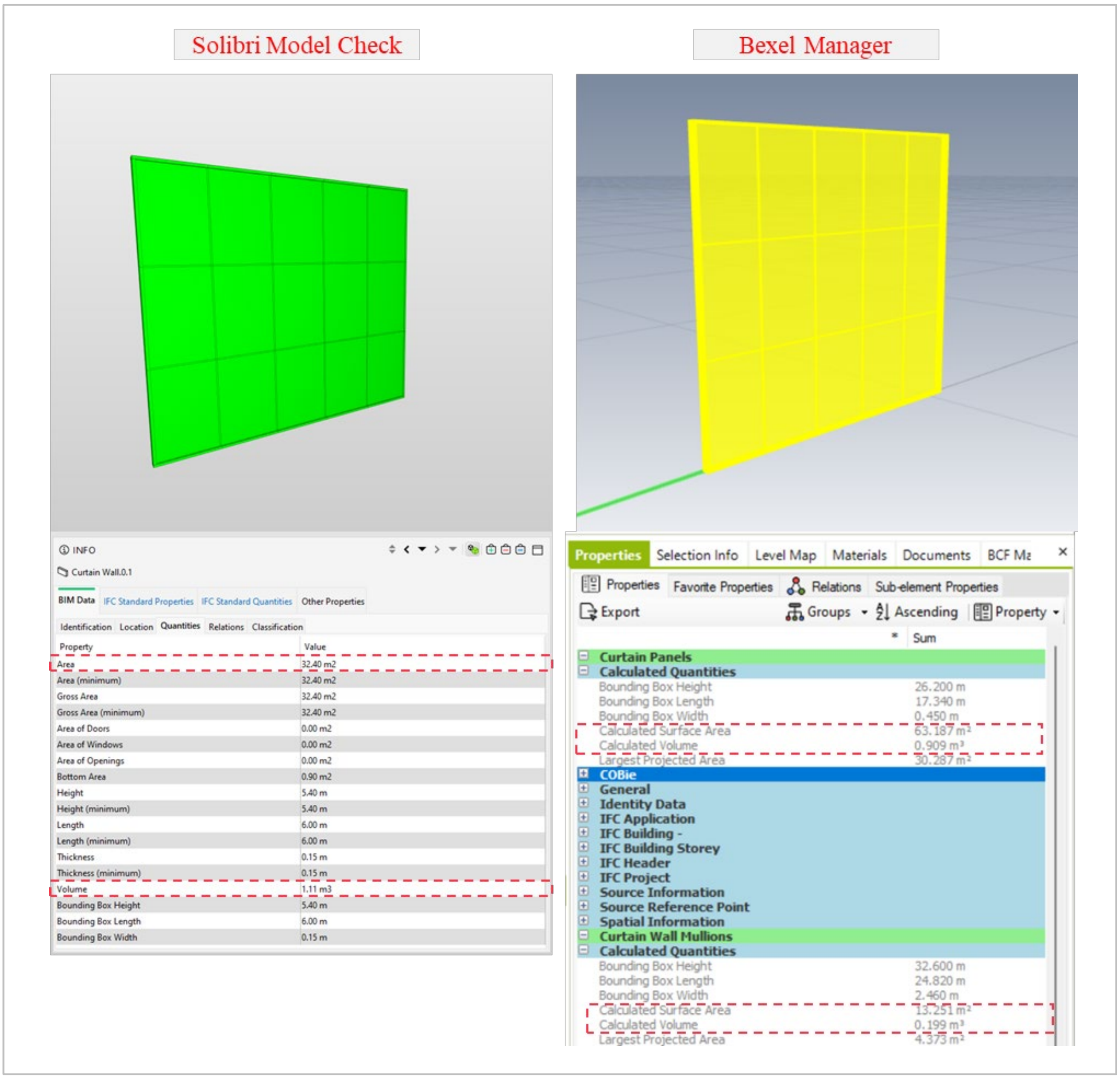


Figure 15: Comparison of curtain system calculated quantities in SMC & Bexel Manager

4.1.3 BIM-based Cost Estimation Exploration

The cost estimation process for each activity within a project's Work Breakdown Structure (WBS) is contingent upon aggregating required materials and other required resources such as labour and equipment. Therefore, the cost estimation process generally consists of three parts: (1) building materials classification; (2) Measurement of the amount of materials, labour, and equipment involved in completing the project; and (3) Calculating the total project cost (Fazeli et al., 2021; Jadid & Idrees, 2007; Ma et al., 2013).

Nevertheless, the process of BIM-based cost estimation might not be more straightforward, primarily due to the complex relationships between BIM Model Elements and How they are represented in the Bill of Quantities (BOQ).

These relationships can manifest in various ways, such as One-to-One, One-to-Many, or Many-to-One, as illustrated in Figure 16, depending on specific modelling criteria and project cost estimation requirements. To acquire a deeper comprehension of this matter, illustrative examples may be simulated, as demonstrated in the following examples:

Example 1: The brick wall and finishing wall can be modelled separately, each representing an individual BOQ Time (One-to-One). If both walls are modelled as a compound Wall, the Model Element encompasses more than one BOQ Item (One-to-Many).

Example 2: Certain projects employ the Lump Sum approach for Cost Estimation. Under this method, groups of model elements are consolidated into a single BOQ Item, streamlining the estimation process (Many-to-One).

Example 3: There are different practices in estimating the cost of a curtain wall in various projects. In some cases, the curtain wall is treated as a unified Element comprising Doors, Windows, Glazed Panels, and Mullions. Conversely, in other instances, individual Sub-Elements are used as separate BOQ Items, obviating the need to quantify the Curtain Wall System as a whole.

Indeed, each project's varying complexities and unique characteristics necessitate adapting information requirements to meet specific project needs.

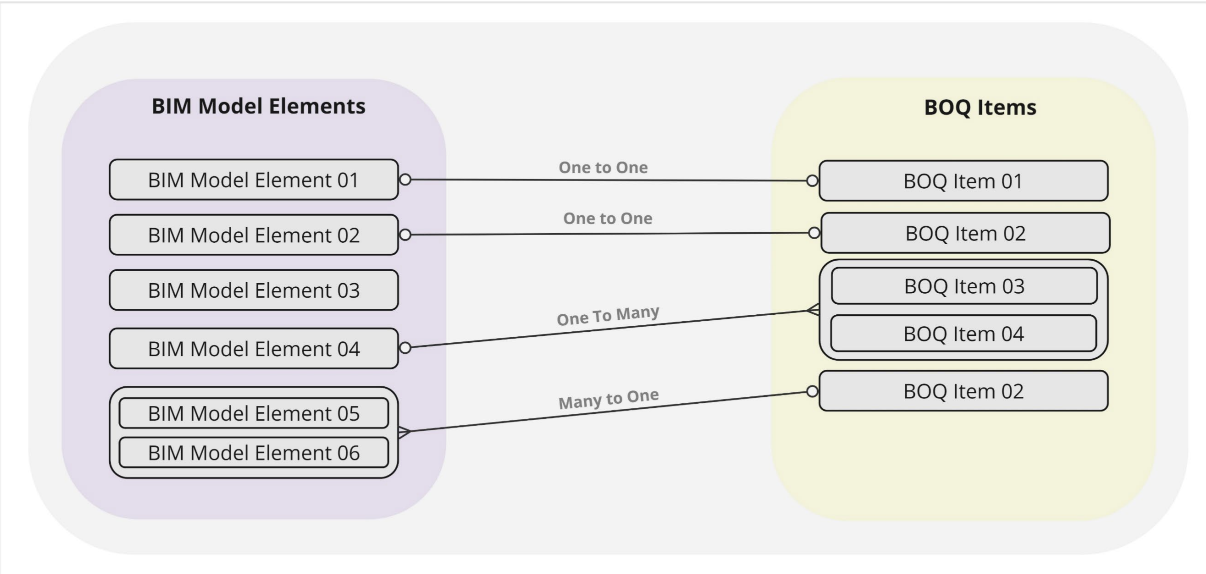


Figure 16: Relation between BIM model elements and BOQ items

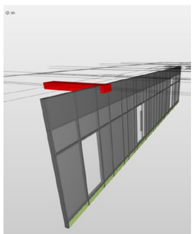
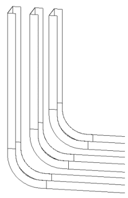
4.1.4 BIM-based Clash Detection

According to ([UNI 11337-5], 2017), Clash Detection is defined as: *"An analysis of the possible interferences between objects, models and outputs with respect to others"*. The clash detection process begins with model preparation, where BIM models representing architectural, structural, mechanical, electrical, and plumbing elements are created or obtained. Then, a clash matrix, a strategic tool used in clash detection processes to systematically track and manage clashes, shall be defined.

The clash matrix defines the coordination levels and the clash tests that shall be conducted. For example, level one of coordination checks the architectural elements' clashes with each other, while level two of coordination defines clashes between architectural elements and structural elements. Clashes can be classified into three main hard clashes, soft clashes, and workflow clashes, as described in Table 8.

Clash detection software then defines rules and criteria for potential conflicts. The software rigorously scans the models, flagging clashes where elements intersect or occupy the same space. Generated clash reports provide detailed information about clash locations and involved elements. Project teams prioritise and resolve clashes using these reports, often necessitating design adjustments or element repositioning.

Table 8: Type of clashes

Clash type	Description	Example
Hard Clashes	Hard Clashes are serious physical conflicts between building components or systems in the BIM model. They can lead to construction problems, delays, or rework, such as ductwork blocking structural beams or plumbing pipes conflicting with electrical conduits.	Structural Beam vs. Curtain Wall 
Soft Clashes	Soft Clashes are minor design or coordination issues that do not result in physical conflicts but require attention. They include design overlaps, minor adjustments, or aesthetic conflicts.	Cable trays are slightly overlapping. 
Workflow Clashes	Workflow clashes are related to project processes and coordination. They do not involve physical conflicts but can impact project efficiency and collaboration.	An HVAC maintenance schedule does not align with the scheduled delivery of spare parts.

Numerous software applications are employed for clash detection, each offering distinct functionalities. For instance, Autodesk Revit can perform clash detection, but it primarily focuses on identifying hard clashes. Users have limited flexibility to specify different clash types. Custom selections are unsupported, and clash tests can only be categorised broadly, such as wall versus column clashes. Additionally, Autodesk Revit cannot fine-tune tolerances, resulting in excessive soft clashes that may not be relevant.

In contrast, other dedicated clash detection software options, such as Autodesk Navisworks, BIMcollab, Bexel Manager, and SMC, provide a broader range of functionalities. These include advanced selection sets, customisable tolerances, and extensive filtration options."

4.2 BMC Proposed Workflows

The BMC process is adapted to extend beyond the conventional approach (Explained in Figure 5, Pg. 10) by integrating automated solutions for selected issues to mitigate the occurrence of commonplace errors inherent during the modelling and information enrichment processes, as illustrated in Figure 17. These solutions will be implemented in Automated and semi-automated codes that fix information inconsistency, add missing information, and report some metadata of the elements reported for taking the proper action.

As an Input to start the BMC process, A set of textual rules is defined and then transformed into computable rules using the RASE model-checking method. Also, the Level of Information Need is another input for the first stage of rule Interpretation, where all rules and information requirements will be translated into computable languages following the subsequent stages of the process.

Three distinct BMC workflows are proposed for the execution of the checking process. The proposed BMC workflows aim to optimise the checking process by integrating automated solutions and addressing common errors encountered during modelling and information enrichment. These workflows are based of the Adapted BMC process.

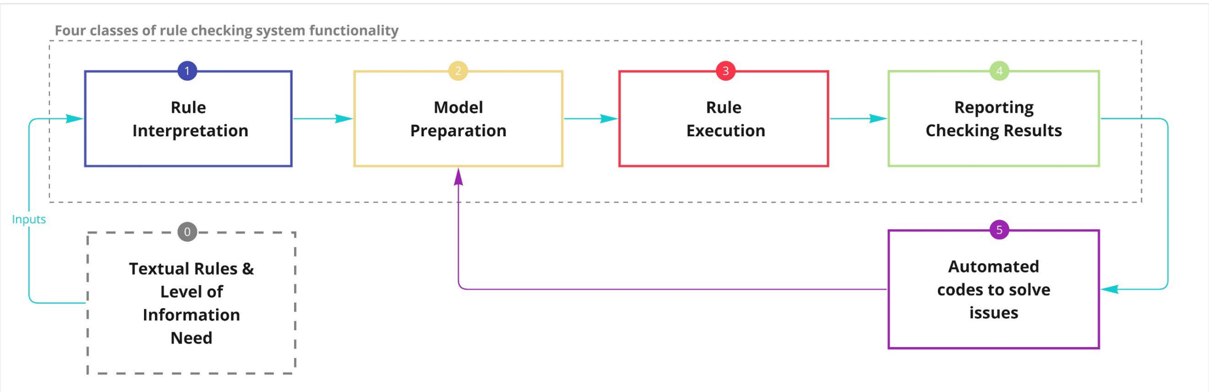


Figure 17: Adapted BMC process

4.2.1 Workflow A - Inside Repository Environment

This workflow aims to perform the entire process of BMC within a designated repository environment. The model preparation and information enrichment processes are taking place inside the repository, establishing an integrated framework for executing the entire process, contingent upon the repository's incorporation of selected BMC functionalities. This workflow significantly expedites direct results mapping, enabling efficient manipulation and editing without necessitating recurrent model exchanges for each checking iteration.

The research employs Autodesk Revit as a modelling environment, augmented by including Autodesk Interoperability Tools to optimise model checking procedures. Furthermore, Dynamo is used to automate some common errors that might be resulted during the modelling or information enrichment processes. Additionally, it might be necessary to use VPL, such as Dynamo or a programming language, such as Python, for specific rules interpretation and execution to access the API Revit parameters.

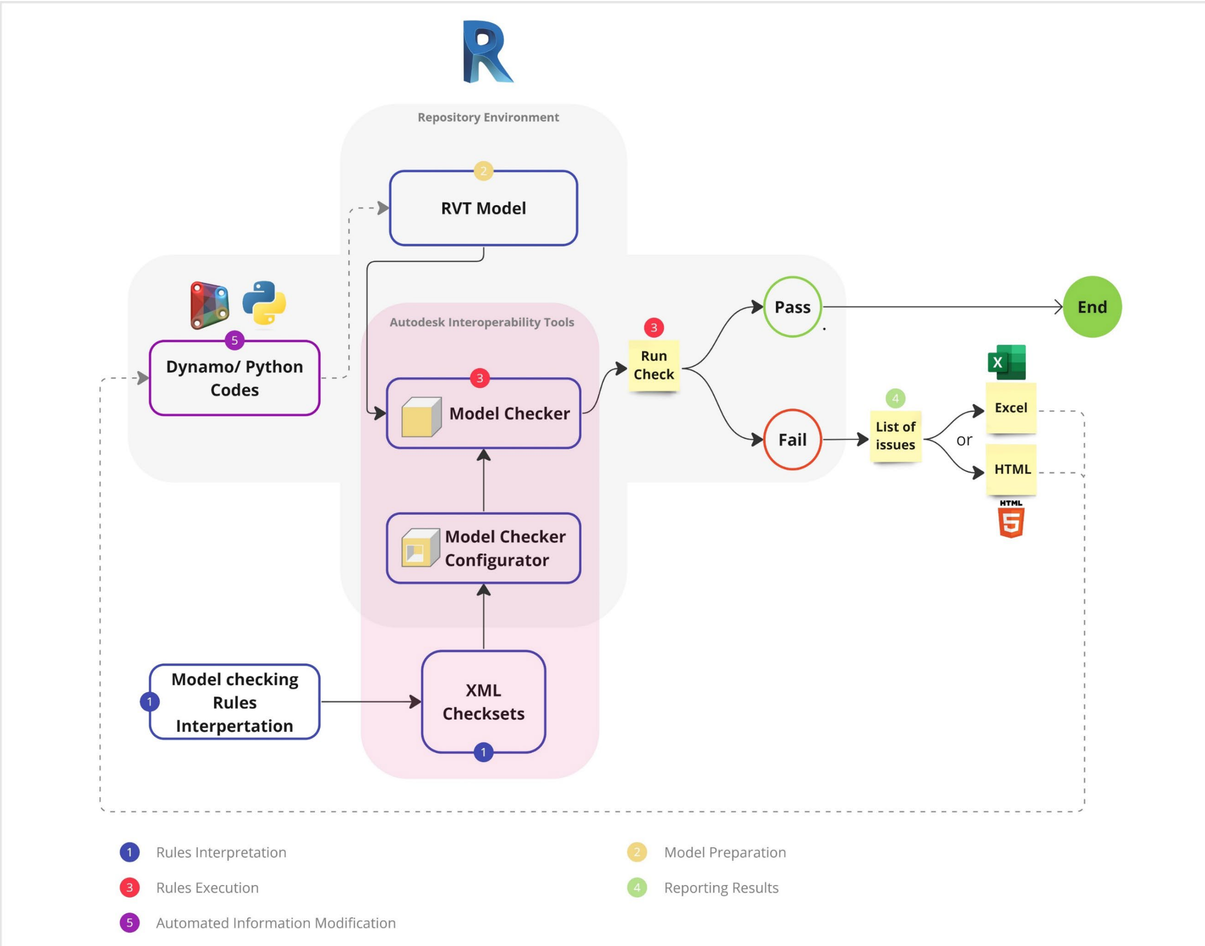


Figure 18: Workflow A - Inside the repository environment

4.2.2 Workflow B - Standalone Software

This workflow utilises dedicated standalone software tailored for BMC implementation or, alternatively, employs supporting tools, plugins, or add-ins specifically designed to complement BMC procedures. The initial phases of model preparation and information enrichment are seamlessly executed within the primary repository environment, with subsequent exportation to exchange formats such as IFC. Following this transfer, the subsequent stages of the process are seamlessly conducted within the designated software until the desired results are generated. Furthermore, the efficient exchange of identified issues is facilitated through the utilisation of BCFs, ensuring seamless integration of issue management with the main repository's workflow. Additionally, Excel or HTML reports can be generated for reporting issues.

Common software is employed for BMC, such as SMC, which has pre-defined and customisable rules. Furthermore, software applications such as Navisworks, Solibri, Tekla BIMsight, BIMcollab, and Bexel Manager similarly allocate specific utilities aimed to facilitate the BMC procedure.

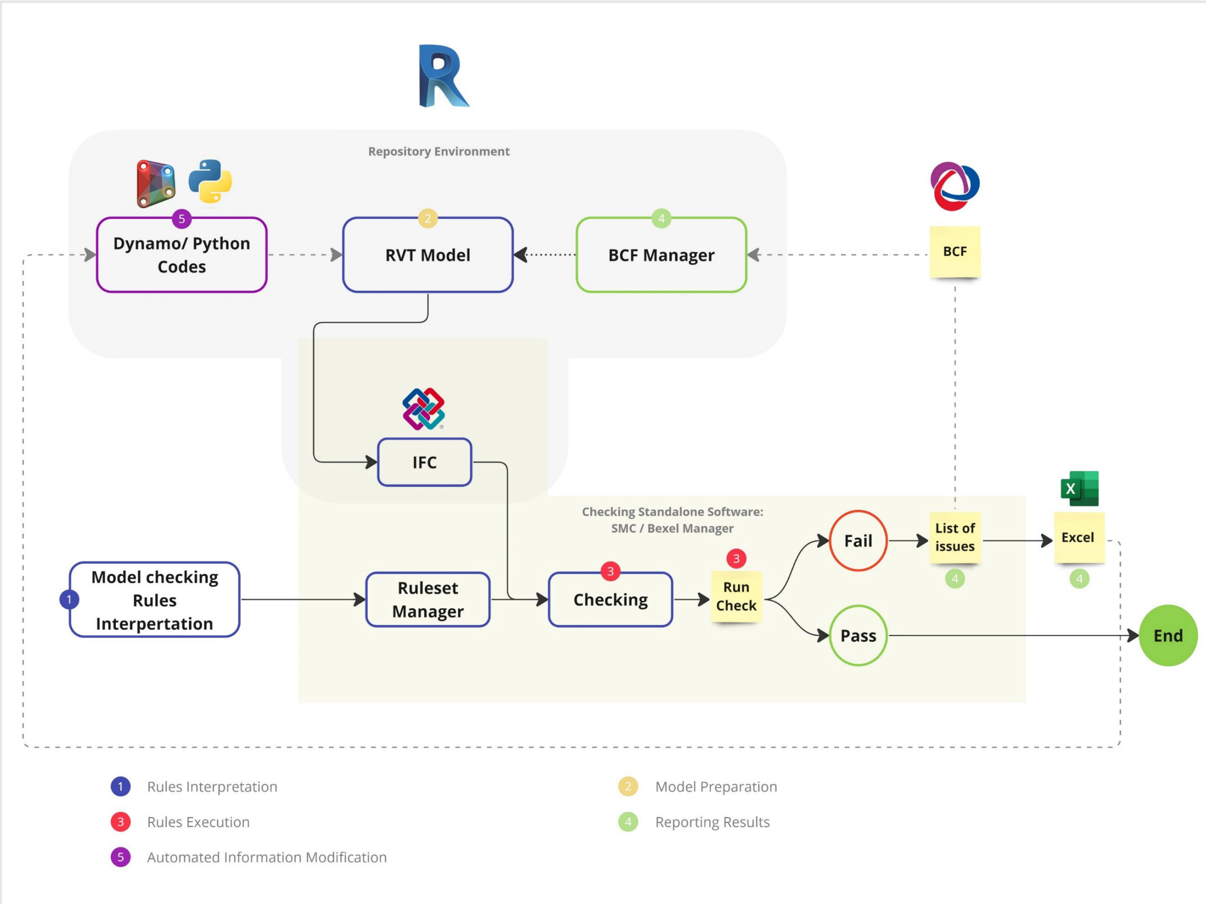


Figure 19: Workflow B - Standalone software

4.2.3 Workflow C - openBIM

In contrast to the previous workflow that incorporated openBIM formats such as IFC and BCF selectively, this workflow aims to utilise openBIM throughout the entire spectrum of BMC. This methodology commences its application right from the initial phase of rule interpretation, facilitated by using IDS. The incorporation of IFC and BCF formats remains integral for the seamless exchange of information with the main repository.

Significantly, this approach deviates from the necessity to rely solely on proprietary software solutions. This deviation is enabled by IDS, which is structured in XML format, allowing for its independent creation and customisation. The execution of rules can be accomplished through any software or tool that supports IDS, thus encapsulating a versatile and adaptable framework for rule execution.

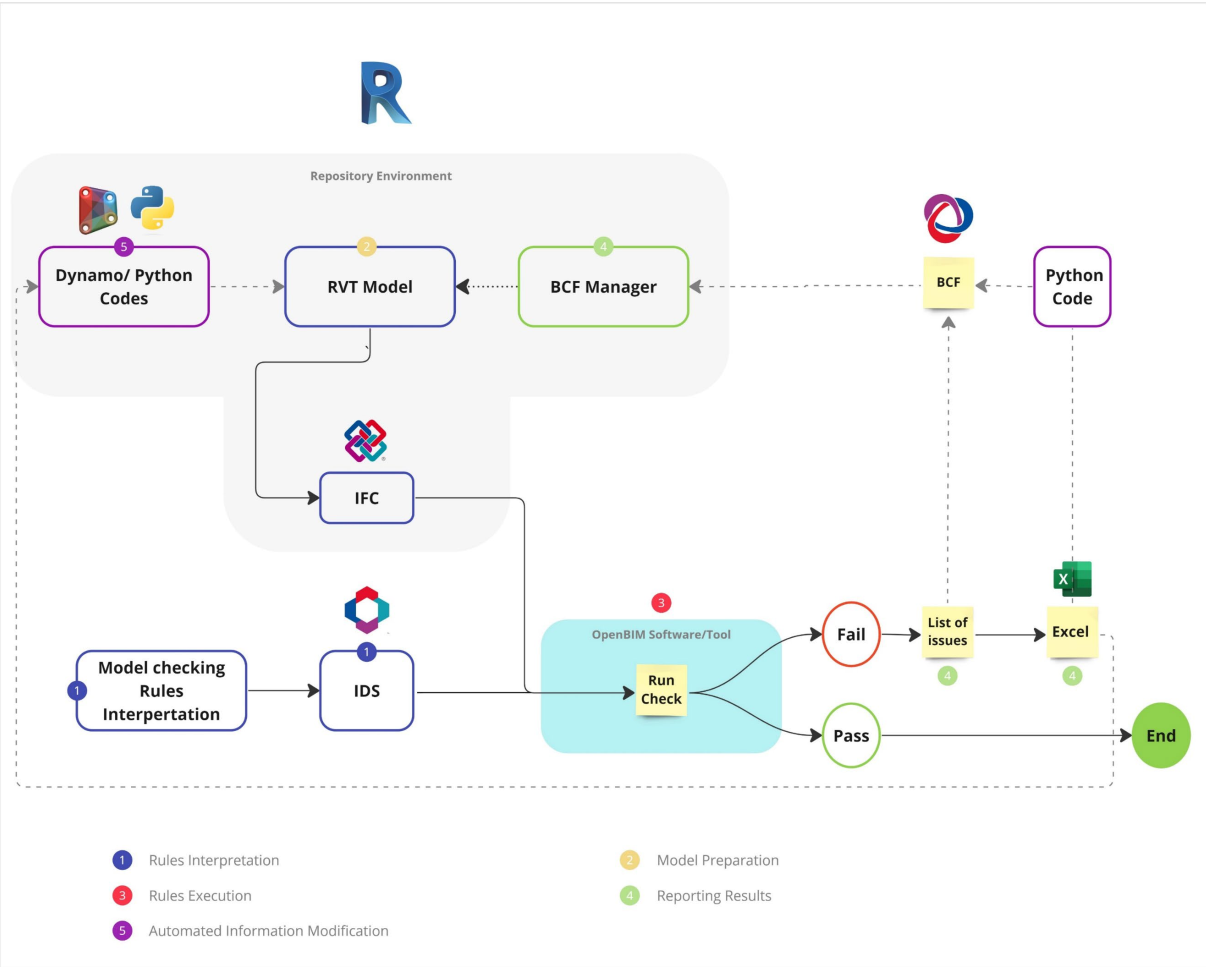


Figure 20: Workflow C - openBIM

4.3 BMC Scenarios

During the project's life cycle, each element in the BIM Model must be geometrically represented in a particular representation and contain specific properties with certain values according to the information requirements for each phase. Throughout the analysis of most common practices, a set of scenarios is created to analyse the information requirements as input for the whole BMC process. Some scenarios are focused on the alphanumerical information requirements with an exploration of possible scenarios of geometrical information requirements.

4.3.1 Scenario 1 – Property Existence and Value

This scenario, as illustrated in Figure 21, involves a series of checks to validate properties based on specified information requirements. Firstly, the existence of each property is verified. Subsequently, each property is checked to ensure it is in the correct property set. Each property's data type is assessed, including categorising properties as String, Integer, Double, or Boolean. Each property is checked to validate if it has value or not. Additionally, Prefixes, Subfixes and regular expressions are employed to ensure the correctness of property values (a detailed explanation of the regular expressions implementation is provided in the case Study Chapter, Pg. 49).

It should be noted that calculated values derived from geometry may deviate from this rule, necessitating the application of additional geometric checks. For example, the Area property can be checked to ensure it contains a value, but further scrutiny is required to ascertain the accuracy of that value.

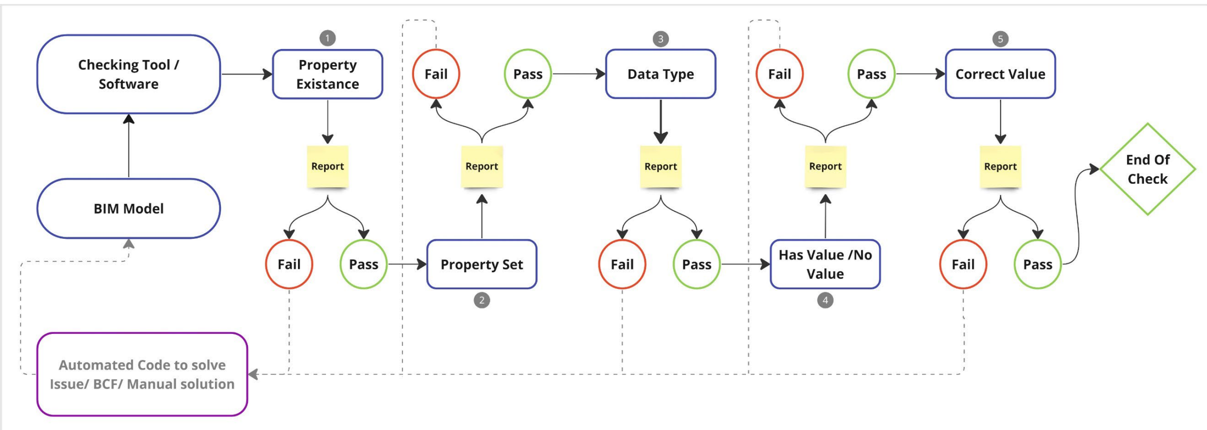


Figure 21: BMC Scenario 1 -Property check

4.3.2 Scenario 2 – Duplicated Property

As illustrated in Figure 22, each property is examined in this scenario to determine whether it exhibits duplication. The checking process concludes if no duplication is detected. However, if duplication is identified, a subsequent sub-check is conducted to group the duplicated properties based on their values.

If the duplicated property shares both the same value and property set, it is considered an error, and one of the properties is removed. On the other hand, if the duplicated property exhibits different values, it is logged in the check results, prompting the need for corrective action to modify the name of one of the duplicated properties.

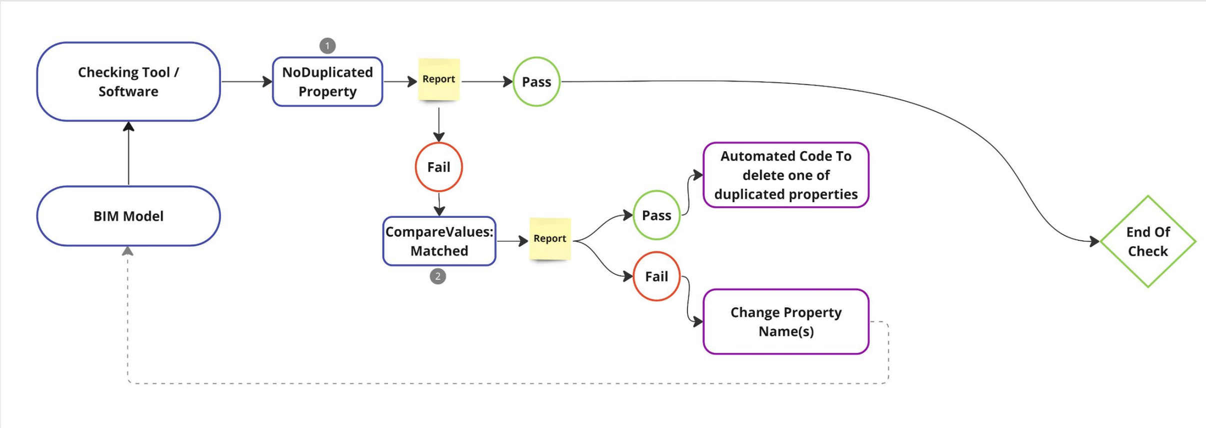


Figure 22: BMC Scenario 2 – Duplicated property check

4.3.3 Scenario 3 - Duplicated Element

In this scenario, as illustrated in Figure 23, the primary objective is to conduct a comprehensive assessment to identify and report any instances of duplicated elements. Furthermore, consideration is given to automating the process by implementing mechanisms to delete duplicated elements, particularly when they exhibit identical geometric and non-geometric attributes. However, careful examination is necessary to ensure the element's retention with the requisite properties while eliminating its duplicated counterpart.

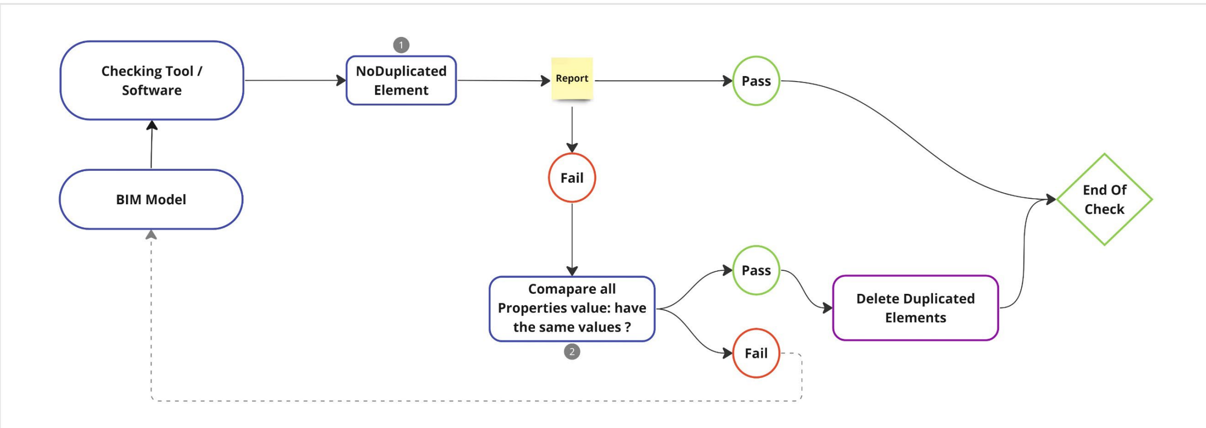


Figure 23: BMC Scenario 3 – Duplicated element geometrically

4.4 Applying Scenarios within Workflow Frameworks

To explain the checking process, this section applies scenario 1 on a simplified model for procedural examination. The objective is to use this scenario for specific properties and describe how each step can be implemented within the corresponding workflow, ensuring alignment with established practices and methodologies.

A simplified model, as shown in Figure 24, focuses on structural Column category and NBS BIM Object Standard implementation. Later in the thesis case study, the same application methodology is used for all categories, with other references needed. It is observed that each element has a unique GUID, and each property and property set have unique GUIDs. Understanding how the information is structured, using Revit Lookup Add-in in the modelling environment, facilitates the process of checking rules computable translation and the automation process for solving the issues.

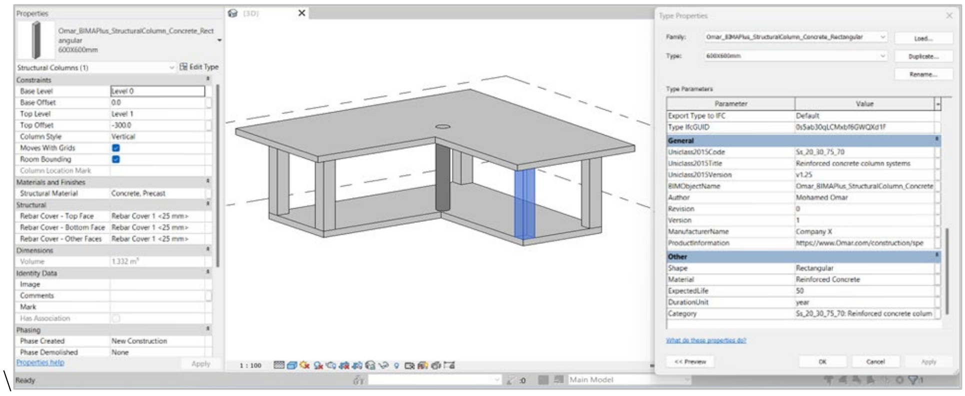


Figure 24: Simplified element with selected properties

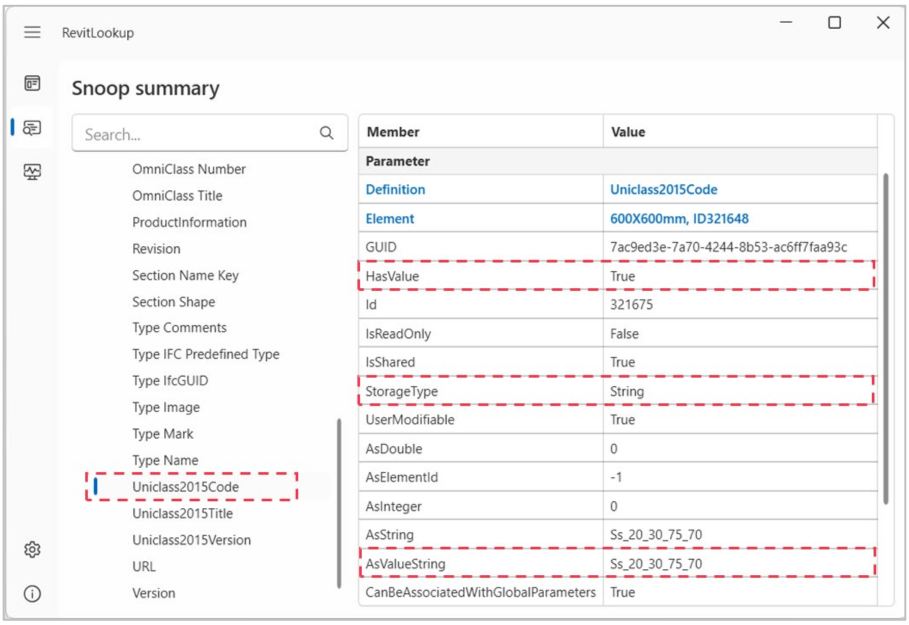


Figure 25: Revit Lookup Addin- Exploration of Unicalss2015code (Type Parameter)

4.4.1 Scenario 1 Application – Workflow A

First, the check is implemented inside the Autodesk Revit repository using the Autodesk Interoperability tool. In this Check, A list of properties is checked for the structural columns category to check the properties' existence, as shown in Figure 26. The check set of Autodesk Interoperability tools is XML format, which can be edited in Autodesk Model Checking Configurator for Revit or Edited as an XML in any XML editors or word processing and spreadsheet programs, as shown in Figure 27.

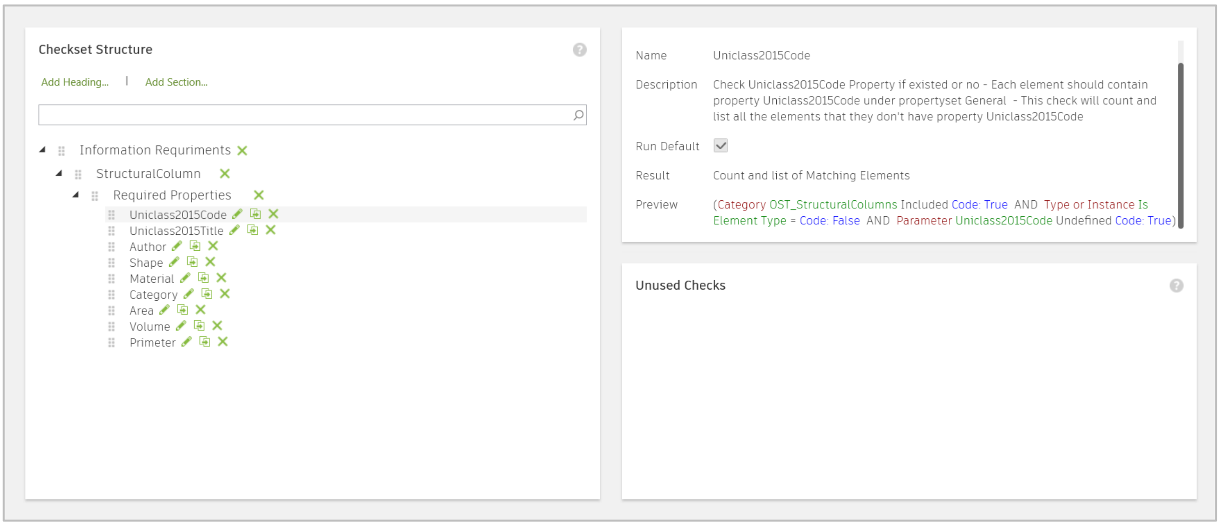


Figure 26: Autodesk Interoperability Tools - Check exitance of property

```xml
</Heading>
<Heading ID="883ccbcb-5bf9-4f0d-b4c3-7f726f888844" HeadingText="Information Requriments"
Description="An Example of checking information requirments " IsChecked="True">
  <Section ID="6a5283de-e036-4f6a-8f8f-017fee4883cc" SectionName="StructuralColumn" Title="" IsChecked="True" Description="">
    <Section ID="95579b90-f567-451f-8d7e-5f8da3c09d41" SectionName="Required Properties " Title="" IsChecked="True"
    Description="">
      <Check ID="8b6c1709-c838-4907-b527-1814cbac2010" CheckName="Uniclass2015Code"
      Description="Check Uniclass2015Code Property if existed or no - Each element should contain property Uniclass2015Code
      under propertyset General " FailureMessage="Uniclass2015Code Property is not existed. Add Unicalss2015code Property"
      ResultCondition="FailMatchingElements" CheckType="Custom" IsRequired="True" IsChecked="True">
        <Filter ID="37b8d2b4-7ae5-431b-a610-5b7083508063" Operator="And" Category="Category" Property="OST_StructuralColumns"
        Condition="Included" Value="True" CaseInsensitive="False" Unit="None" UnitClass="None" FieldTitle="" UserDefined="False"
        Validation="None" />
        <Filter ID="1a12949a-d7aa-4f01-b00c-ebd43767e97b" Operator="And" Category="TypeOrInstance" Property="Is Element Type"
        Condition="Equal" Value="False" CaseInsensitive="False" Unit="None" UnitClass="None" FieldTitle="" UserDefined="False"
        Validation="None" />
        <Filter ID="41d4e1a2-98b2-484f-89e8-fac23f21154a" Operator="And" Category="Parameter" Property="Uniclass2015Code"
        Condition="Undefined" Value="True" CaseInsensitive="False" Unit="None" UnitClass="None" FieldTitle="" UserDefined="False"
        Validation="None" />
      </Check>
```


Figure 27: Autodesk Interoperability Tools – Editing checks in XML

Using the Check Set Created, Model Check Run would be used for the rule execution. First, the check gave an overview of the percentage, as illustrated in Figure 28. Then, A detailed report containing the Element ID for each Check, as shown in Figure 29, is generated and shows whether it passes or fails the check.

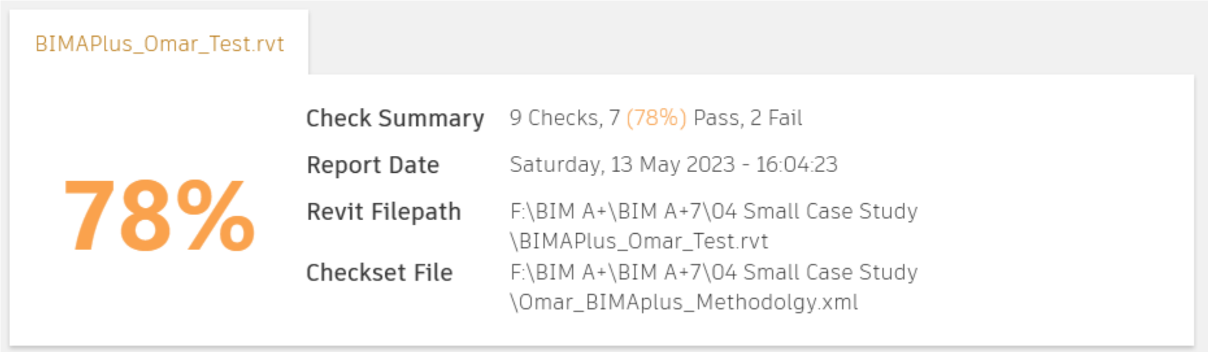


Figure 28: Autodesk Interoperability Tools - Check summary

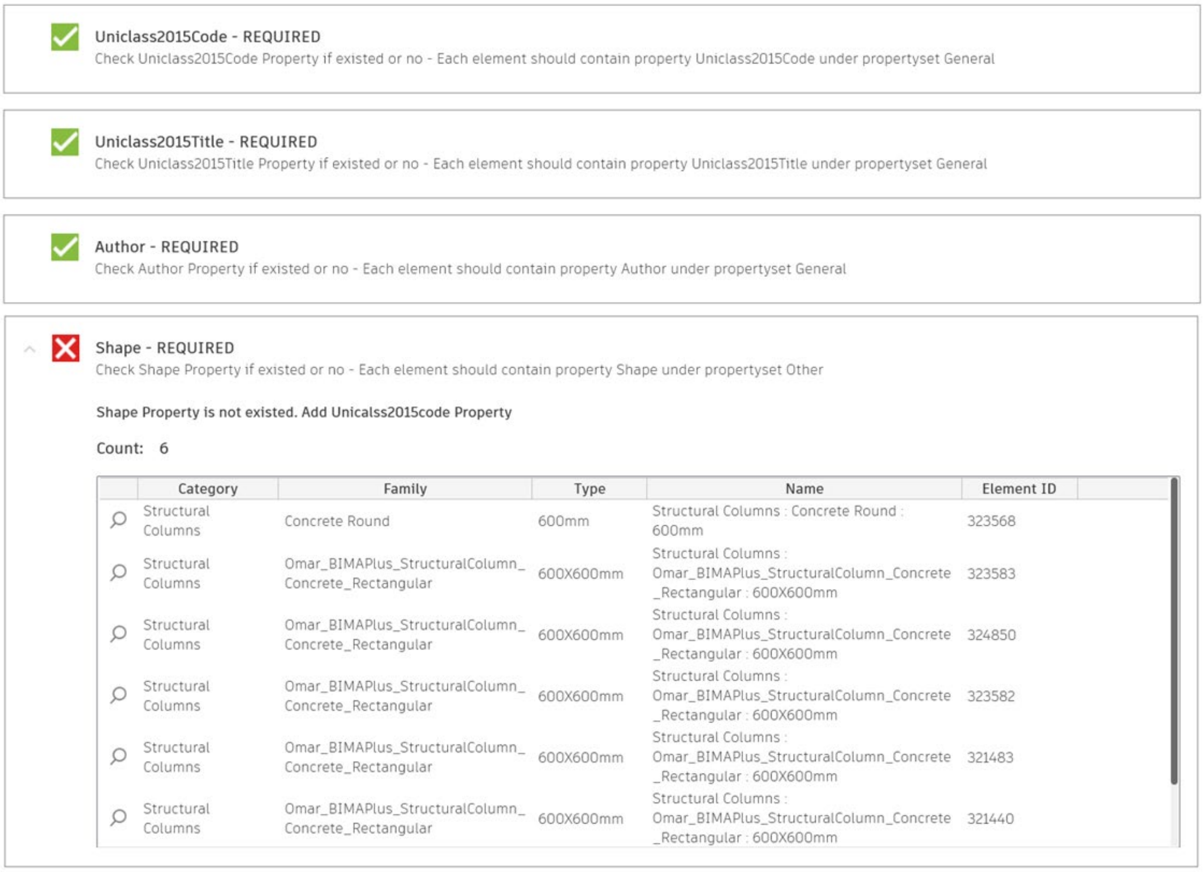


Figure 29: Autodesk Interoperability Tools - Check results

There is the possibility to select the element directly inside the repository environment and solve the issue. Also, this Report can be exported to HTML or Excel format, which can be used in the automation process of solving the issues. Other Check examples that can be done inside the Autodesk Model checker configurator are checking the property to have value/ no value and checking the property value using regular expression, as shown in Figure 30 and Figure 31.

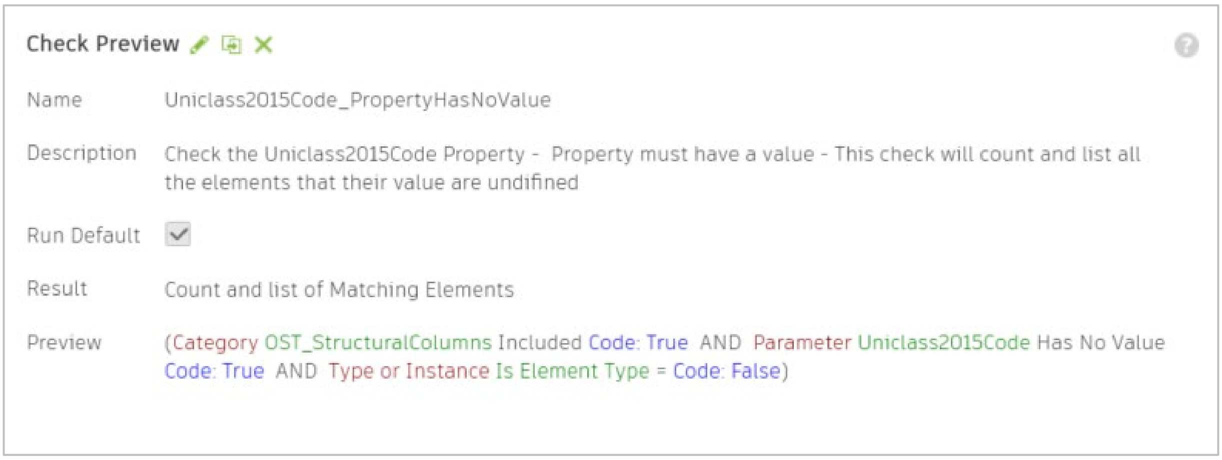


Figure 30: Autodesk Interoperability Tools - Check if the property has value/ no value

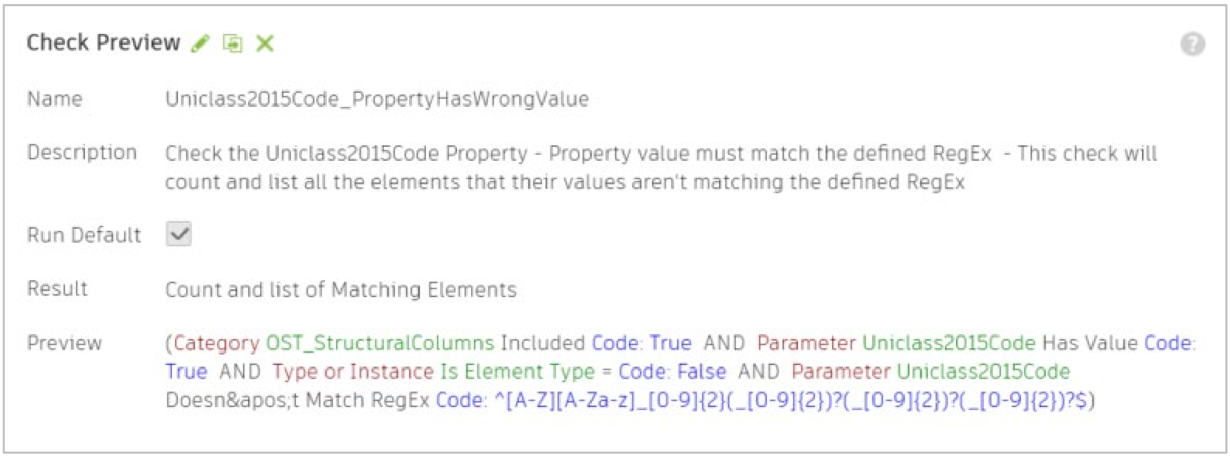


Figure 31: Autodesk Interoperability Tools - Check if the Property has a wrong value

4.4.2 Scenario 1 Application – Workflow B

To implement the model preparation stage, it is necessary to export the model to IFC format. In this case, the model is exported to IFC 4 format with a pre-defined User-Defined Property Set to map the needed properties correctly in line with NBS, as shown in Figure 32.

In this Workflow, SMC can be an option to implement the rule interpretation and rule execution stages. SMC Rules Libraries are used to translate the rules into a computable language. Each rule has a Name and SMC support tag, which can be used as a reference for classifying rules. In this scenario, the Rule of Required Property Sets with Tag SOL/203/2.5 and Property Values Must Be from Agreed List with Tag SOL/9/3.1.

As Shown in Figure 33, the checked components are selected, and the property set with the properties are defined. It is feasible to define the data type and add other conditions. As a result of the Check, A set of issues were classified and assigned with Element ID. It is possible to check whether the Property does not exist or is an issue in the value within the same rule as Shown in Figure 34. The final stage is to group the problems, create issues and exchange them to the main repository, using BCFs as shown in Figure 35.

```
PropertySet:    COBie T    IfcRoot
        AssetType    Text
        Category     Text
        Color Text
        DurationUnit        Text
        ExpectedLife        Integer
        Finish       Text
        InstallationDate    Text
        Material     Text
        Name  Text
        NominalHeight       Text
        NominalLength       Text
        NominalWidth        Text
        ReplacementCost     Text
        Shape Text
        ModelNumber Text    Model
        ModelReference      Text
        Manufacturer        Text  Manufacturer

PropertySet:    BOS_General T    IfcRoot
        Author       Text  Author
        BIMObjectName       Text
        ManufacturerName    Text
        ProductInformation       Text
        Revision     Text
        Uniclass2015Code    Text
        Uniclass2015Title   Text
        Uniclass2015Version      Integer
        Version      Text
        ManufacturerURL     Text  URL
```

Figure 32: IFC Export setting - Pre-defined User Defined

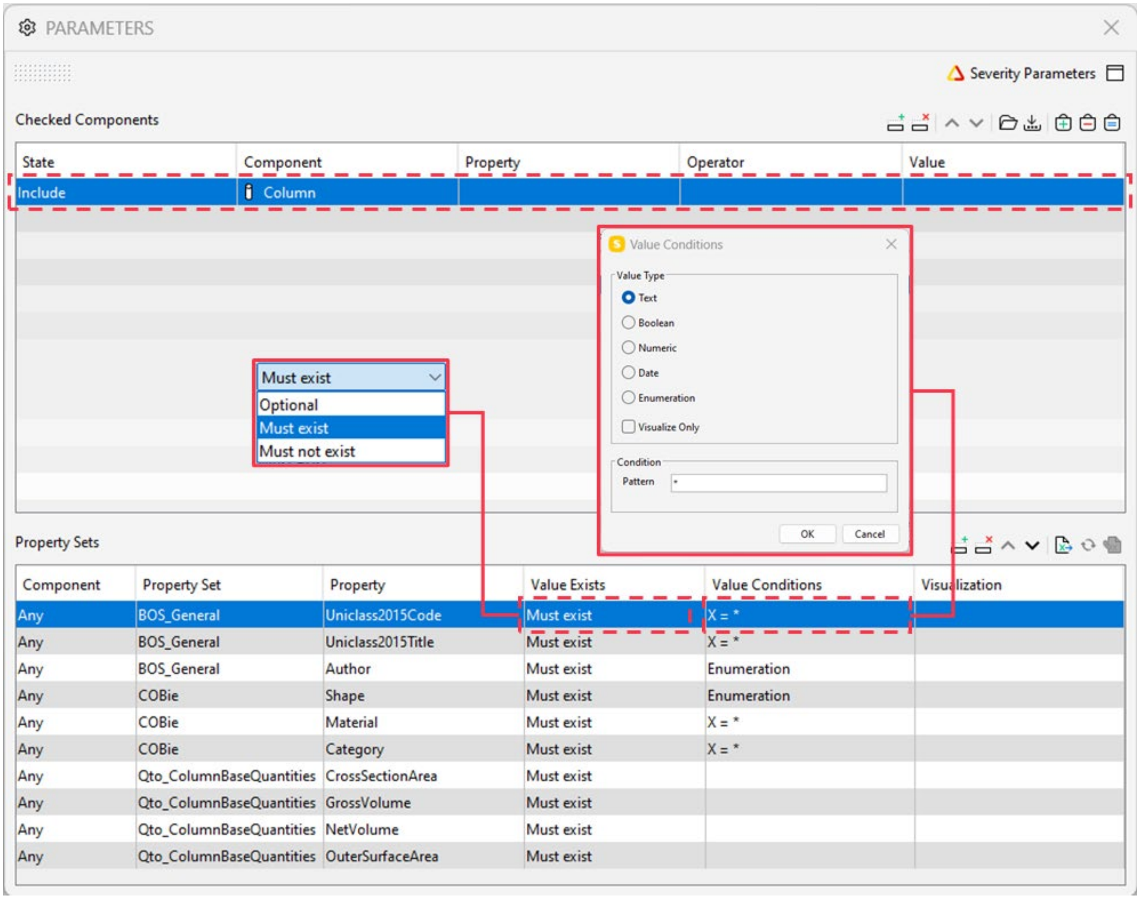


Figure 33: SMC – Rules execution

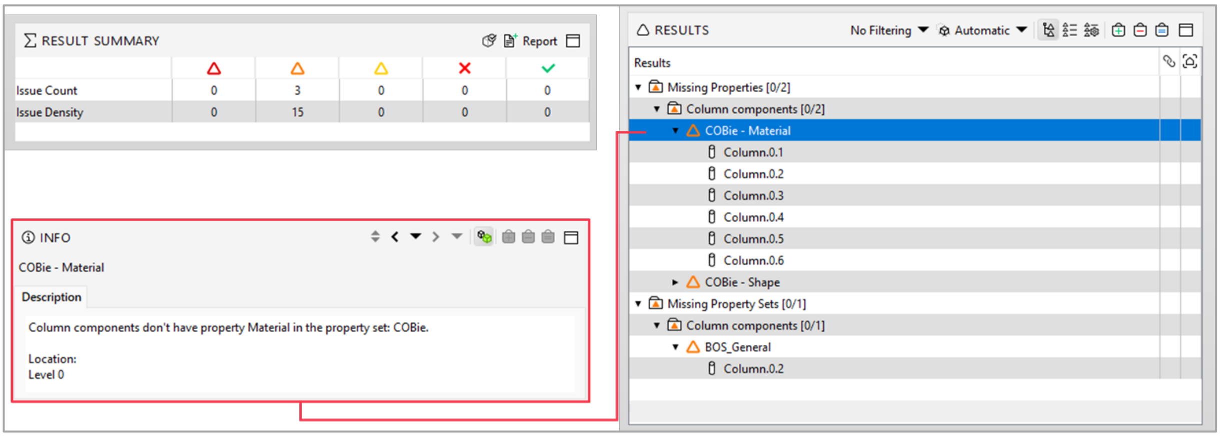


Figure 34: SMC - Check results

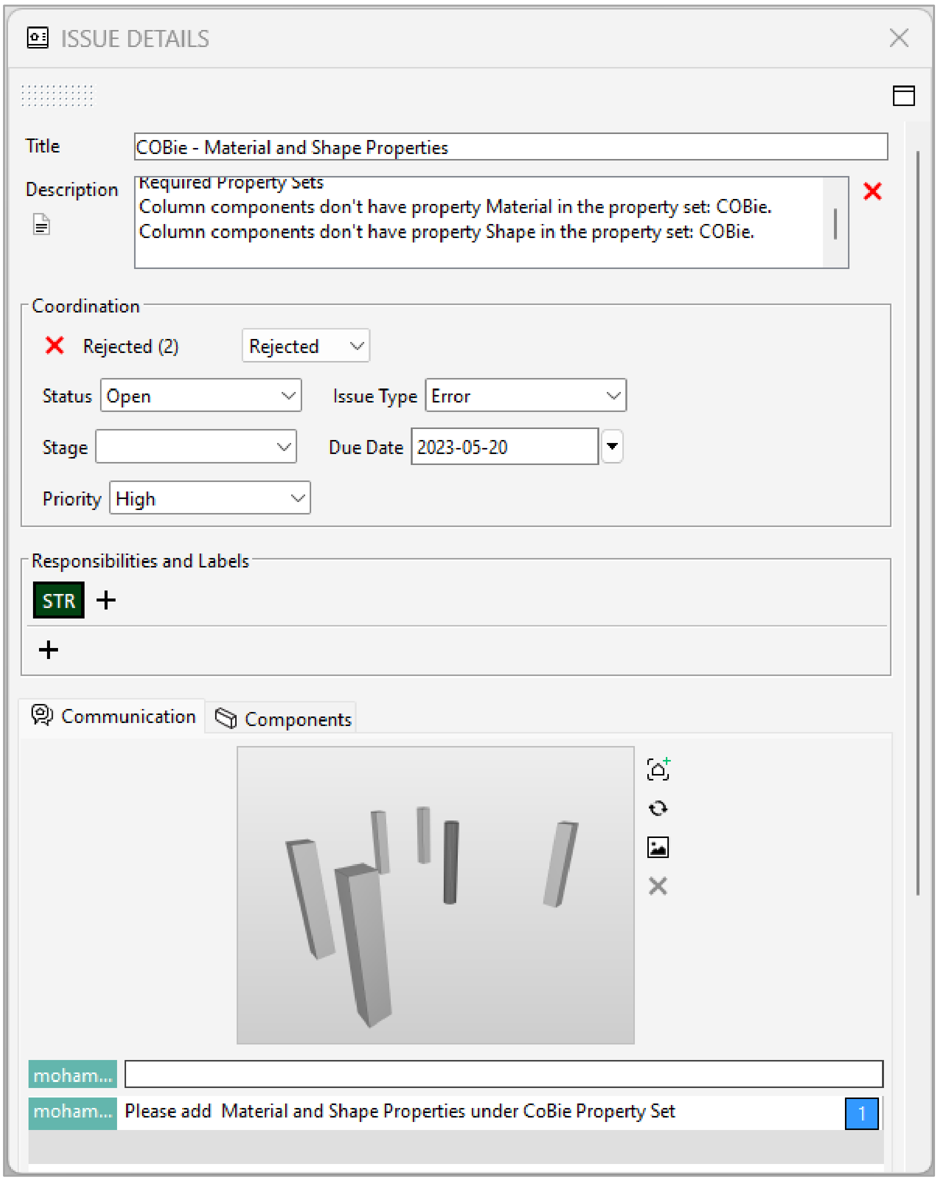


Figure 35: SMC - Issue creation using BCFs

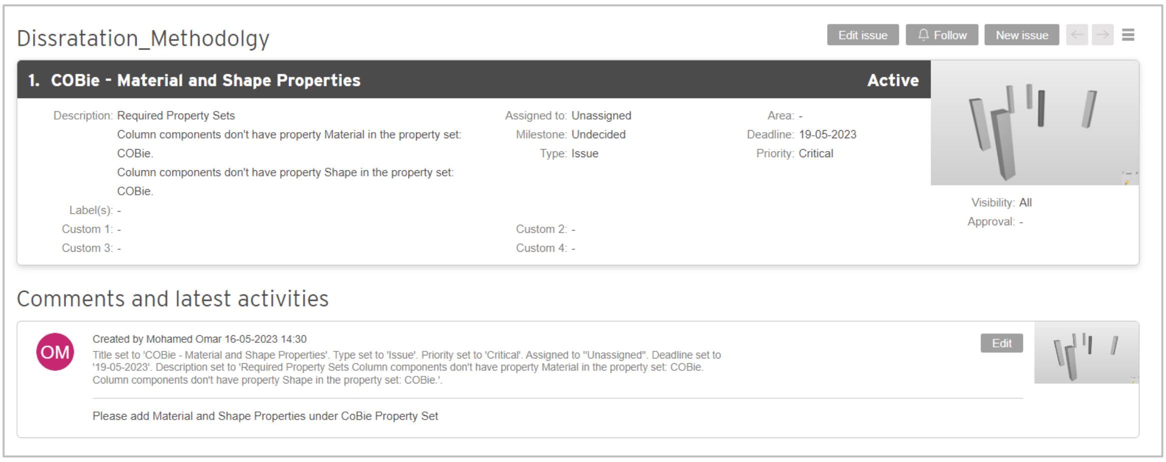


Figure 36: SMC - Collaboration using BCFs

Another Possible Software that can be used is Bexel Manager. Bexel Manager is a BIM-based software mainly used in the construction phase with powerful 4D and 5D simulation features. In this case, some checks can be implemented inside Bexel Manager using Property Checker Add-In. Bexel Manager has Excel Templates. Then, the following step is importing the Excel file, defining the current phase, and evaluating, as shown in Figure 37. It is possible to create an automatic selection set for each group of checks, as shown in Figure 38.

Parameter Name	Pset	Value Type	Phase	Condition	Key
Uniclass2015Code	BOS_General	Text	Design Development	match(['$'],'[A-Z][A-Za-z]_[0-9]{2}(_[0-9]{2})?(_[0-9]{2})?(_[0-9]{2})?')	
Uniclass2015Title	BOS_General	Text	Design Development		
Author	BOS_General	Text	Design Development	in(['$'],'Mohamed Omar', 'Structural Engineer')	
Shape	COBie	Text	Design Development		
Material	COBie	Text	Design Development		
Category	COBie	Text	Design Development		
CrossSectionArea	Qto_ColumnBaseQuantities	Area	Design Development		
GrossVolume	Qto_ColumnBaseQuantities	Volume	Design Development		
NetVolume	Qto_ColumnBaseQuantities	Volume	Design Development		
OuterSurfaceArea	Qto_ColumnBaseQuantities	Area	Design Development		

Figure 37: Bexel Manager – Property Checker Excel template

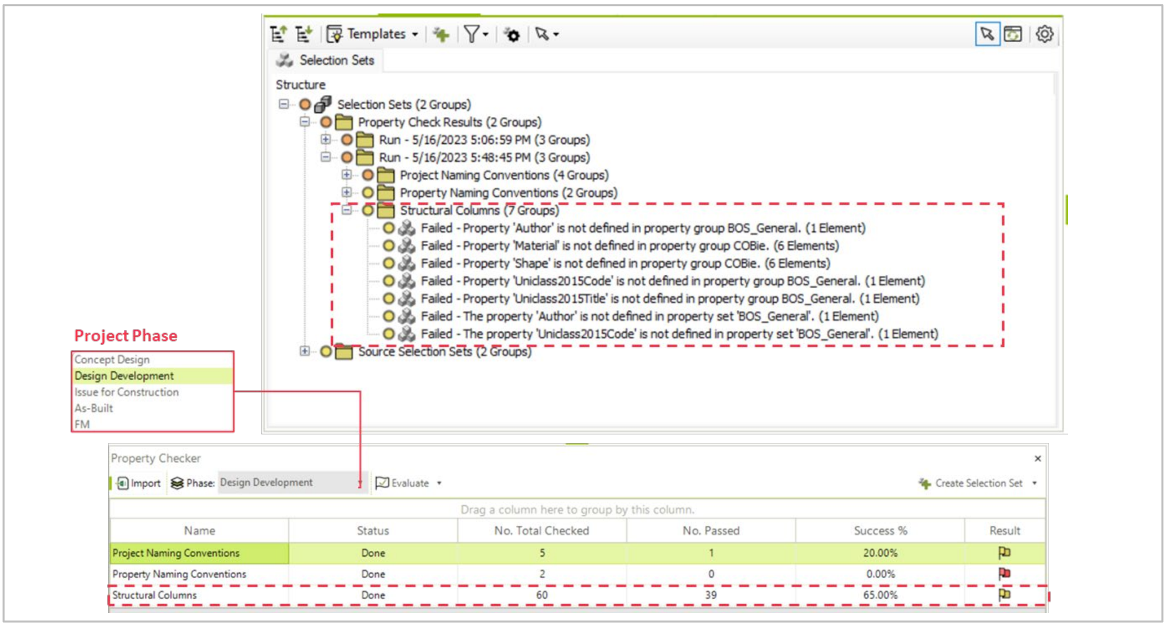


Figure 38: Bexel Manager – Property Checker and issues creation

4.4.3 Scenario 1 Application – Workflow C

In this part, the rule interpretation stage is conducted, translating rules into IDS format. An IDS sample of buildingSMART is used as a reference with IDS Documentation. First, Info was defined for the title, copyright, version, description, author, date, and Purpose, as shown in Figure 39. Consequently, the specification, applicability and requirements would be defined. In this scenario, the IFC version was IFC 4. In the applicability, IfcColumn was defined as the entity name. All rules can be interpreted using IDS Schema in the requirements, as shown in Figure 40. Also, regex can be implemented to check the convention naming.

IDS format can be complied with IFC files using many software such as BIMCollab and Blender. In this application, BIM Blender is used, and the result can be exported as an HTML, as shown in Figure 41.

```
F: > BIM A+ > BIM A+7 > 04 Small Case Study > IDS.ids
<?xml version='1.0' encoding='utf-8'?>
<ids xmlns="http://standards.buildingsmart.org/IDS"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://standards.buildingsmart.org/IDS/ids_05.xsd">
    <info>
        <title> MohamedOmar_BIMAPlus_IDS_Test</title>
        <copyright>Mohamed_Omar</copyright>
        <version>1.0.0</version>
        <description>
        This is an example of Some rules implemented on IDS Formart using BuildingSMART Samples as a reference
        </description>
        <author>mohamedahmedomar807@gmail.com</author>
        <date>2023-05-04</date>
        <purpose>Mohamed Omar Dissertation Checking Rules</purpose>
    </info>
```

Figure 39: IDS - Info

```
        </applicability>
        <requirements>
            <property measure="IfcText" instructions="Each Element should contain a classification property Uniclass2015Code "
            minOccurs="1" maxOccurs="1">
                <propertySet>
                    <simpleValue>BOS_General</simpleValue>
                </propertySet>
                <name>
                    <simpleValue>Uniclass2015Code</simpleValue>
                </name>
                <value>
                    <xs:restriction base= "xs:string" >
                        <xs:pattern value="^[A-Z][A-Za-z]_[0-9]{2}(_[0-9]{2})?(_[0-9]{2})?(_[0-9]{2})?$"  />
                    </xs:restriction>
                </value>
            </property>

            <property measure="IfcVolumeMeasure" instructions="Each Element should contain a  property GrossVolume "
            minOccurs="1" maxOccurs="1">
                <propertySet>
                    <simpleValue>Qto_ColumnBaseQuantities</simpleValue>
                </propertySet>
                <name>
                    <simpleValue>GrossVolume</simpleValue>
                </name>
                <value>
                    <xs:restriction base= "xs:numFacet" >
                    </xs:restriction>
                </value>
            </property>
```

Figure 40: IDS – Applicability

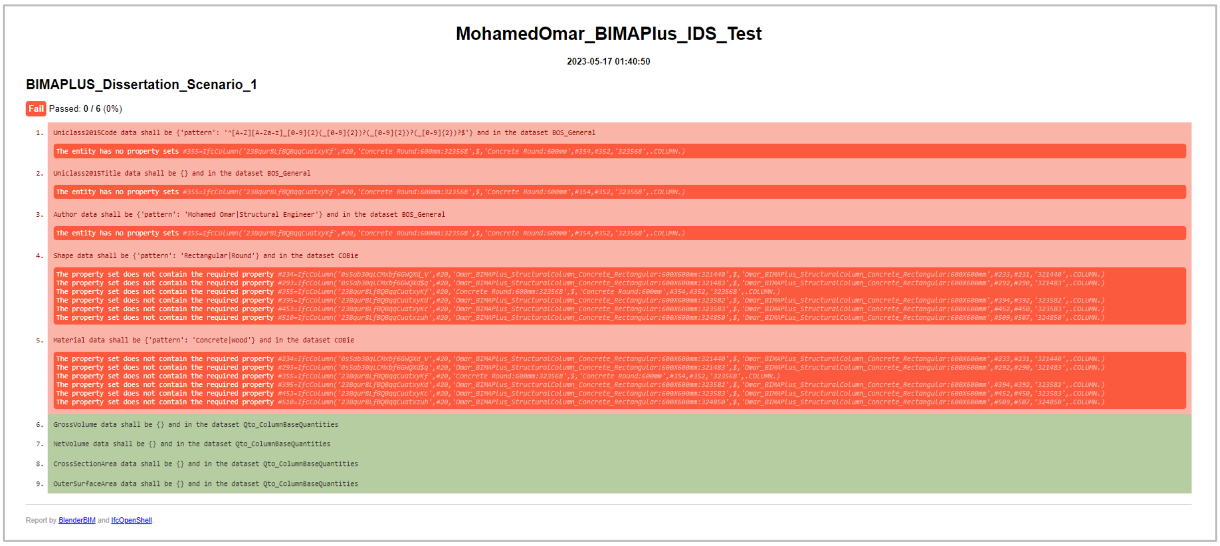


Figure 41: IDS – HTML results

5 CASE STUDY

This chapter undertakes the practical implementation of BMC workflows and scenarios to gain substantial insights into the challenges and solutions that arise during the BMC process. The predefined workflows are executed on a project undertaken by Protim Ržišnik Perc company (Thesis Partner). This process also involves defining information requirements tailored for the particular BIM use of QTO and cost estimation as input for the BMC process.

5.1 Project Description

The client intends to construct an administration building for its own needs on the parcel of land near the current production facility. The final solution must ensure functionality and represent the overall architectural image of the company. It must include internal and external landscaping, all in keeping with the existing layout of the area. It will be predominantly office space with an accompanying Programme of business activities. The client wants to manage the facility using a BIM model. As follows the indicative areas of the building:

- Total area of the building plot: 5,220 m2

- Gross floor area: 4,672 m2

- Elevation: assumed 1+2

- Employees: approx. 105 seats

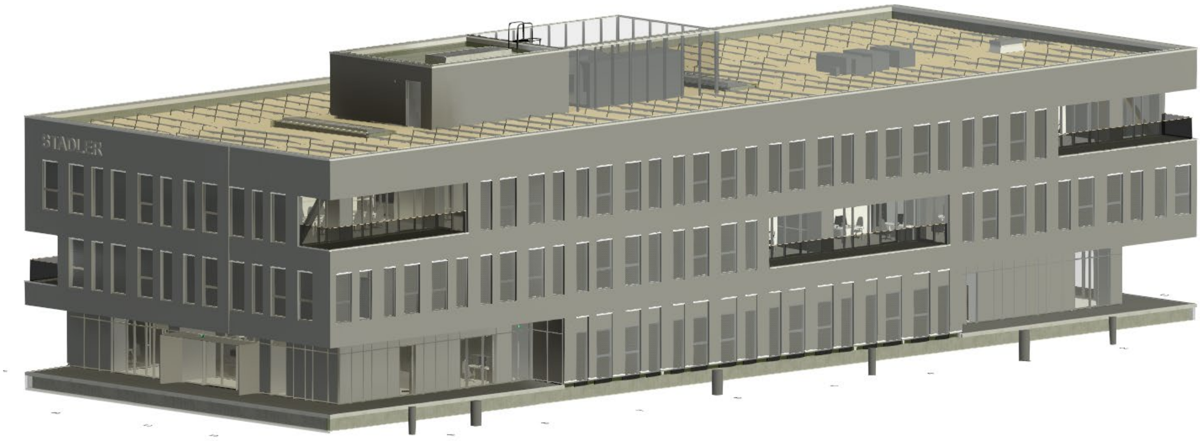


Figure 42: Case study project BIM model – 3D View

5.2 Project Phases

The project has five main phases, as explained in Table 9. The checking process occurs in the detailed design phase (PZI phase). It is possible to use the same process for all phases. Nevertheless, each phase will have a customised Level of Information Need.

Table 9: Project phases

Phase (In Slovenian Language)	Description
Idejna Zasnova (IDZ)	Conceptual Design
Idejni Projekt (IDP)	Preliminary Design
Projekt Za Pridobitev Gradbenega Dovoljenja (PGD)	Design For Construction Permit
Projekt Za Izvedbo (PZI)	Detailed Design
Projekt Izvedenih Del (PID)	As-Build Design

5.3 Model Composition

In architectural and structural design processes, it is customary to manage architecture and structural models separately, usually in distinct files. However, there might be flexibility in combining both aspects into a single file, depending on some specific project requirements and the modelling standards of the company. In the case study, the architectural and structural models were modelled into a unified file. This approach allows for enhanced collaboration and streamlines project management during the design phase.

Revit Worksets were utilised to ensure efficient organisation and control over the project. Worksets divide the project into logical sections, enabling multiple team members to work simultaneously on different components. Access to specific sections of Structural, Architectural, and plumbing can be controlled, as shown in Figure 43.

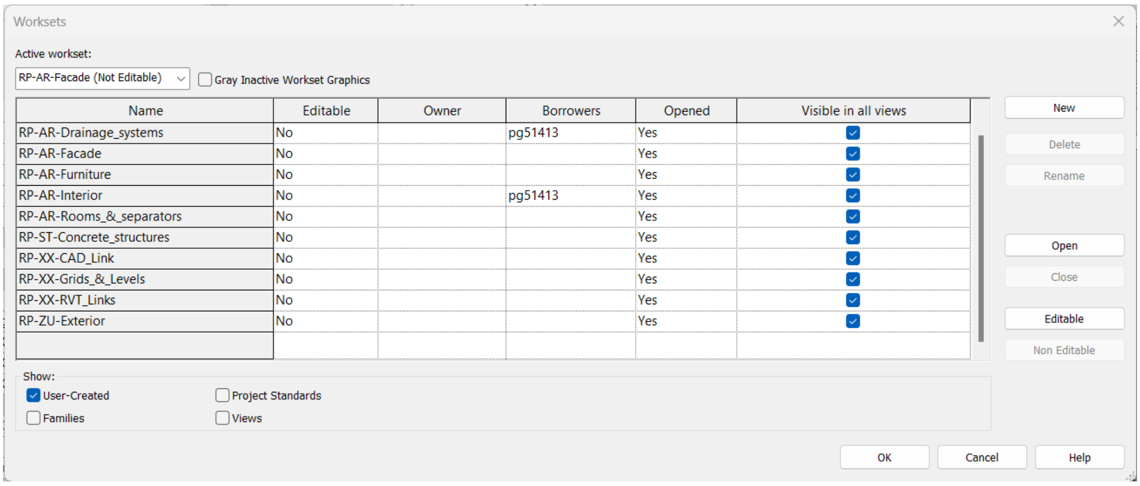


Figure 43: Case Study current worksets

5.4 Level of Information Need for QTO and Cost Estimation Purpose

A project-specific approach is essential in the context of QTO and Cost estimation. This necessity arises due to the variability in project types, where some are lump-sum contracts, and others are measurable contracts. Additionally, the information requirements for each element may differ based on the unit of measurement used in the project.

As a result, developing a customised Level of Information Need is crucial to establish the specific information requirements for each unique project, considering the sub-divisions mentioned earlier. In this part, a comprehensive and detailed Level of Information Need is defined and tailored to meet the particular demands of the project at hand. These information requirements form the rules' interpretation process inputs, serving as the initial step to start the BMC process.

Based on the data presented in Table 3, Pg.15, a structured template has been devised and is illustrated in Table 10. This template aims to facilitate a systematic and uniform method for gathering the essential information related to the elements within the case study, ensuring consistency throughout the project. The focus is the information related to BIM uses of QTO and Cost estimation. The practical implementation of the Level of Information Need framework on the case Study Elements is provided in Appendix A, Pg.93, for reference.

Table 10: Project Level of Information Need framework template

	Project: Project Number (V158770)			
	Purpose: QTO & Cost Estimation			
	Actor: Lead Appointed Party- Design Consultant			
No.	Element Name (Header)			
General Information	**Model Element**	Element Name		
	Category	Revit Category (Walls, Doors, Windows, …)		
	Information delivery milestone	Project Phase: (IDZ) Conceptual Design, (IDP) Preliminary Design, (PGD) Design For Construction Permit, (PZI) Detailed Design, (PID) As-Build Design		
	Family Type	System Family, Loadable Family, In-Place Family		
Geometrical Representation	**Detail (Using LOD)**	N/A (if needed: 100, 200,300, 350, 400)		
	Dimensionality	0D,1D,2D,3D		
	Location	Absolute, Relative		
	Appearance	No colour, Single colour, Material colour, Material colour and textures		
	Parametric behaviour	N/A (if needed: Explicit geometry, Constructive geometry, Parametric geometry)		
Alphanumerical Information	**Identification**	Family Name and/ Or Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Structural	Structural	Boolean (Yes/no)
		Structural	Structural Usage	(Bearing, Shear, NoBearing)
		Identity Data	Type Mark	Defined as Project Need
		Identity Data	Mark	Defined as Project Need
		Identity Data	Workset	Refer to the company worksets manual

		Dimension	Material	Defined/ Naming Convention (one of Company material list
		Dimension	Volume	Defined/ Unit
		Dimension	Area	Defined/ Unit
		Dimension	Length	Defined/ Unit
		WBS	WBSCode	
Documentation	**Set of documents**	N/A		

5.4.1 Elements General Information

Within this domain, each Element is delineated by fundamental details, including its associated Revit category employed for element modelling the family type to which it belongs, whether System family, Loadable family, or In-place model. Generally, In-place model families are not the preferred option in most cases. Additionally, the information delivery milestone, denoting the project phase, is established in alignment with the company's designated project stages.

5.4.2 Elements Geometrical Representation

In the context of QTO and Cost estimation, the emphasis on Geometrical representation lies in the attributes of Dimensionality and location. To illustrate, the Dimensionality requirement for a railing can be either 1D, 2D, or 3D contingent upon the mode of measurement employed - whether it pertains to the entire length of the railing or entails computations of sub-components such as Area of Glass and Length of Aluminium Profiles and their thicknesses. In the context of the case study, the element is measured solely by its overall length, resulting in a dimensionality of 1D for this application. Consequently, it is essential to acknowledge that in different projects, the dimensionality requirement may vary and can be subject to change based on specific measurement methods and project requirements.

Moreover, the aspect of Location assumes paramount significance, as it facilitates a profound understanding of whether the element in question is hosted or part of other elements, thereby establishing a clear nexus between the element and its environment. While various other geometrical attributes may exist, their relevance to these uses may not be needed.

5.4.3 Elements Alphanumerical Information

Family name and/or Type name are employed as identifiers for Elements. In addition, a set of properties must be defined and have correct values. Each element shall have essential information concerning its Type, Type Mark, Workset, Material, and calculated quantity (Count, Volume, Area, etc.). Furthermore, their structural usage shall be clearly defined for certain elements such as walls, columns, floors, and stairs. A unique Mark for each instance may also be required for elements like doors and windows. The precise information on each component can be found in Appendix B on Page 105.

5.4.4 Documentations

Previous examples of other projects' Bills of Quantities and Materials could be used as a reference. Nevertheless, there is no compulsory need for specific documents for these uses in this project phase.

5.4.5 Regular Expression (regex) Definition

Regular Expression (regex) is a pattern that the regular expression engine attempts to match in the input text. It allows text identification using a pattern instead of an exact string. The pattern consists of one or more-character literals, operators, or constructs that are commonly used for parsing text with general-purpose languages, validating content in web forms using Javascript, and searching text files for specific patterns (Chapman & Stolee, 2016; *Regular Expression Language - Quick Reference*, n.d.).

In the domain of BIM, regular expressions are often employed to enforce naming conventions for BIM models. For example, a BIM project might require all rooms to be named using a specific format, such as "Floor-RoomNumber," where "Floor" is a numerical value and "RoomNumber" is an alphanumeric identifier. A regular expression can be designed to check each room's name against this pattern, ensuring compliance with the required naming convention. Any deviation from the standard can trigger an alert or error message, guiding users to correct the naming and maintain uniformity throughout the project.

In the case study project, a set of regexes was developed to implement the company naming convention criteria for Model Elements, Materials and other related properties needed. The whole regex translation can be found in Appendix C, Pg.107. Below are some examples of regex implementation to check the naming conventions of BIM elements Identifiers (Family Name and/or type Name) that will be used later to check processes of identifiers names,

regex Implementation- Example 1:

The ceiling Element is a system family. Referring to the Level of Information Need framework (Appendix A, Pg. 93) of the ceiling, the identifier used is the Type Name. As Shown in Figure 44, the Type name consists of 5 main parts, which can regex translated as follows:

1- Origin-Discipline: String prefix (RP-AR)
2- Elements Origin: list of sub-types using ( ) as a boundary for a list, | as a separator.
3- Material: list of materials.
4- Differentiator (optional): While it is the same concept of creating a list, a quantifier was used to make it implemented as optional (??: known as "Lazy quantifier" in the regular expression, and it is used to repeat one of the list zero or one time)
5- Thickness: Range of number from 0 – 9 using star sign *, which repeats the number unlimited times. A suffix of mm was added to make sure the unit is correct.

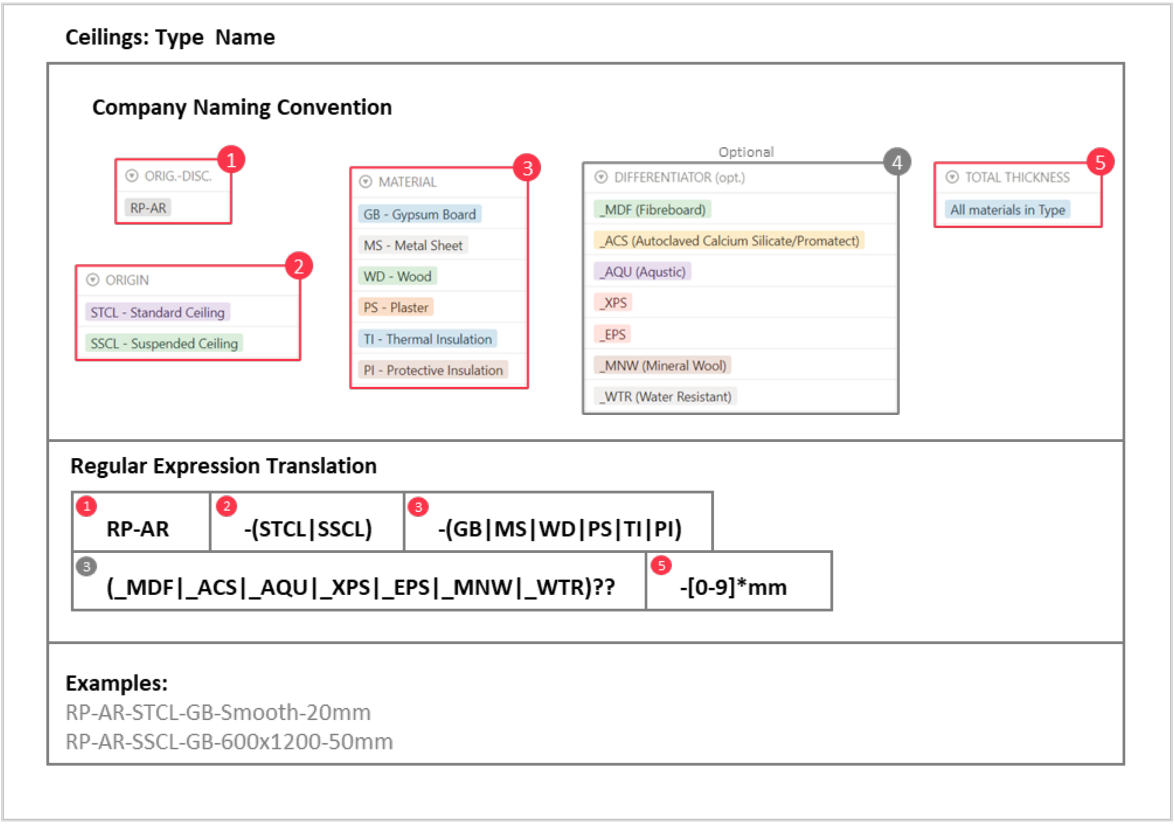


Figure 44: regex implementation/Ceiling identifier

regex Implementation- Example 2:

Doors are loadable families with two identifiers: Family Name and type Name. For Family Name, the regex implementation, as shown in Figure 45, can be explained as follows:

1- Origin-Discipline: Prefix String (RP-AR) represents the company name and architecture discipline.
2- Elements Origin: list of sub-types using abbreviations in the naming convention. For instance, SGL stands for Single door.
3- Differentiator(optional): an optional list using the lazy quantifier ??
4- Frame Type(optional): an optional list using the lazy quantifier ??
5- Leaf Type(optional): an optional list using the lazy quantifier ??
6- Addition description(optional): an optional list using the lazy quantifier ??
7- File Type / Revit Version (optional): It consists of two parts; String (M3-) represents the model file type, which is 3D. The second part describes the Revit version, which can be Revit 20- 25. This can be translated by putting the prefix of R2, and this integer ranges from 0-5.

The Type Name identifier uses the same concept except for regex part 3 of the Type name. A range of Characters and numbers were added to give the freedom to put a unique identifier when needed.

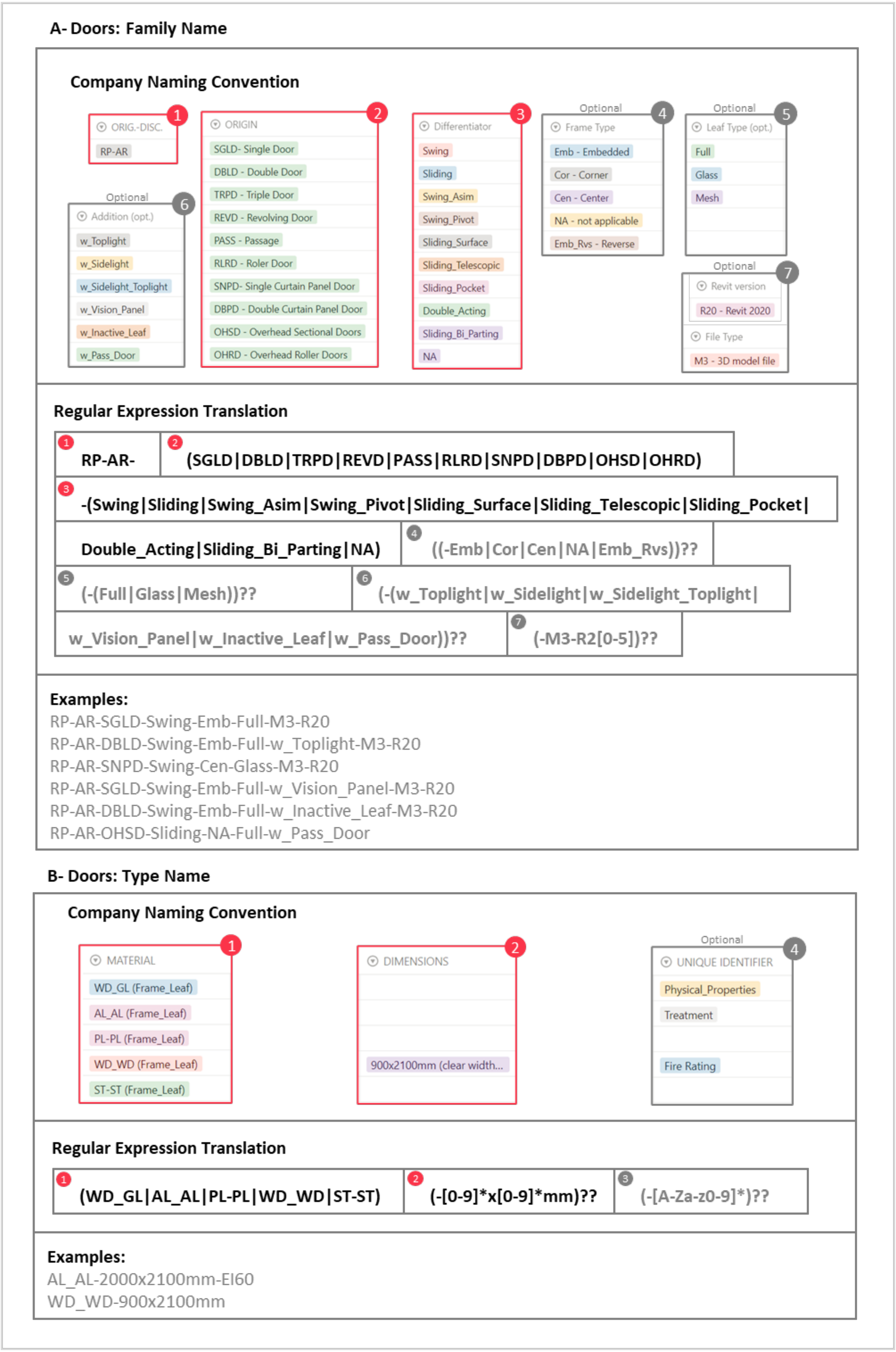


Figure 45: regex implementation/ Doors identifiers

regex Implementation- Example 3:

The structural foundation category encompasses both system family and loadable family elements. Three identifiers were devised to effectively manage both types of families: one for the family name of the loadable Family category and two identifiers for both loadable and system families, as illustrated in Figure 39. The approach follows principles similar to the previous examples, albeit with additional constraints. Notably, the material options were restricted to Reinforced concrete, which rendered the need for a list of materials unnecessary. Instead, an "RC-" string prefix was introduced to indicate the material used for the foundation elements. Additionally, the dimensions of the foundation could be either one thickness or represented in 3D dimensions. Therefore, a list of the two options was created to accommodate this variability.

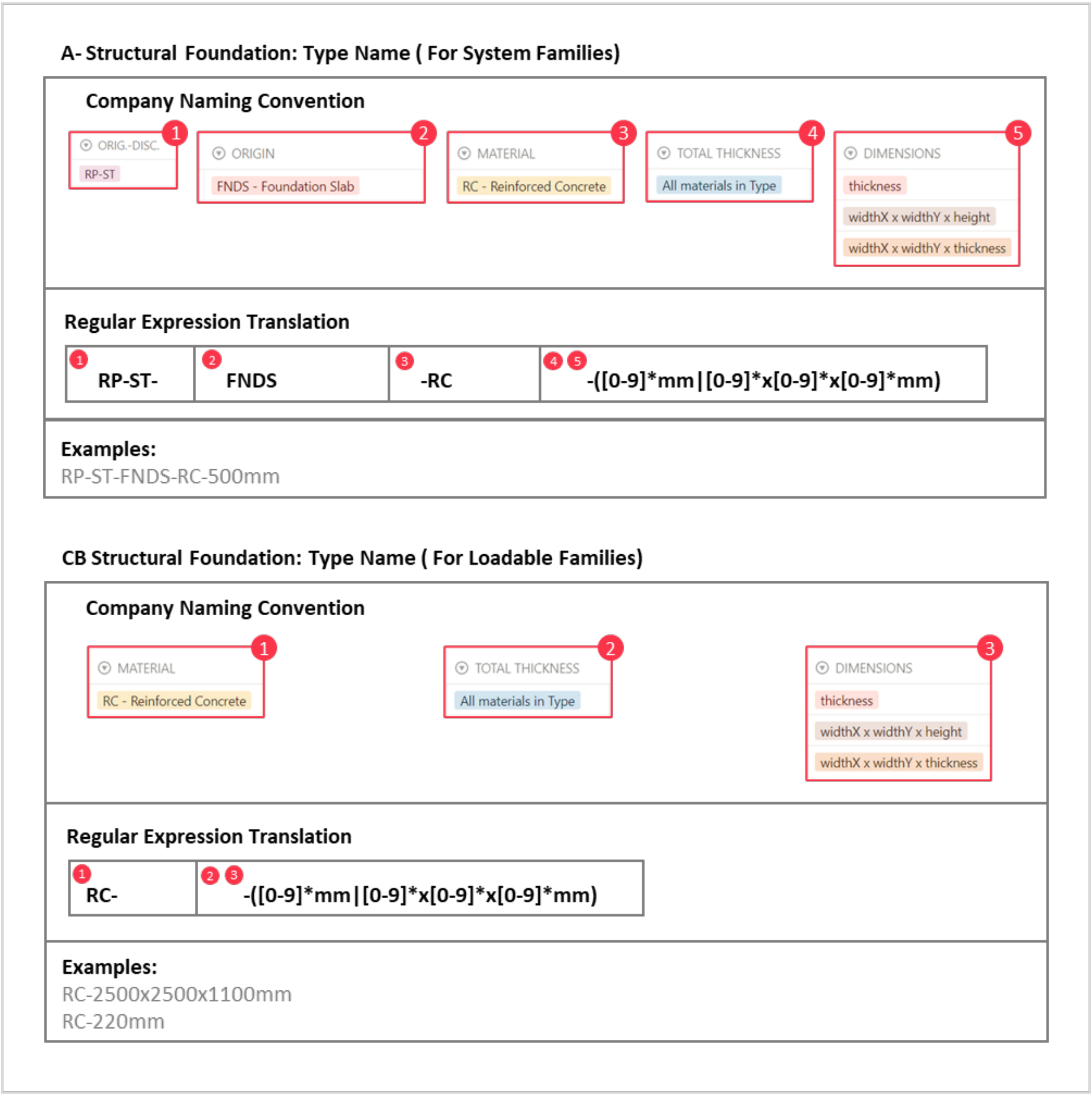


Figure 46: regex implementation/ Structural Foundation identifiers

5.5 BIM Model Rules and Metrics

BIM Model Rules is the initial step before the implementation of BMC processes. This involves meticulously examining a predefined set of quality rules, encompassing geometrical and Alphanumeric information requirements, to assess the BIM model's adherence to company standards. The aim is to ensure the model's accuracy and compliance with project-specific needs before its utilisation in various BIM uses such as Clash Detection, 2D Documentation, QTO, and Cost Estimation. As such, Rules are not limited to only the defined Level of Information Need, but also some Quality rules shall be included. It has to be defined to be computed later.

In defining the necessary rules, the process incorporates references from existing company BIM model rules and metrics. These rules form the foundation for rule interpretation, with modifications to enhance organisation and hierarchy. Also, the rule interpretation process considers established practices from Autodesk Revit 2024, 2023, Dutch Revit Standards, and Penne State BIM Standards.

Appendix B (Pg. 105) presents the defined set of rules, encompassing quality, geometrical, and information requirements. In this context, the previously established Level of Information Need and regular expressions (regexes) are incorporated as essential components of the rule interpretation process. By adhering to these defined rules, the BIM model's quality and accuracy can be effectively assured, facilitating the BMC Process throughout the project lifecycle.

5.6 BMC Implementation

This section aims to examine the impact of BMC on the Information quality of the BIM model. The analysis will encompass a series of automated quality, geometrical, and information checks to assess the effectiveness of BMC in improving the accuracy of QTO and Cost estimation processes. Additionally, the study focuses on evaluating the overall model quality and conducting geometrical and alphanumerical checks.

To begin, case study elements are selected, and a Basic Bill of Quantities (BOQ) and Bill of Materials (BOM) are obtained as part of the initial analysis. Subsequently, each workflow undergoes the prescribed processes, following the previous descriptive implementation stages. The iterative nature of this process facilitates the identification of potential adaptations and enhancements that BMC may bring to improve the quality and integrity of BIM models.

5.6.1 BMC Implementation of Workflow A

This workflow uses Autodesk Revit as the modelling environment with Autodesk Interoperability Tools for model checking. Instead, Dynamo codes are used for specific rules that cannot be implemented through Autodesk Interoperability Tools.

5.6.1.1 Rules Interpretation

As illustrated in Figure 47, the rules' check set structure is divided into three main headings: General Model quality checks, Geometrical checks, and Information Requirements checks. Rules are derived from the textural rules developed in Appendix B.

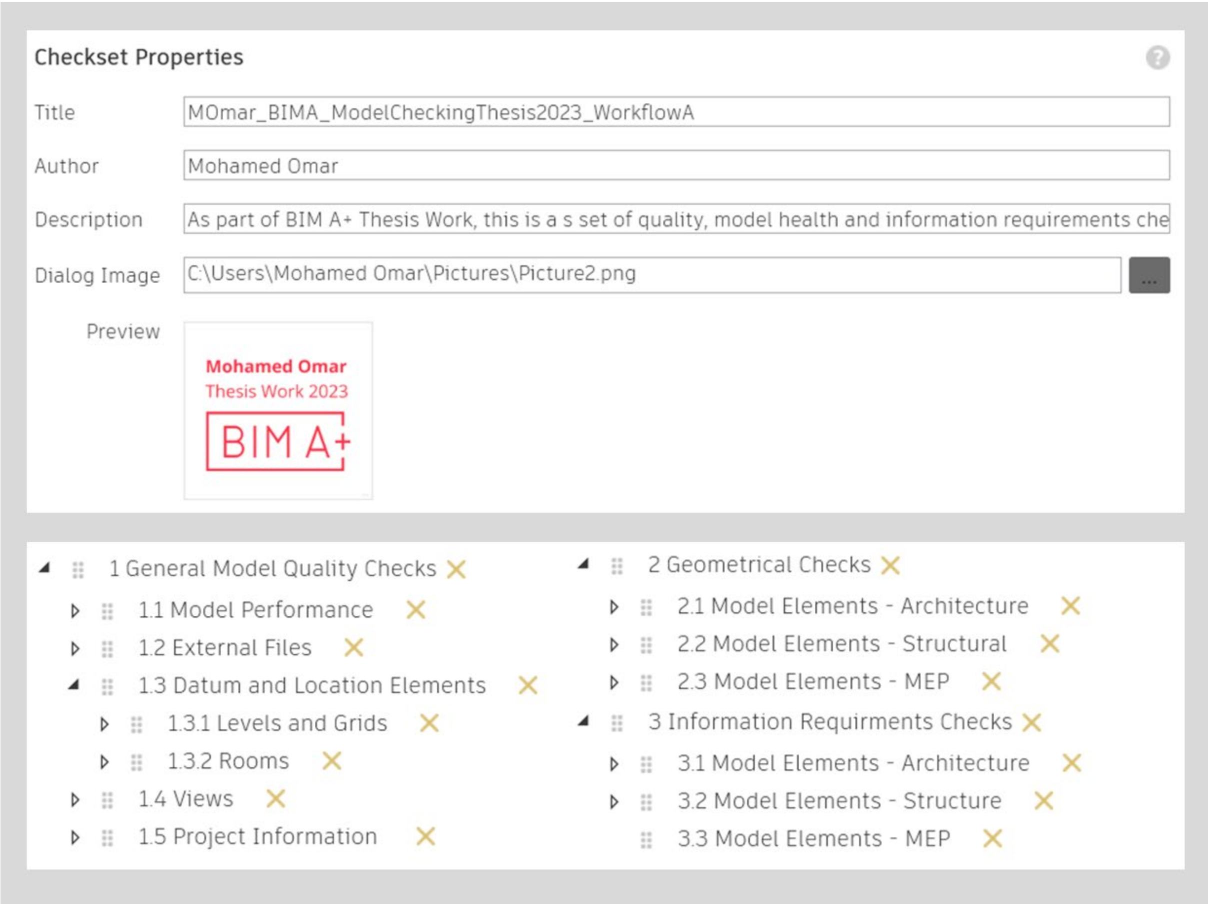


Figure 47: Workflow A - Checkset structure

Rule translation usually has two parts; the first part is the setting/category/ sub-category the rule is relevant to, and the second part is the characteristics, whether it is defined or not, has value or not. Using pre-built rules such as file size, warning Count, Pinned Links, etc., is possible. However, most checks should be customised to implement the company standards.

As illustrated in Figure 48, detailed checks were created. The full XML for the whole checking set with all sets can be found at: https://github.com/mohamedomar807/MohamedOmarBIMAPlus.

Figure 48: Case Study - Workflow A- Rules interpretation

To illustrate the Rules interpretation process, below are some examples of translating textural company rules, Defined Level of Information Need and regexes to create the checks.

Example 01: Checking Type Name (Level of Information Need Identifier) regex for Structural Foundation

The Structural Foundation Category can be a system family or a loadable family. As elucidated before in regexes of Structural Foundation Identifiers (see Figure 46, Pg. 52), there are two regexes for Type Name. Parameter- Host was used in filters to accommodate the rule for both family types. In the case of a loadable family, the Host Parameter has a value. In contrast, the Host parameter is not defined or has a value for system families, as elaborated in Figure 49.

Operator Or is used to define the group of filters and the regex rule is implemented for each family type. The check fails if any element does not match the regex.

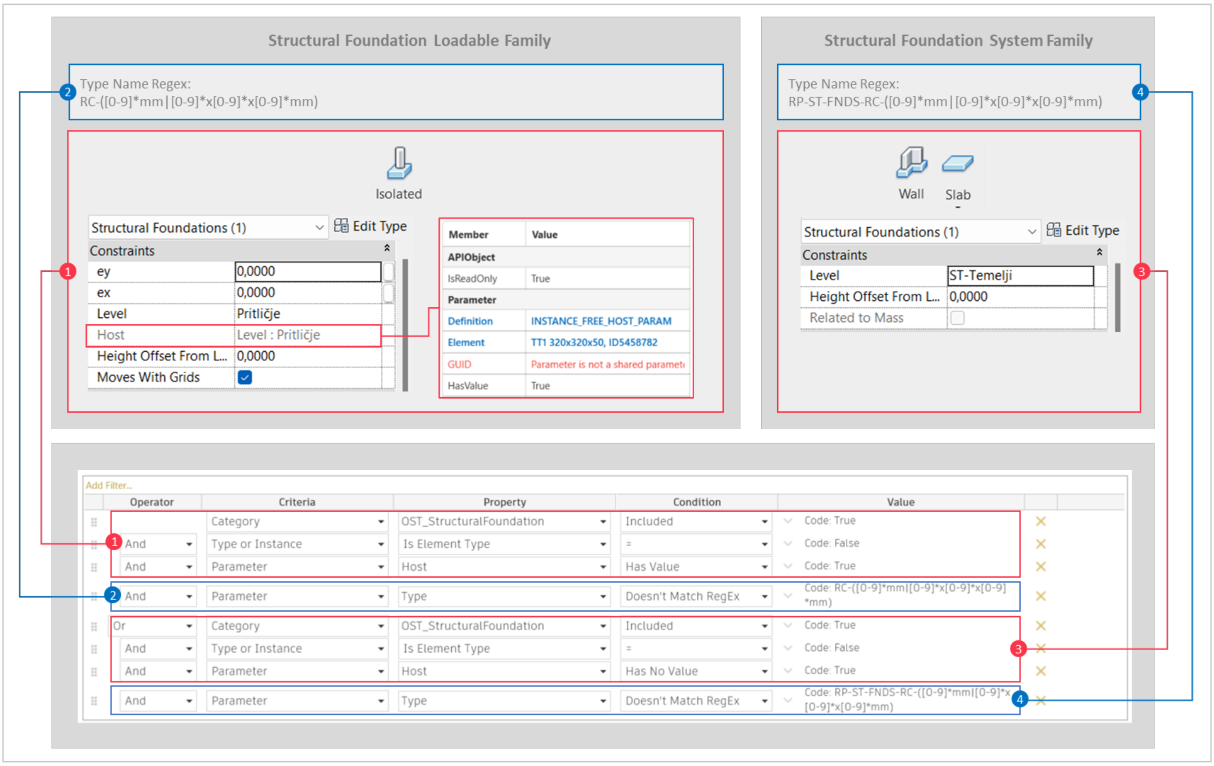


Figure 49: Checking structural foundation identifier - Type Name

Example 02: Checking Worksets Interior/ Exterior Walls and Interior/ Exterior Curtain Walls

Curtain Walls, Interior partitions, exterior Finishing walls, Shearing Walls and Bearing Walls belong to the walls category. That is why it is essential to check the worksets and ensure each element is modelled according to the worksets' breakdown structure. For this example, as elaborated in Figure 50, Parameter Angle is used to classify curtain walls and partitions. Angle parameters are only defined for curtain walls. Also, The Structural walls were excluded by using the "Does not contain" Condition. Other filters, such as structural Usage and structural (Boolean Parameter), were used in further checks to get the needed results.

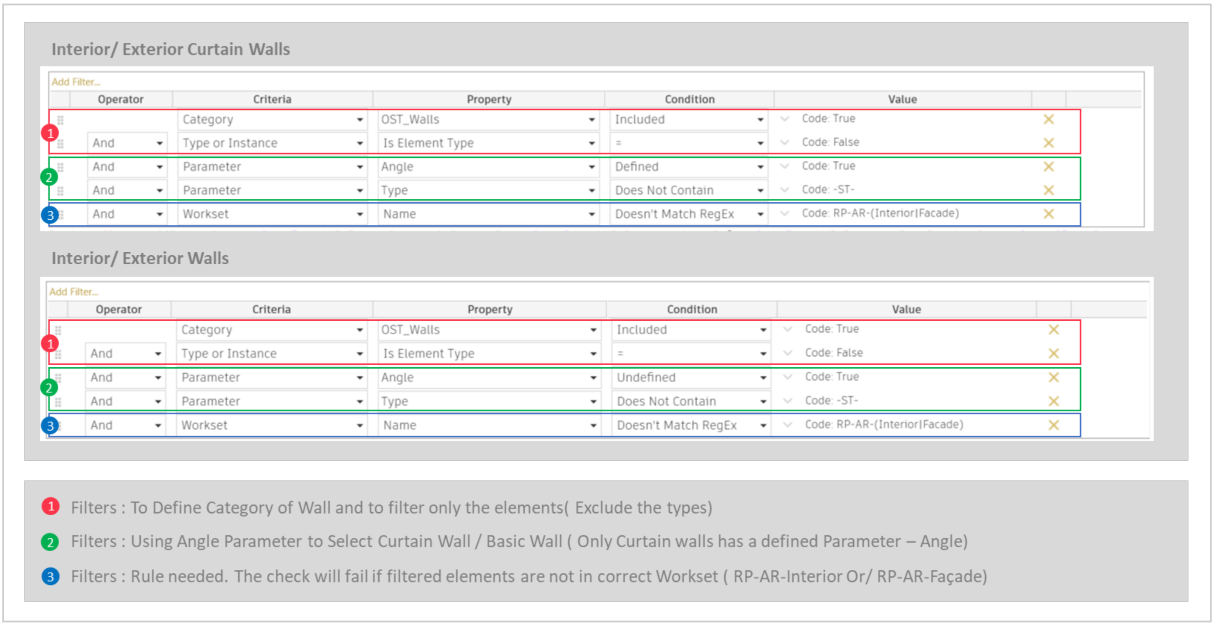


Figure 50: Checking interior/ exterior walls and interior/ exterior curtain walls - Workset

Example 03: Checking Quantity property(s) for Architectural (Finishing Floors)

As the Level of Information Need framework defines, the Finishing Floor shall have Area and Volume properties and value. As illustrated in Figure 51, each of the Area and Volume properties is checked twice to check if it is defined/ Undefined and then to check if the property has value/ no value. While it is required to check that Area and Volume Property API Parameter: IsReadOnly have a true value, it is impossible to check it using Model Checker Configurator.

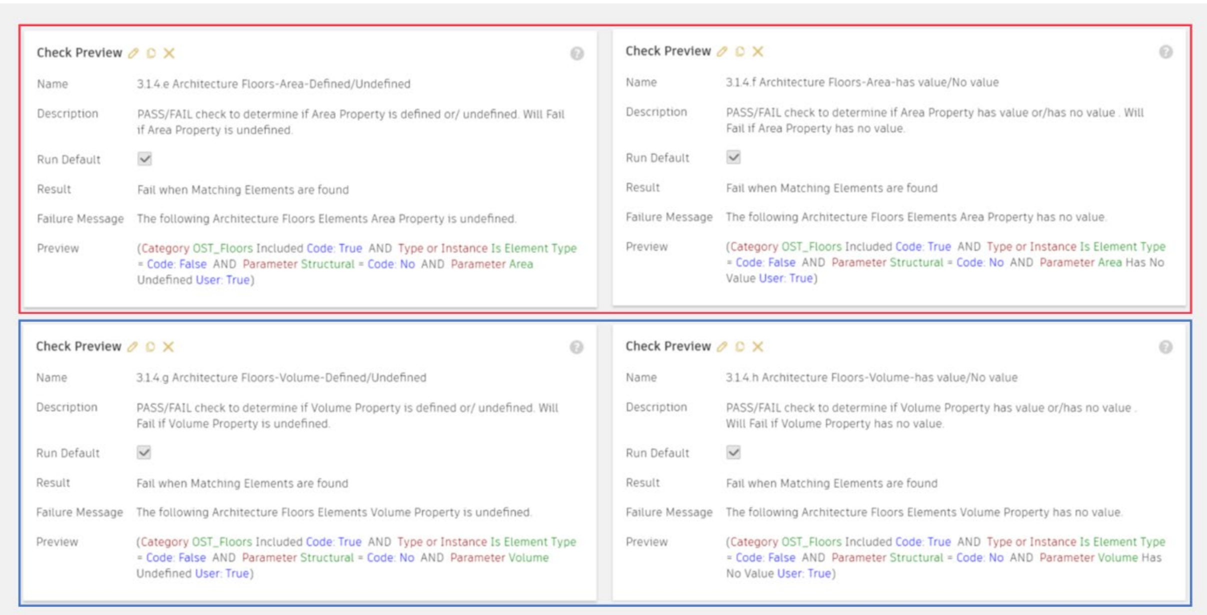


Figure 51: Checking finishing floors - Quantity properties of Area and Volume

Example 04: Duplication of architectural elements

While the Model checker configurator is not used mainly for clash detection, it can also be used for basic checks, such as duplication of elements, as shown in Figure 52.

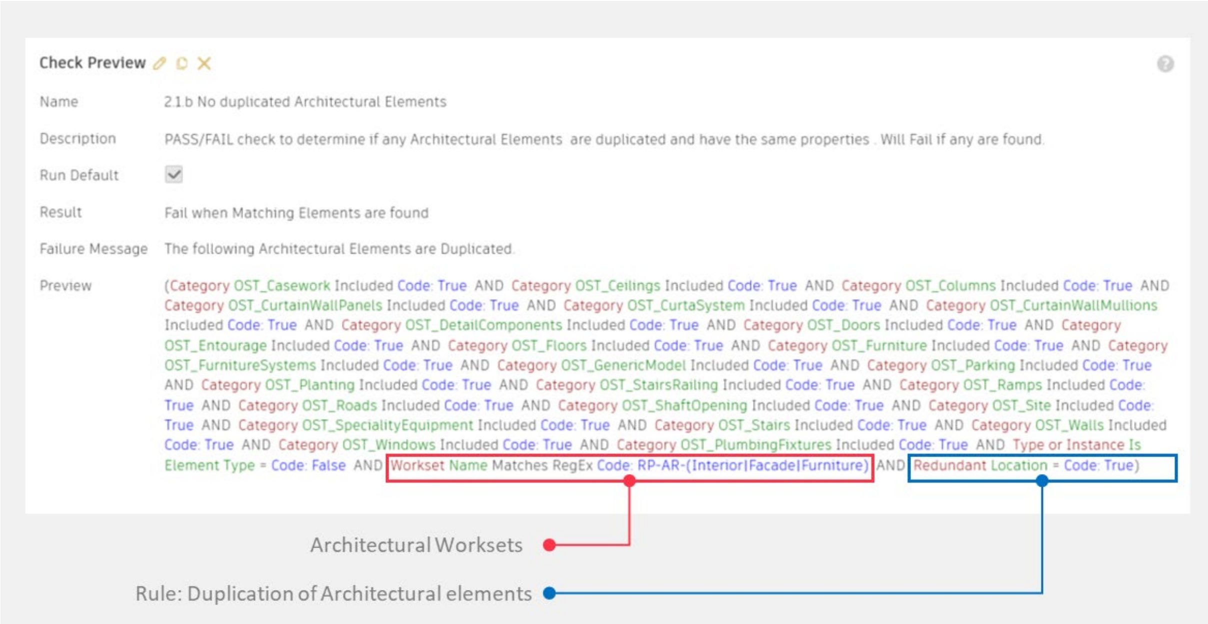


Figure 52: Checking duplication of architectural elements

5.6.1.2 Building BIM Model Preparation

The original RVT files are used in this workflow, and the modelling and information enrichment processes are implemented inside the repository. Exporting to other formats, such as IFC or NWC, is unnecessary. As described in the model composition, Architectural and structural elements are modelled in the same file. The Architectural/ Structural RVT file is used in this case study.

5.6.1.3 Rules Execution

Information requirements are focused more on Architectural/ Structural elements. Thus, The Checks are executed on the Architectural/ Structural file. It is possible to run the same checks on the MEP files simultaneously to showcase the possibility of running the checks on many files.

5.6.1.4 Reporting Checking Results

Excel and HTML reports are generated. Excel information can be used for the graphical representation of Results using Power PI or HTML for exchanging results. Both results have GUID for the failure element; the action needed is defined in the failure message.

Also, the Check summary, as shown in Figure 53, can be used as a general indicator of the quality of the model. However, each result should be checked to match all geometrical and information requirements needed. Below are some of the examples of the results.

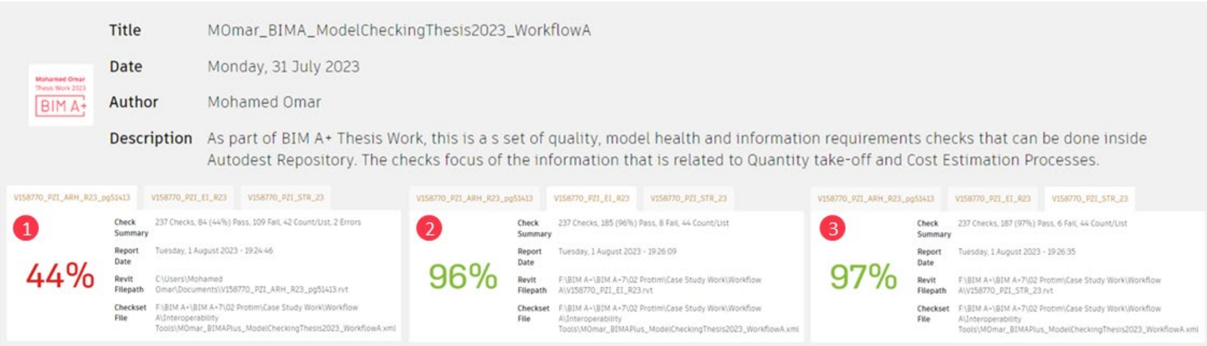


Figure 53: Check summary for architectural/ structural and MEP files

Example 1: General Check - In-place Families List

The outcome of this check comprises a list of In-Place Families, Shown below in Figure 54. It is essential to note that this result is not regarded as a failure. The textual rules explicitly define that these In-Place Families might be utilised in cases where calculations of area, volume, and tagging are unnecessary. As such, the check merely identifies the presence of In-Place Families, and further checks are designated to assess whether the specified conditions for their usage are satisfied.

1.1.g In-Place Families. - REQUIRED
COUNT and LIST of all in-place family elements in the model. In-place families can signifcantly impact model size and performance and should be used sparingly.
Count: 4

Family	Name
Furniture 1	Furniture 1
Logo	Logo
Walls 1	Walls 1
Walls 2	Walls 2

Figure 54: Check Results - In-Place Families List

Example 2: Geometrical check – Duplicated architectural elements

The outcome of this check is a list of all duplicated elements. As shown in Figure 55. Each element's GUID can be utilised to identify and examine the failed elements for necessary actions.

Example 3: Information Requirement check – Basic Walls Workset

The outcome of this check is a list of interior walls modelled in the wrong Workset, as indicated in Figure 56. Listed walls are modelled in the furniture Workset, while they shall be modelled in the Interior architectural Workset.

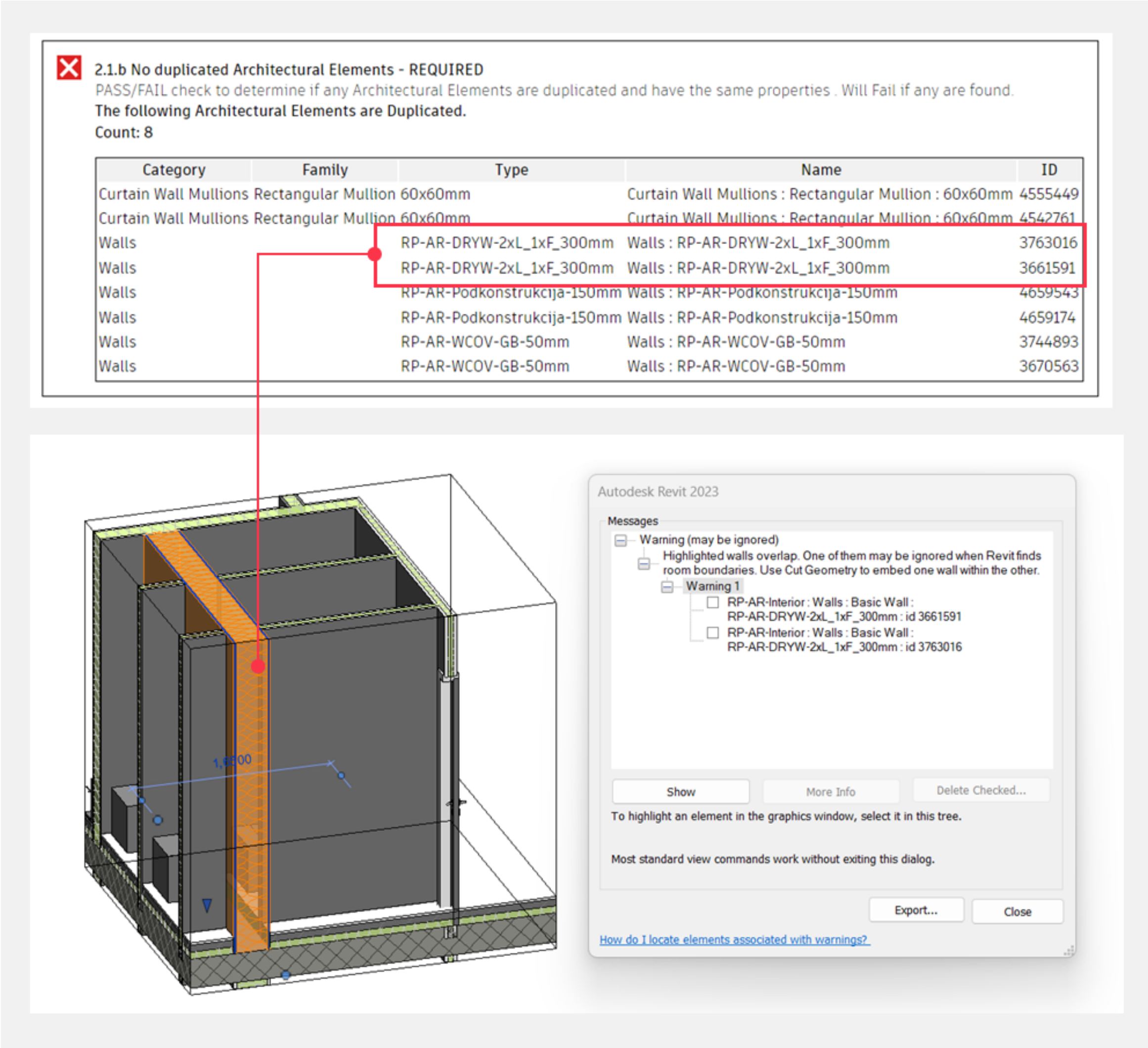


Figure 55: Check Results – Duplicated architectural elements.

3.1.7.1.c Basic Walls-Workset-Correct/Wrong Value - REQUIRED

PASS/FAIL check to determine if any Interior and Exterior Walls are not modeled correctly in Structural Workset. Will Fail if any are found in wrong workset.

The following Interior Walls Elements are not modeled in correct workset.

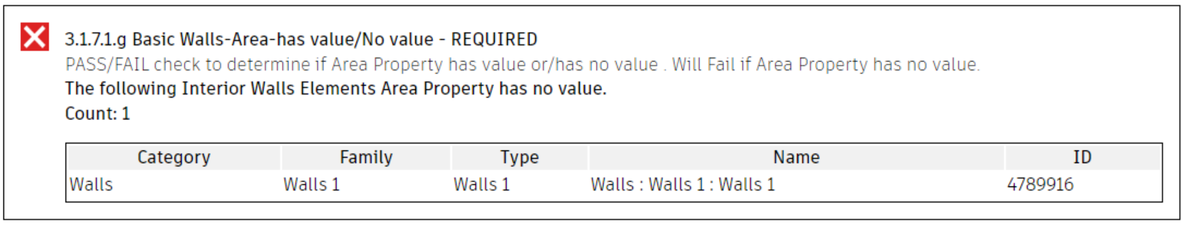
Count: 29

Category	Type	Name	ID
Walls	RP-AR-WCOV-GB-100mm	Walls : RP-AR-WCOV-GB-100mm	5133942
Walls	RP-AR-WCOV-GB-100mm	Walls : RP-AR-WCOV-GB-100mm	5133941
Walls	RP-AR-WCOV-GB-100mm_EI	Walls : RP-AR-WCOV-GB-100mm_EI	5139661
Walls	RP-AR-WCOV-GB-100mm_EI	Walls : RP-AR-WCOV-GB-100mm_EI	5139603
Walls	RP-AR-WCOV-GB-20mm	Walls : RP-AR-WCOV-GB-20mm	5198711
Walls	RP-AR-WCOV-GB-20mm	Walls : RP-AR-WCOV-GB-20mm	5198642
Walls	RP-AR-WCOV-GB-20mm	Walls : RP-AR-WCOV-GB-20mm	5198589
Walls	RP-AR-WCOV-GB-50mm	Walls : RP-AR-WCOV-GB-50mm	5344235
Walls	RP-AR-WCOV-GB-50mm	Walls : RP-AR-WCOV-GB-50mm	5344422
Walls	RP-AR-WCOV-GB-50mm	Walls : RP-AR-WCOV-GB-50mm	5344517
Walls	RP-AR-WCOV-GB-50mm	Walls : RP-AR-WCOV-GB-50mm	5344608
Walls	RP-AR-WCOV-GB-50mm	Walls : RP-AR-WCOV-GB-50mm	5345245

Figure 56: Check Results – List of basic walls with wrong workset

Example 4: Information Requirement check – Basic Wall Area Value

As indicated before, In-Place families might be utilised in cases where calculations of area are unnecessary. In the results found in Figure 57, this wall has no value for the area. In that case, the Area shall have a value calculated from geometry.


3.1.7.1.g Basic Walls-Area-has value/No value - REQUIRED
PASS/FAIL check to determine if Area Property has value or/has no value . Will Fail if Area Property has no value.
The following Interior Walls Elements Area Property has no value.
Count: 1

Category	Family	Type	Name	ID
Walls	Walls 1	Walls 1	Walls : Walls 1 : Walls 1	4789916

Figure 57: Check Results – List of the basic walls with No value for Area property

5.6.2 BMC Implementation of Workflow B

In this workflow, standalone software for checking processes is used. Thus, the optimal way to exchange information is to work with IFC as an exchange format between the repository environment and the checking software. In the case study, SMC is used as it has powerful abilities of geometrical and alphanumerical checks besides design checks.

5.6.2.1 Rules Interpretation

As delineated in Figure 58, Built-in Rules of SMC can be used for model validation and clash detection. Nevertheless, extending the scrutiny to encompass the geometrical issues for QTO and cost estimation for accurate results proves advantageous. The information requirements can be customised following the Level of Information Need defined in Appendix A.

These rules check that the model contains the required property sets and properties. It can also check that the properties have (or do not have) a value and that the value type is acceptable. It is possible to check the Basic regexes for textural properties.

Workflow A Checkset structure can be replicated similarly, wherein the requirement to assess each step discretely might not be imperative. SMC allows evaluating concurrently these steps as a cohesive unit, subsequently categorising identified concerns within the check results. Thus, the checks for each Ifc Entity are defined in one Check, as shown in Figure 59. However, Identifier values can be checked separately to check their consistency with the company naming convention system, as explained in Figure 60.

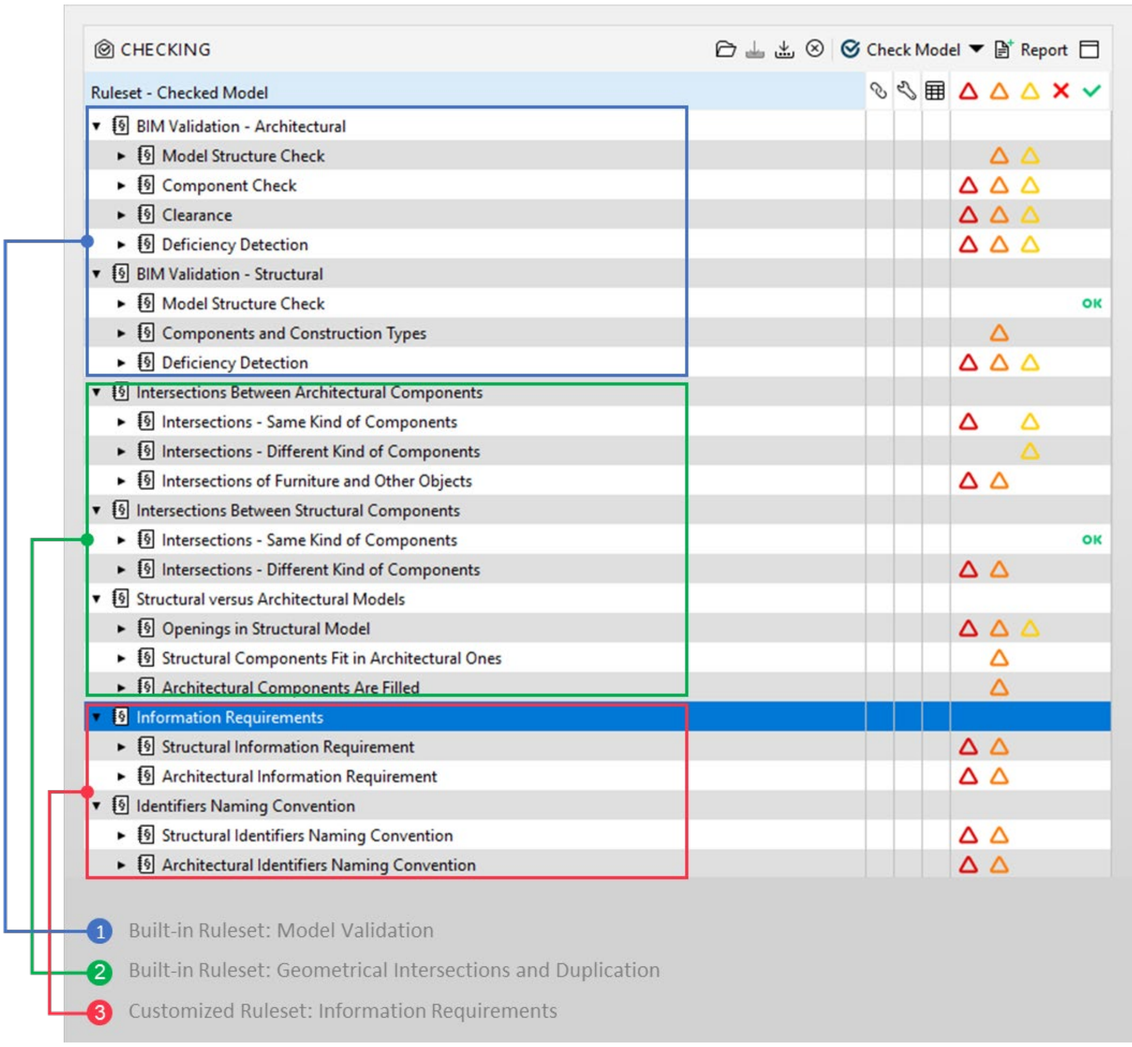


Figure 58: Workflow B - Ruleset structure

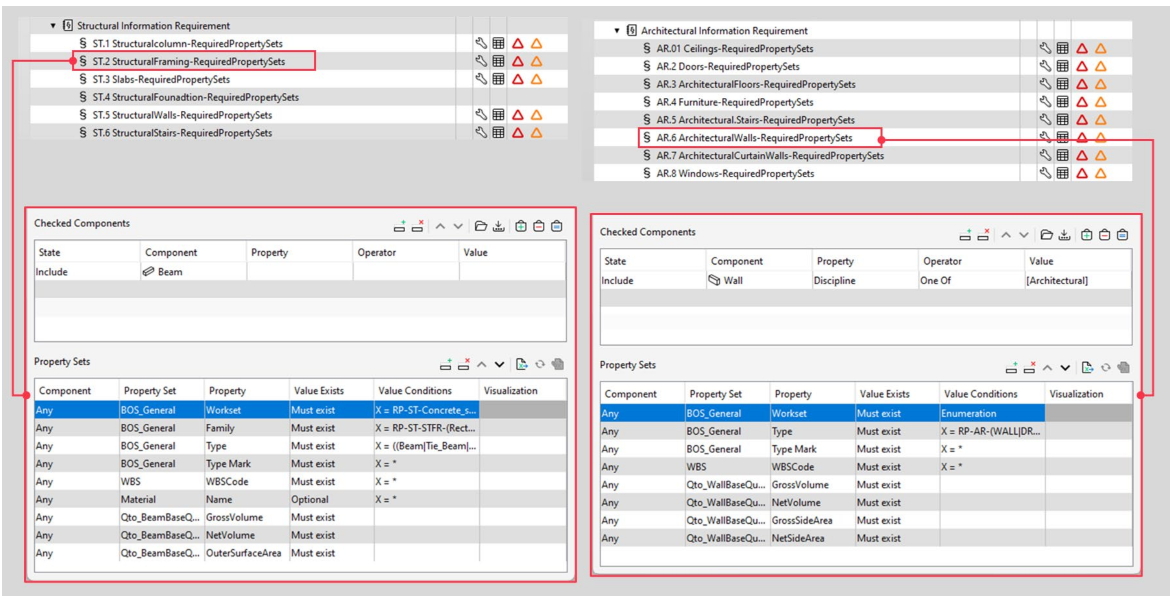


Figure 59: SMC information requirements rules

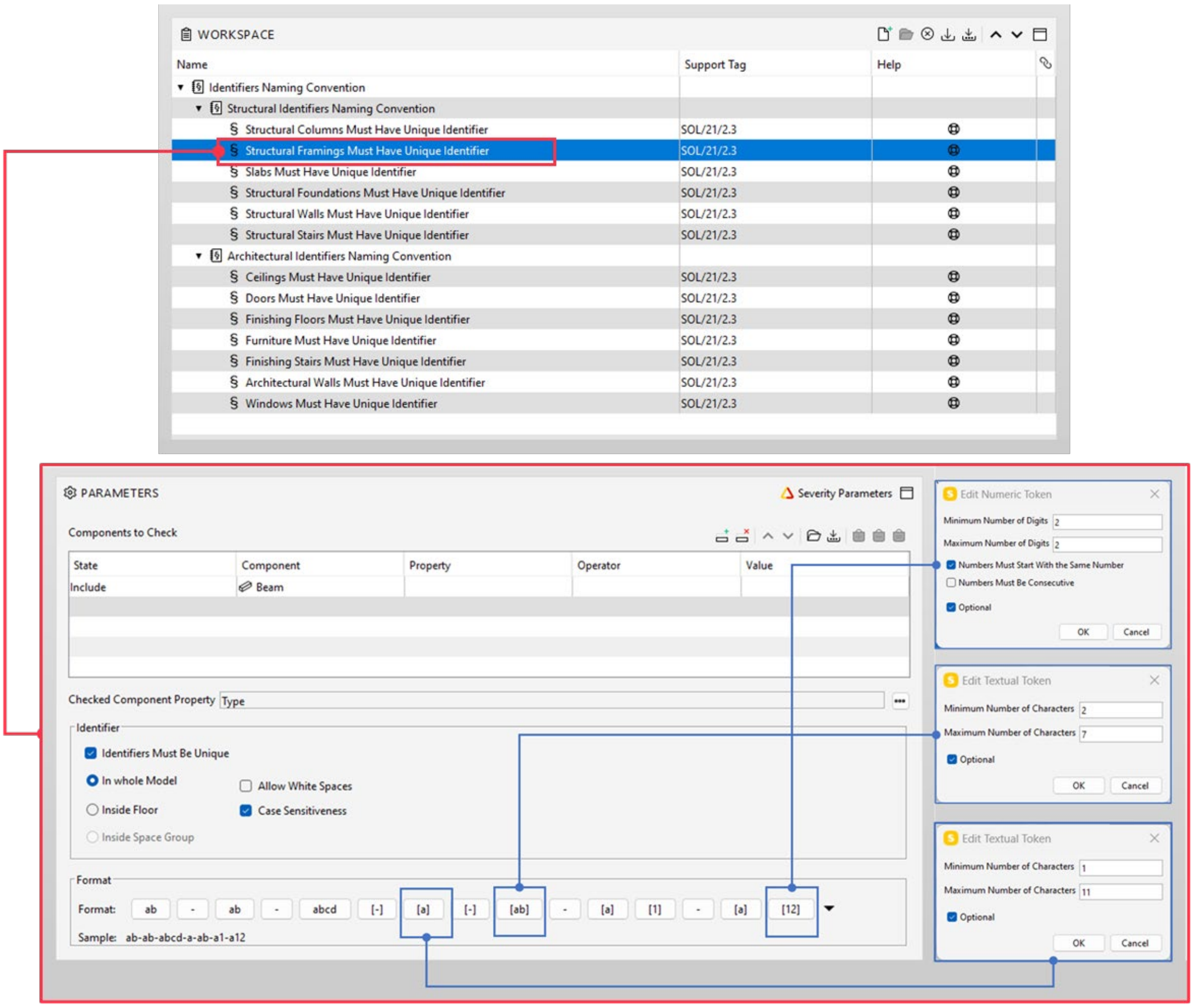


Figure 60: SMC - Identifier naming convention checking

5.6.2.2 Building Model Preparation

In this phase, each Workset of architectural and structural models is exported separately, using IFC 4, as shown in Figure 61. The needed properties of Family Name, Type Name, Workset, Type Mark, and Mark are mapped.

Materials are exported to IFC not as elements but as element properties. Material Data exported in IFC4 may have varying displays depending on the IFC viewer used, illustrated in Figure 62, as follows:

A- BIMVision and Navisworks: no Material tab; no such data displayed.

B- usBIM Viewer and Solibri: Material tab displayed; it contains material name only.

C- FZK Viewer and BIMcollab: Material tab displayed; data displayed.

Architectural Models and structural are opened in Solibri, where the checks are implemented, as shown in Figure 63.

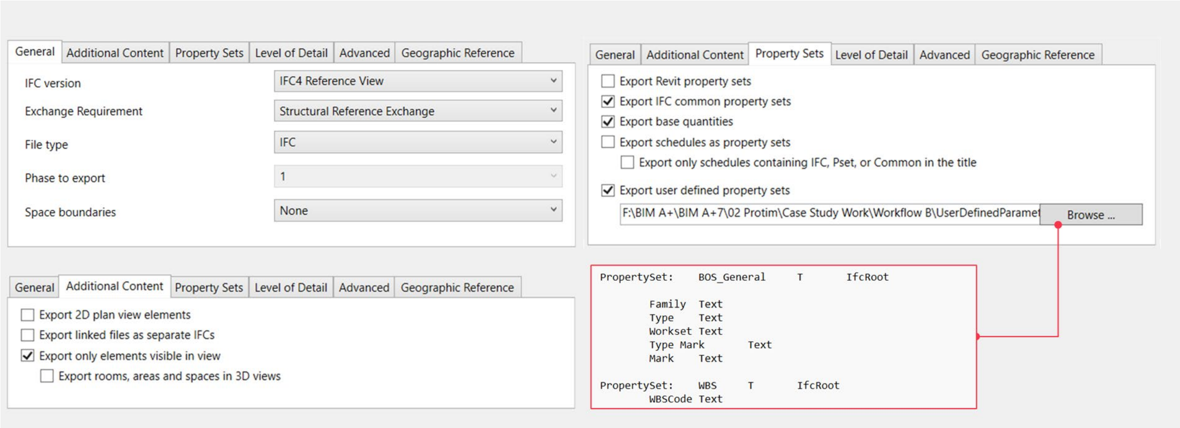


Figure 61: Case study IFC export setting

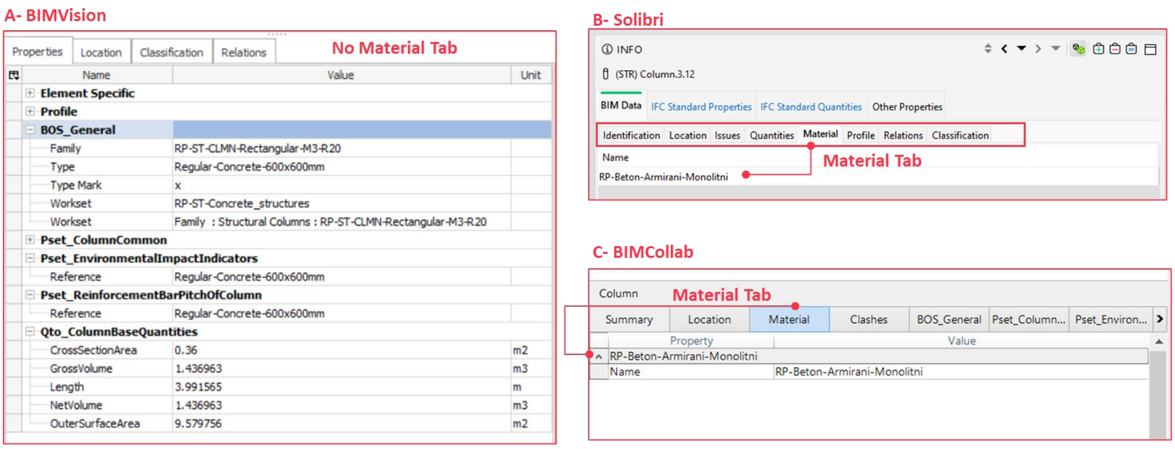


Figure 62: Material data exported in IFC4.

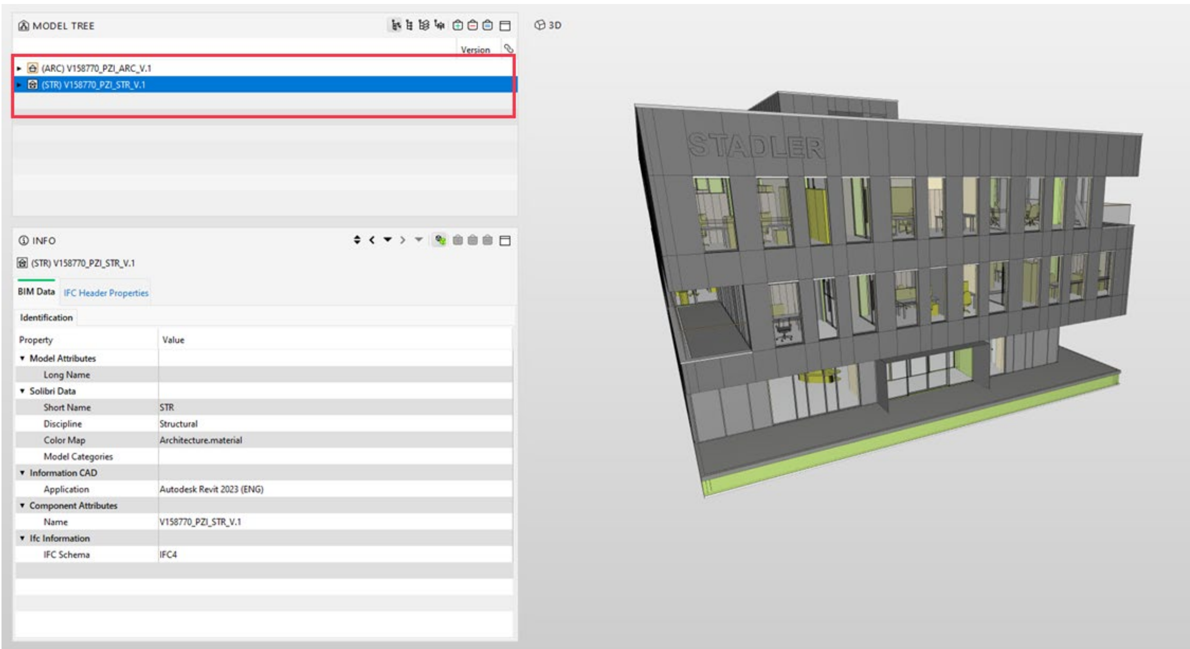


Figure 63: Case study - SMC model tree

5.6.2.3 Rules Execution

The execution is done inside SMC using the rulesets already prepared for the IFC Architectural and structural models. The model can be updated, facilitating the possibility of subsequent re-evaluation of the assessment.

5.6.2.4 Reporting Checking Results

Excel and PDF reports are generated. Also, it is possible to create issues from the check results and save it as BCF. It can be used to exchange issues with the repository, while the Excel results can be used to semi-automate some information requirements, as explained later. Below are some of the examples of the results.

Example 1: Model Validation - Doors and windows

This check results in the doors and windows related to different levels than its host. It may cause conflicts if some QTO or other information is extracted based on the level. As shown in Figure 64, the door is related to the structural level, while its host (the wall) is modelled on the architectural level.

Example 2: Model Intersection – Structural versus Architectural model

For this check, it results in the intersections (clashes) between Curtain walls and structural elements. Any hard clash may affect the geometry accuracy, which affects the QTO accuracy.

As shown in Figure 65, the beam intersects with the curtain wall, which needs to be modified by the design team, and this intersection affects the quantities for both Beams and Curtain walls.

Example 3: Information Requirements - Architectural Walls

This check results in the elements that do not meet the Information requirements defined. As shown in, there are three main types of issues as follows:

1- Missed Property Sets: No WBS Property Set for All Walls exists. So, it means that WBSCode Property is not defined for all architectural walls.
2- Missed Property: Some elements Do not have a Type Mark. Also, those elements' quantity properties of Area and volume are missed.
3- Not acceptable Property Values: Some elements have the wrong Workset. Some of the walls are modelled in the furniture Workset, while they shall be modelled in the interior Workset. Also, some elements do not follow the basic naming convention. Nerveless Naming convention can be double-checked from the identifiers' checks.

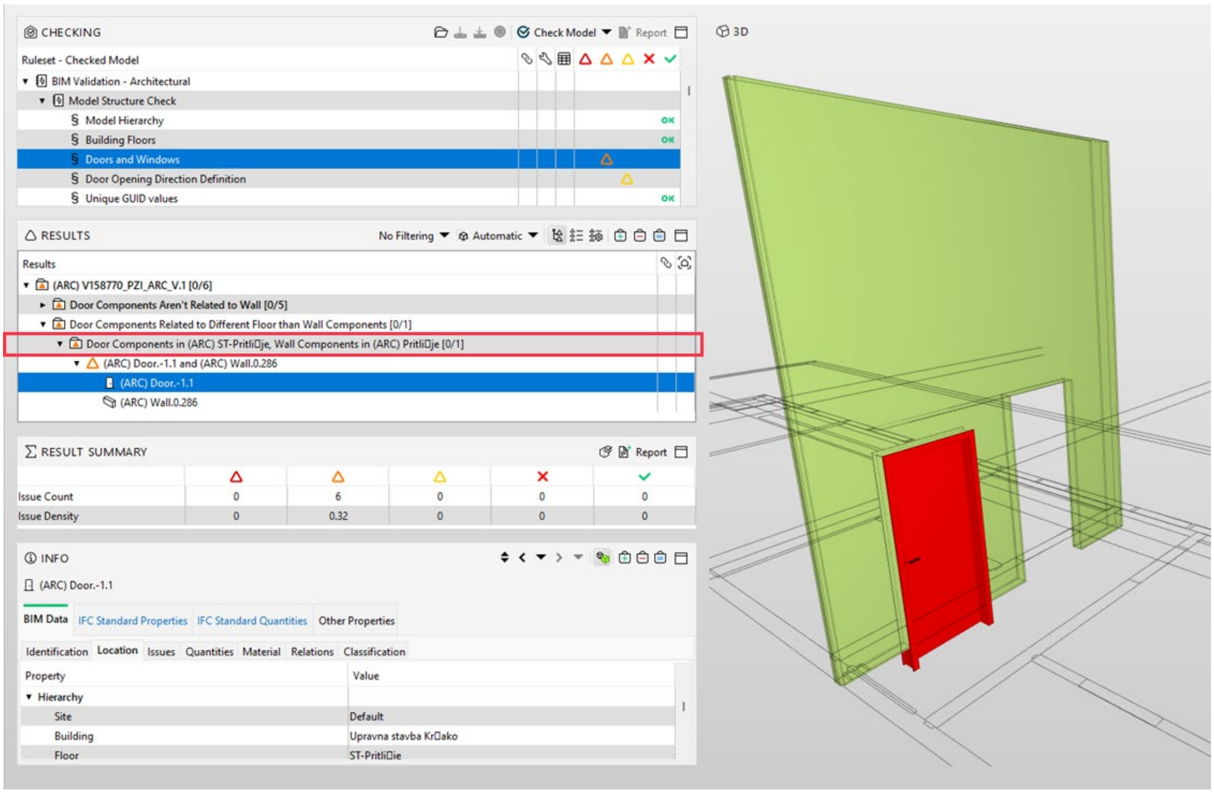


Figure 64: Check results – Model validation – Door and its host level

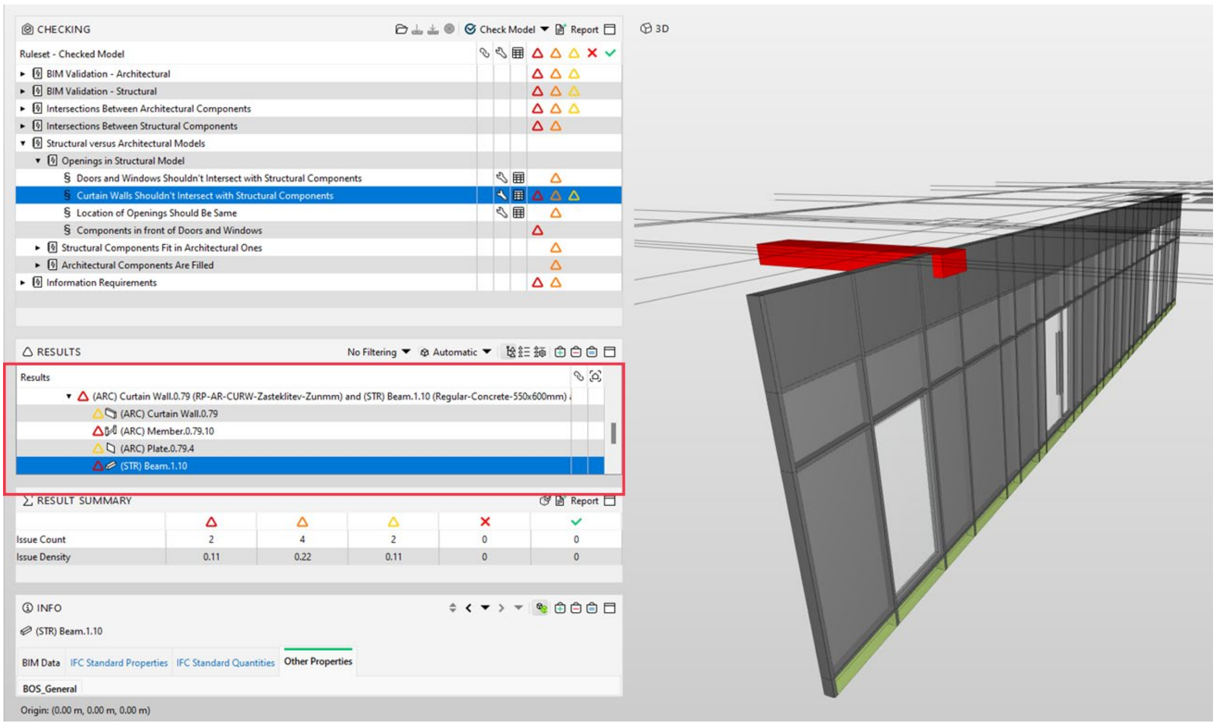


Figure 65: Check results – Intersection – Curtain wall Versus Beam

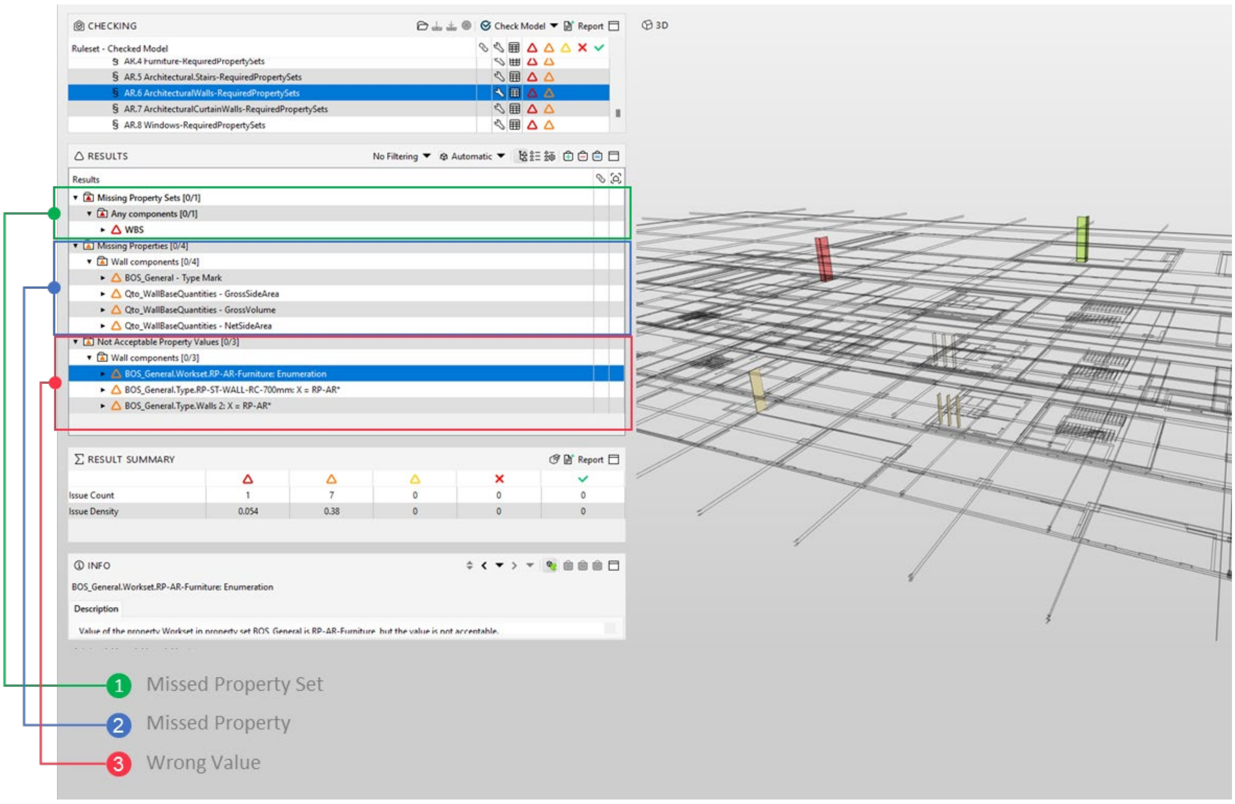


Figure 66: Check results – Information requirements – Architectural walls

5.6.3 BMC Implementation of Workflow C

This workflow embraces the utilisation of openBIM formats, accommodating various openBIM software platforms supporting IDS without imposing any restrictions. The BIM model is exchanged through the IFC format, while the rules are encoded within IDS file(s). As previously mentioned, IDS cannot be used to require particular geometry. Consequently, the primary emphasis within this workflow is directed towards the Level of Information Need.

5.6.3.1 Rules Interpretation

IDS files are created for architecture and structural models to check the alphanumerical information. IDS Converter is used to create the base IDS as an IDS converter makes an IDS with simple specifications capable of indicating which properties and values to be checked. However, further developments are edited in XML to add more constraints to the applicability, as shown in Figure 67.

Also, usBIM.IDS Editor can be used for editing IDS specifications in both readable format and computational tabs, as shown in Figure 68.

```xml
<specification name="AR.6.a ArchitecturalWalls-TypeName" ifcVersion="IFC4"
 description="PASS/FAIL check to determine if any Architectural Walls Type Names are not following the Company Naming convention.
  Will Fail if any are found." minOccurs="0" maxOccurs="unbounded">
    <applicability>
        <entity>
            <name>
                <simpleValue>IFCWALL</simpleValue>
            </name>
        </entity>
        <property measure="IfcBoolean" minOccurs="1" maxOccurs="unbounded">
            <propertySet>
                <simpleValue>Pset_WallCommon</simpleValue>
            </propertySet>
            <name>
                <simpleValue>LoadBearing</simpleValue>
            </name>
            <value>
                <simpleValue>No</simpleValue>
            </value>
        </property>
    </applicability>
    <requirements>
        <property measure="IfcText" minOccurs="1" maxOccurs="unbounded">
            <propertySet>
                <simpleValue>BOS_General</simpleValue>
            </propertySet>
            <name>
                <simpleValue>Type</simpleValue>
            </name>
            <value>
                <xs:restriction base="xs:string">
                    <xs:pattern value="RP-AR-(WALL|DRYW|PEDW|PARW|WCOV|WFIN|WINS|CURW|STFC|WFAC|FCOV|CLNW|CLNC|CASC|LUVR)(-(RC|CB|
                    AC|WD|SN|GB|BR|PT|GL|CR|MM|PL|TI|FO|FI|PS|AP|HI|PI|TX))??((-|_)[0-9]*x(RC|CB|AC|WD|SN|GB|BR|PT|GL|CR|MM|PL|TI
                    |FO|FI|PS|AP|HI|PI|TX))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PIR|_MDF|_ACS|_ACM|_BTM|_STF|_PVC|_AQU|_VIS|_WTR|_HPL)??
                    (-|_)[0-9]*mm" />
                </xs:restriction>
            </value>
        </property>
    </requirements>
</specification>
```

Figure 67: IDS specification of Type Name in XML editor

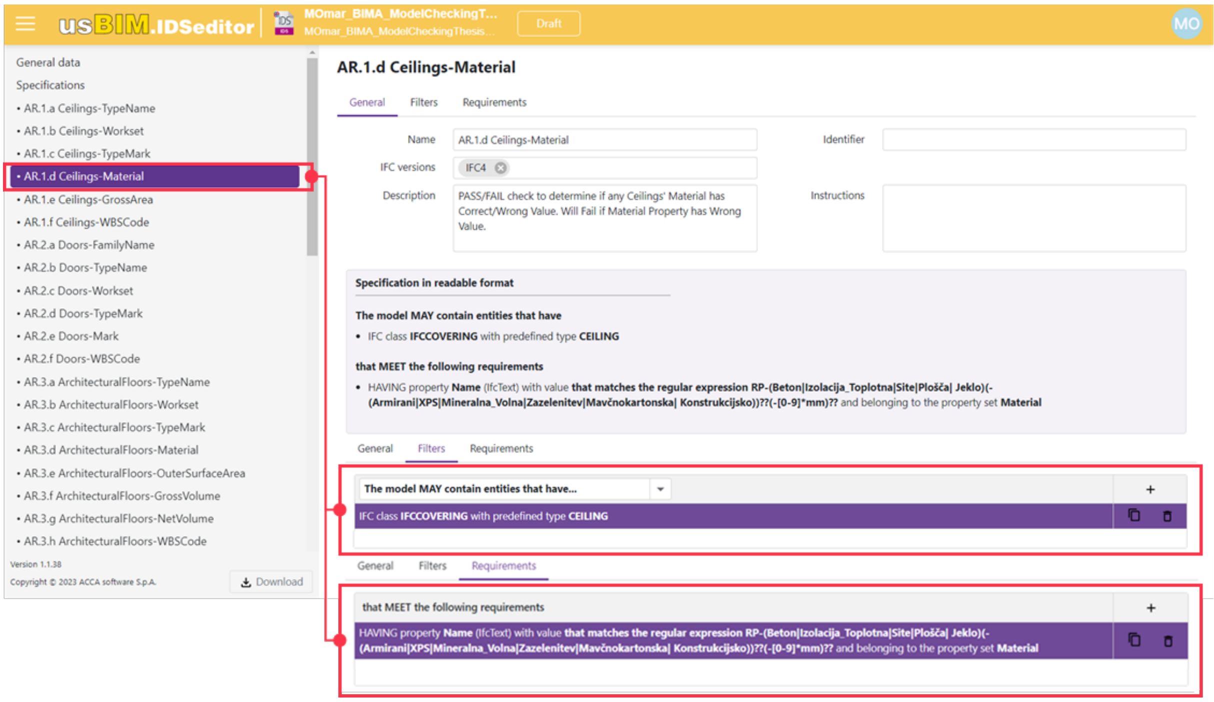


Figure 68:usBIM.IDS Editor- Specification editing

5.6.3.2 Building Model Preparation

IDS can be checked against IFC files. Thus, the same IFC files exported and utilised in Workflow B (following the identical export configuration as depicted in Figure 61 and Figure 62, Pg.64-64) are used in this Workflow.

5.6.3.3 Rules Execution

The rules are executed using Blender, employing the IFC Add-In within the software. The verification of IDS against the IFC files was conducted seamlessly, eliminating the necessity to open either the IFC files or the IDS manually. This process of execution is illustrated in Figure 69. Also, usBIM.IDS Validator or any other software supporting IDS can be used for rule execution. In the usBIM.IDS case, the IFC file is opened and then validated against the IDS file, as shown in Figure 70,

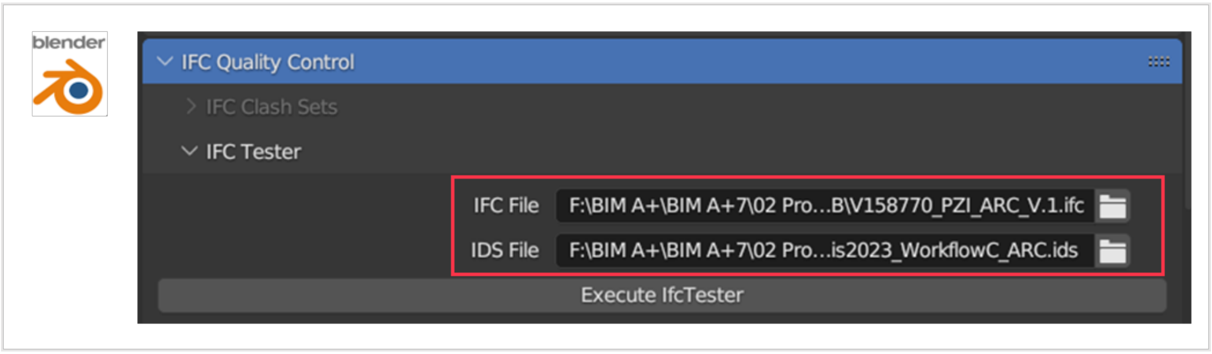


Figure 69:Workflow C Execution using Blender IFC Add-In

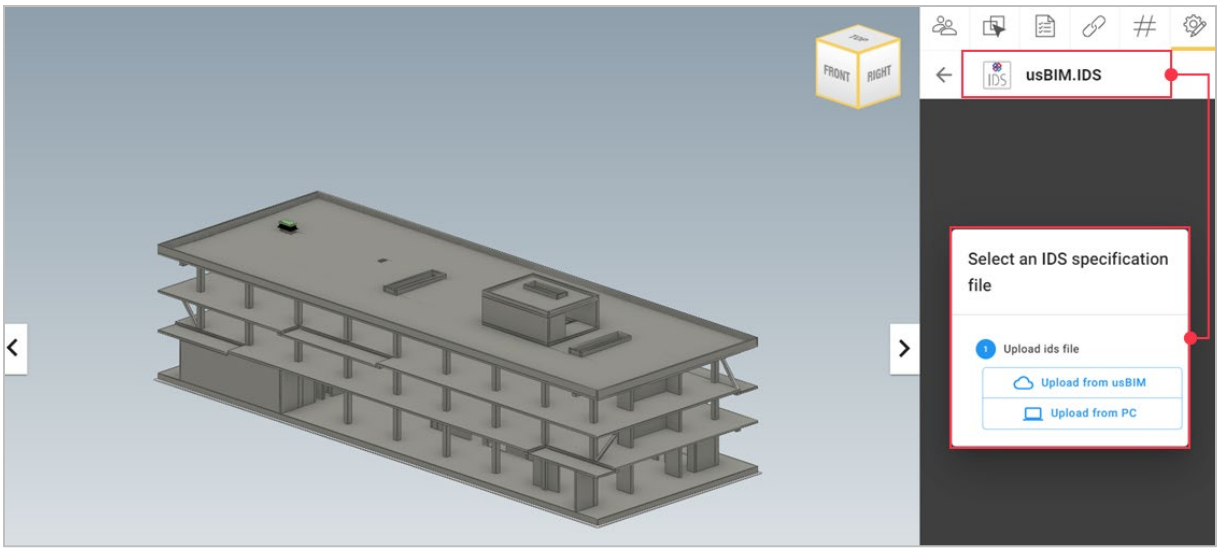


Figure 70: Workflow C execution using USBIM IDS Validator

5.6.3.4 Reporting Checking Results

Depending on the software used, the results could be exported as HTML, Word or other formats. For the case study, Both Blender and usBIM are used. For example, for the specification of Architectural wall Worksets, a list of elements with their GUID resulted in the HTML report, as shown in Figure 71.

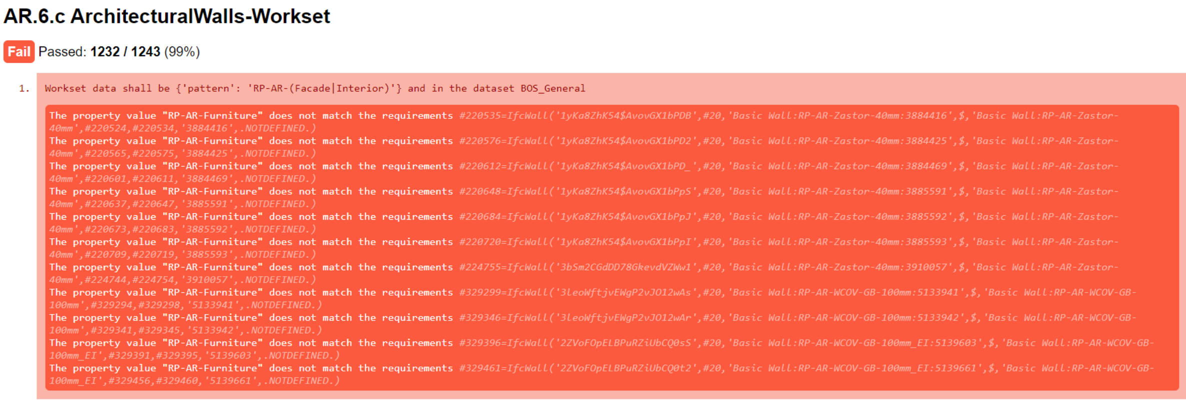


Figure 71: HTML result of IDS checking using Blender

As described in rules execution, it is also possible to open IFC using usBIM and with usBIM.IDS Validator, Checks can be executed and directly navigate to the issues and edit in IFC. As illustrated in Figure 72, the Elements can be selected and edited in place in the IFC file.

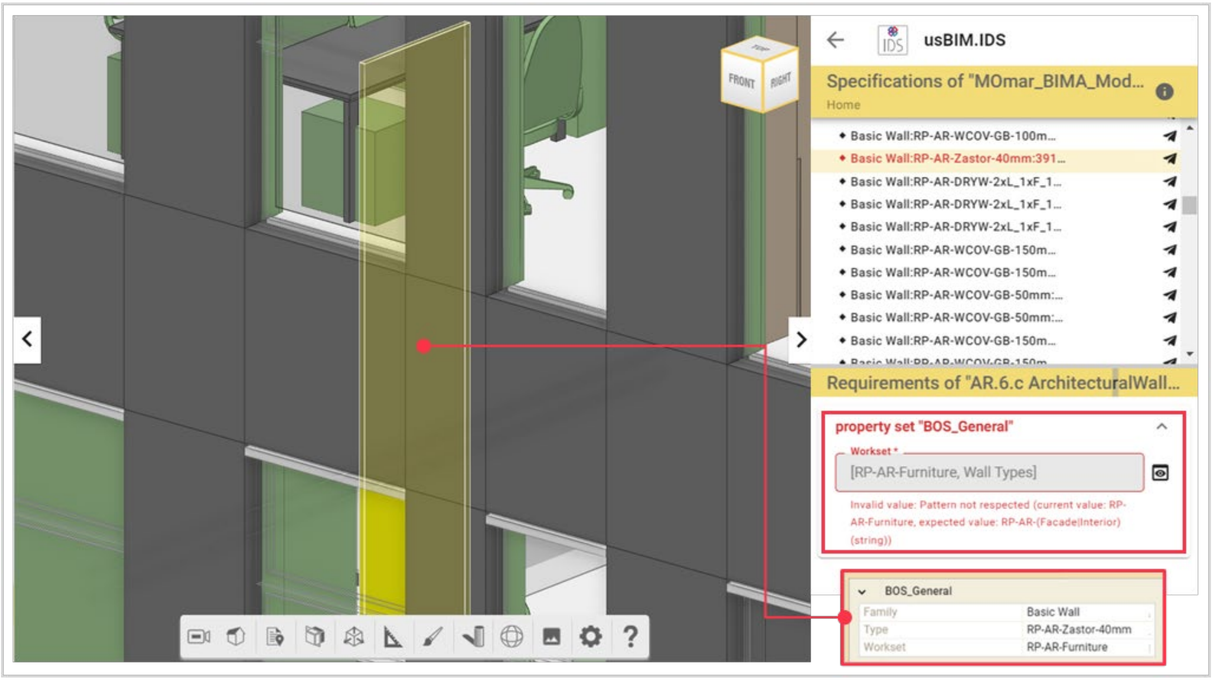


Figure 72: usBIM IDS Validator - Architectural wall Workset property

5.6.4 Automated/Semi-Automated Codes Solving Resulted Information Issues

Various automated and semi-automated add-ins, such as Diroots, can streamline the alteration of information following the checking and verification process. Also, some dynamo codes have been devised to expedite these procedural tasks as follows:

A- Elements Selection by check code / Keyword:

This Code facilitates the grouping selection of elements that yield identical verification outcomes by utilising a designated check code or keyword. As shown in Figure 73, the code extracts the data from the Excel report and then groups elements with the check code or any word key.

For example, elements that resulted as duplicated architectural elements (Figure 52, Pg.,58) can be selected by check code “2.1.b”. Another example, all elements that have issues related to Workset can be selected by checking the keyword “Workset”. As illustrated in Figure 72, Dynamo Player is used to automate the whole selection process.

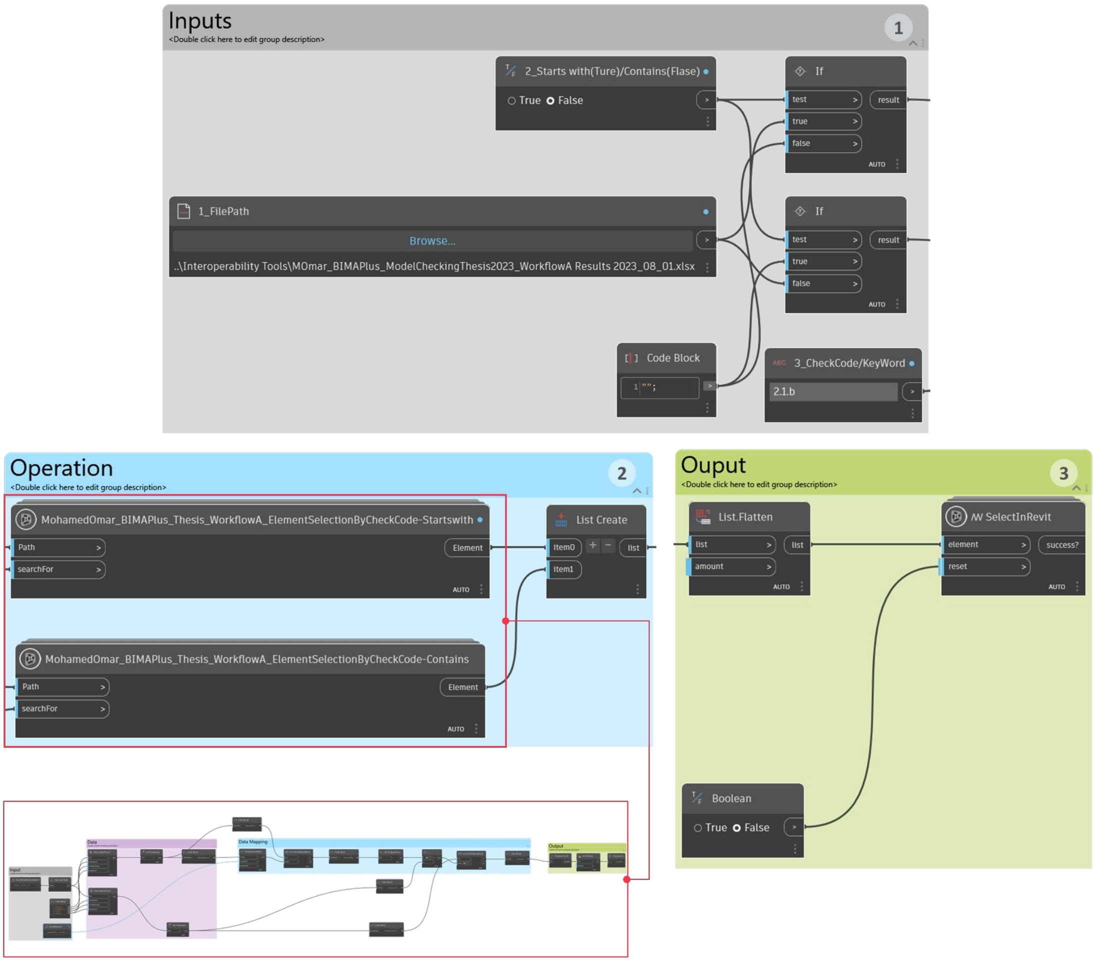


Figure 73: Dynamo code A - Element Selection by Check Code or Keyword

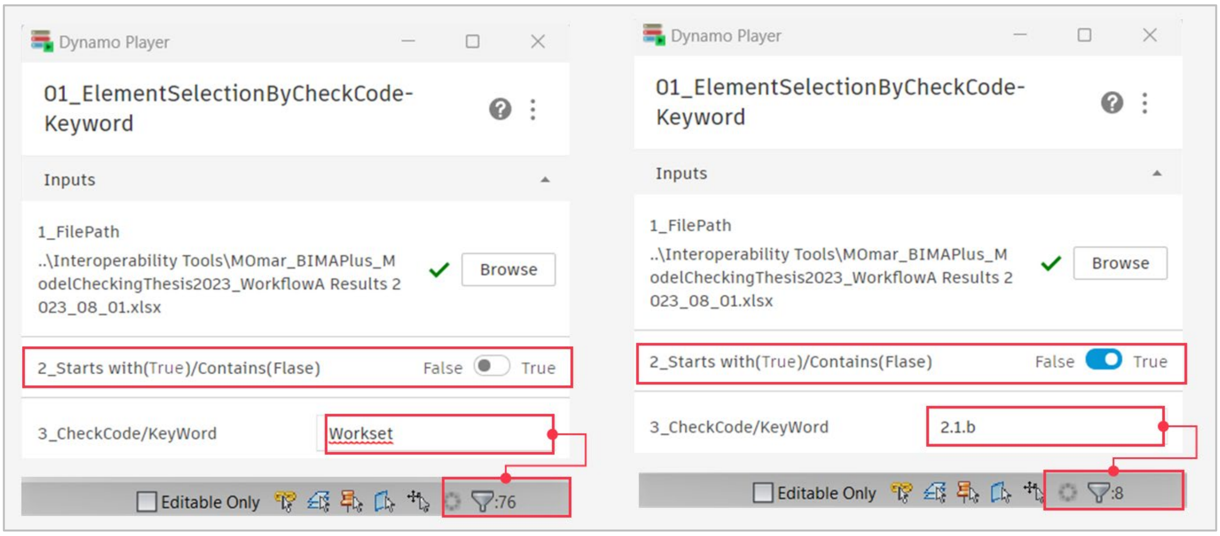


Figure 74: Dynamo Player – Using check code or keyword to select elements

B- Delete Duplicated Elements:

To regulate the elimination of duplication elements while ensuring the retention of essential information, a comparison of dissimilar values is conducted to make informed decisions. A Dynamo code is formulated to generate a report about these parameters to semi-automate the process of dealing with duplicated elements.

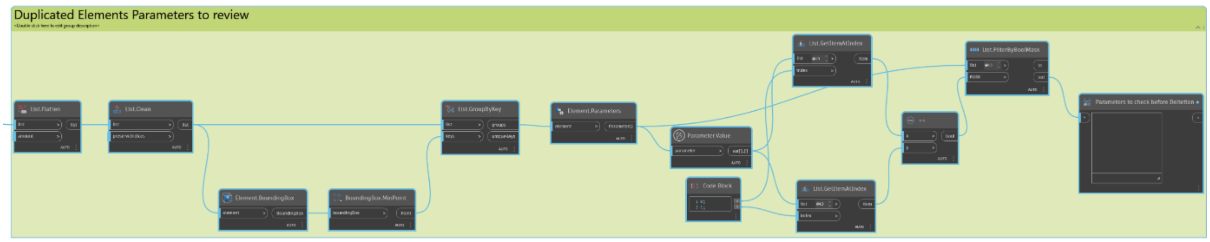


Figure 75: Dynamo Code B - Compare duplicated elements properties

C- Editing Parameter Values:

A dynamo code has been developed to automate the procedural manipulation of alphanumerical information within the model, as exemplified in Figure 76, with the principal objective of adding prefixes or replacing values to the associated parameters. This dynamo code is confined to parameters wherein the property IsReadOnly is set to False. Nevertheless, the prospect exists for creating additional dynamo targeted towards Family Names and Type Names.

Furthermore, pursuing this objective may be complemented by utilising the Diroots Add-In, thereby enabling parameter manipulation through the sequential employment of the Selection Code and, subsequently, the Diroots Add-In toolset.

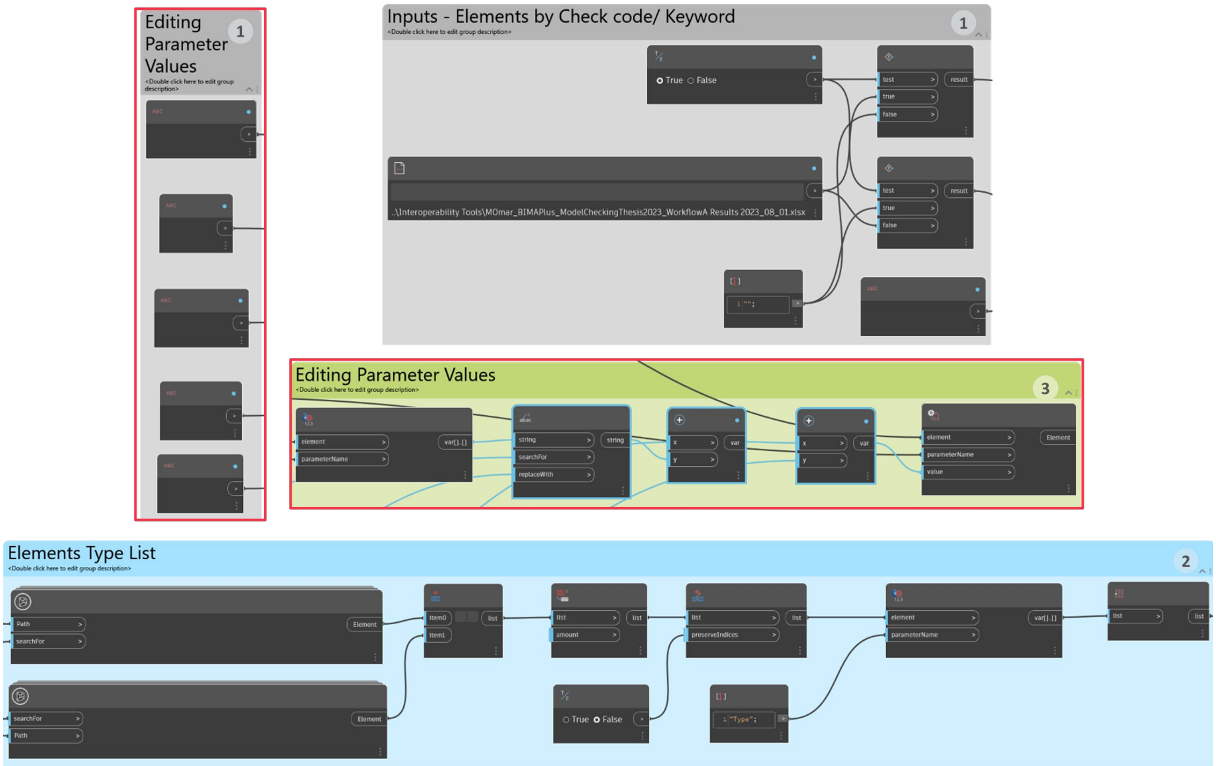


Figure 76: Dynamo code C – Elements Parameter Values Editing

For instance, to modify the Family name, Type Name, and Type Mark Properties within the Windows Category as reported in the checking report, the Dynamo code is employed to alter the Type Mark, as depicted in Figure 51. Similarly, to revise the windows' Family Name and Type Name, the selection Dynamo code displayed in Figure 54 is utilised to pinpoint the concerns, subsequently isolating them within the view. Following this, the Diroots Add-In is enlisted to effect changes to the family name and type name, as elucidated in Figure 55.

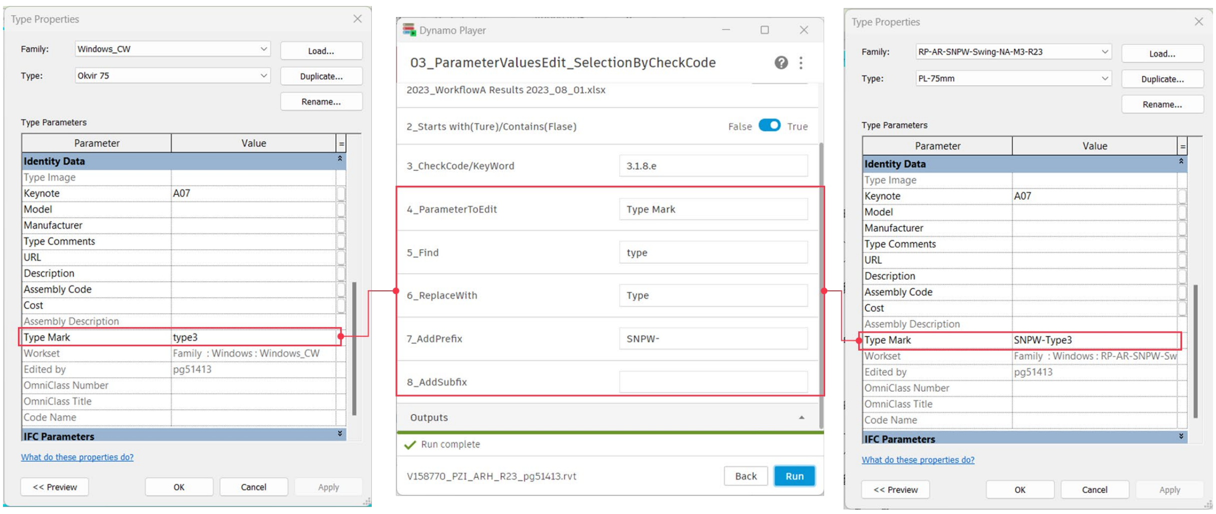


Figure 77: Dynamo Player – Using edit parameters to edit the Parameter value

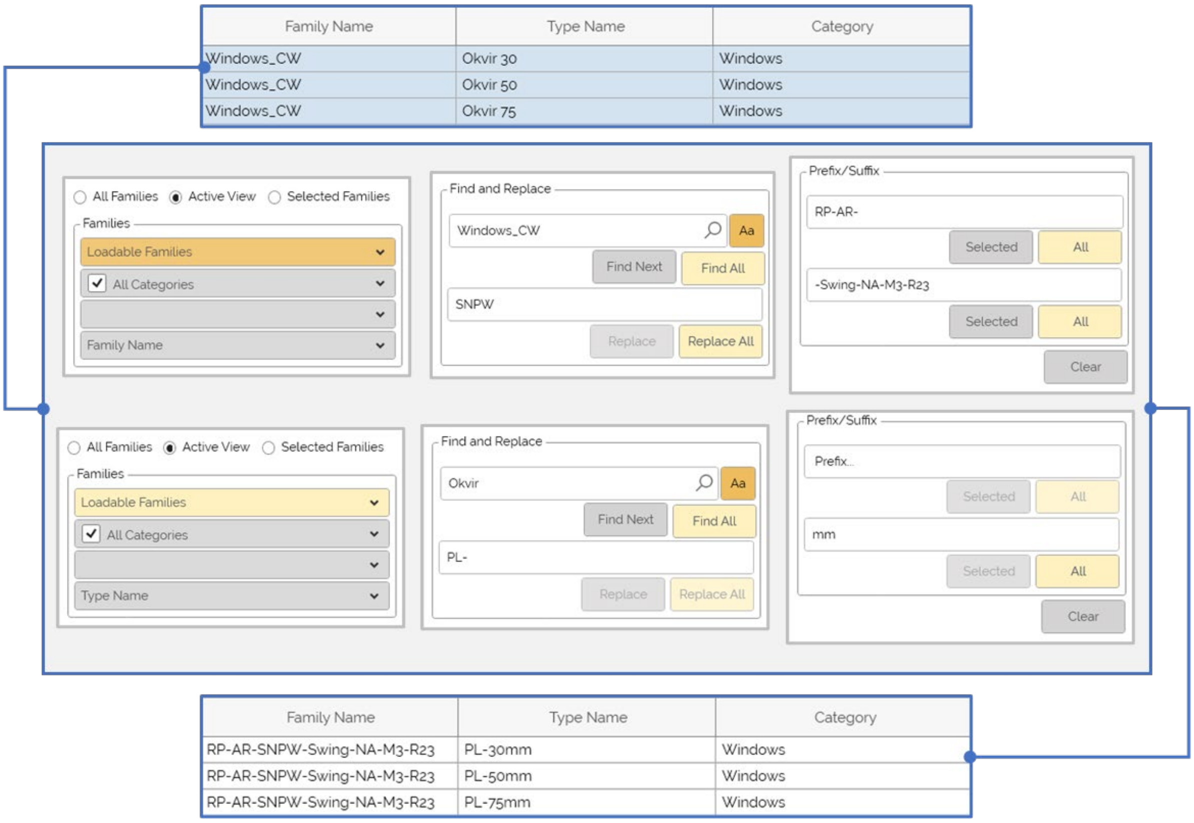


Figure 78: Diroots Add-In to edit Family Name and Type Name

6 DISCUSSION

This chapter includes an overview of implementing the information requirement framework of the Level of Information Need. Also, an overview of the workflows implemented throughout the analysis of case study results, a discussion about each workflow limitation and opportunity, and an extensive comparison of the used software are covered.

6.1 Level of Information Need Framework Implementation Discussion

In this section, a discussion is aimed at analysing the implementation process of the Level of Information Need framework in the case study. The Level of Information Need is different for each project and phase and must be updated continuously throughout the whole lifecycle of the project. While the framework assesses most of the information requirements in the EIR, other upper-level information requirements, such as OIR and PIR, should be considered.

The case study focuses on specific BIM uses (QTO and cost estimation). On the project level, the Level of Information Need framework shall be defined for each BIM use and afterwards be complied with other frameworks of Level of Information Need defined for other BIM uses. For instance, while the Railing element could have 2D dimensionality in the framework defined for QTO, it shall be 3D dimensionality for the Clash detection purpose. So, A complied framework for the information requirements shall be updated upon all BIM uses.

Regarding the geometrical representation in the Level of Information Need framework, the LOD specification concept was used to give an idea about the geometrical representation using the LOD 100,200,350,400. However, it is recommended for other BIM uses where the geometrical representation affects the BIM use to be customised for each category.

6.2 BMC Workflows Review

While each workflow was implemented for specific rules, this section explores a more general review of each workflow, assessing the limitations and opportunities for each workflow and the possibility of mixing the usage of more than BMC in the same project.

6.2.1.1 BMC Workflow A Review

The whole process was done inside the repository throughout the implementation of workflow A. The alphanumerical information was mainly translated using the model checker configurator as one of Autodesk's Intermobility tools.

It is faster to manipulate Checkset in XML for editing and creating similar rules. As a concept, regexes are not limited to checking the naming convention but can facilitate the editing process. For instance,

the check IDs shall be removed to copy a check, section or heading using XML. Deleting the IDs individually is time-consuming in a large ruleset such as this case study, which consists of more than 500 checks. In that case, developing a regex for XML manipulation to select All IDs at once was done. As an Example: Check ID sample: ID="e1a000e0-dc8f-45bd-9041-02720bdc549d" - regex: `ID="[a-z 0-9 -]*`”). By this simple regex, All IDs for Checks, sections, and Headings could be selected (in the case study, 1469 IDs was selected inside XML by one-click).
Furthermore, some rules could not be translated as Autodesk Interoperability Tools do not support accessing the API parameters of properties. For instance, one of the rules states: “Location: The project base point shall be located at the intersection” of grids one and A”. In that case, VPL, such as Dynamo, can be used to implement the rule, as shown in Figure 79. Thus, combining Autodesk Interoperability Tools with Dynamo covers more rules to be interpreted by using Dynamo for preparing some semi-automated codes to solve the resulting issues.

Nevertheless, A critical risk that shall be mentioned is the risk of automating the process of solving the resulting issues, as it might cause information loss. For example, while it is possible to delete the duplicated elements automatically, the suggested approach was to report the difference between them in terms of information included in each. Subsequently, the final decision will be for the BIM model manager/ coordinator to take the proper action. Another proposed approach for Automated codes that edit the property values is to export the current values with data in a CSV file for archiving purposes.

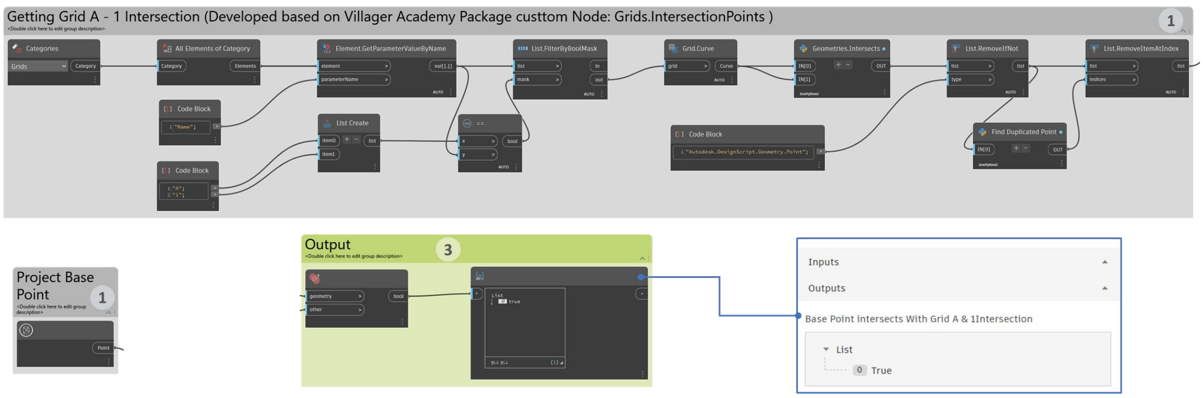


Figure 79: Showcasing of using VPL (Dynamo) in the rule interpretation stage

6.2.1.2 BMC Workflow B Review

Using standalone software designated for the BMC process gives many capabilities and functionalities that enhance the BMC process. As an observation of the software used in this workflow, SMC could be selected in the design phase because of its capabilities of geometrical checking of clearance and accessibility and clash detection besides information consistency. Bexel Manager would be more beneficial for the construction phase, considering the capabilities of Bexel Manager to integrate time and cost with more extensive capabilities to check alphanumerical information.

Furthermore, Other factors of software license cost, users' software capabilities and interoperability with other software used for other BIM uses of QTO, Cost estimation, clash Detection, Solar analysis, Structural analysis, etc., shall be considered.

Throughout this workflow implementation, the information exchange is very critical as the whole process of checking is done outside the repository. Incorrect or incomplete export settings will cause information loss, giving wrong results. Using openBIM formats in exchange provides more flexibility to exchange information and interoperability with various software. For instance, Files from Autodesk, Tekla, and ArchiCAD can be checked simultaneously with the same Checkset using the designated software and issued back to each repository using BCF.

However, a critical concern in this workflow is that the checking process is done through black-box tools that usually use rule templates based on hard-coded and inflexible rules.

6.2.1.3 BMC Workflow C Review

As a main principle of this workflow, it is aligned with openBIM as a vendor-neutral collaborative process. With IDS, the Interpretation of rules is done with a high potential to define information requirements and link to bSDD. It is now aligned with the ISO 19650 framework and is expected to be a formal buildingSMART standard soon.

IDS provides flexibility in the checking process except for the last part of the resulting issues. The explored tools issue HTML and Excel reports; other software support to issue BCFs are not free software, which does not align with the core idea of this workflow. However, using Python to navigate the result inside the repository or even create BCFs is possible. On the other hand, a limitation of the IDS Geometrical checks is not possible to be checked by IDS.

Great potential is expected as IDS Editing and Validation will be supported with other software such as BIMcollab, giving a chance to merge the usage of workflow B and C for enhancing the BMC process.

6.2.1.4 BMC Workflows SWOT Analysis

As previously discussed, each workflow has its uniqueness, and while the sequence of implementation is similar, each workflow has its opportunities and limitations. A SWOT[1] analysis is conducted, as explained in Table 10, to get better abstract knowledge for each workflow.

---

[1] SWOT analysis, an acronym for Strengths, Weaknesses, Opportunities, and Threats, is a strategic tool to assess a studied subject's internal and external environment. This simple yet powerful framework identifies strengths and weaknesses, such as resources, capabilities, limitations, and external opportunities and threats in the industry. By conducting a SWOT analysis, companies gain valuable insights to inform their decision-making processes, helping them leverage their strengths, address weaknesses, seize opportunities, and mitigate potential threats.

Table 11: BMC workflows SWOT analysis

Workflow A - Inside Repository	
Strengths: - No need to exchange the model in other formats (using the RVT model directly). - All BMC steps can be done inside the repository. - Checking rules can be edited in XML. - Supporting regex.	Weaknesses: - Cannot access API Parameters of properties. - Limited capabilities of geometrical checks and code validation. - Limited to the RVT files (or the repository file extension). - Does not support using data templates to create the rules. - To navigate the results, some rules may be partially duplicated (for instance, to check Property exitance and value, it should be defined as two rules to be able to define the problem).
Opportunities: - using VPL (Such as Dynamo or Grasshopper through using Rhino inside Revit) in complex Rules. - Using Plugins and Add-In (such as Diroots and conVoid) to automate the process of solving resulting issues. - More Interoperability Tools will be realised by Autodesk (Shared Parameters Tools, Validation Tool for Docs).	Threats: - Interoperability with other software/tools. - Instability of New version of some Autodesk Interoperability Tools such as Model Configurator. - Risk of stopping support to Autodesk Interoperability Tools.
Workflow B – Standalone Software	
Strengths: - the variety of software solutions in the market (SMC, Bexel Manager, BIMcollab, …). - Supporting openBIM exchange formats of IFC and BCF. - In SMC, there is an ability to customise rules besides the predefined rules. - powerful capabilities of geometrical checks, accessibility, clearance, and information consistency.	Weaknesses: - Some Software does not support all regexes patterns (for instance, SMC does not support all regexes patterns). - Analysing and group issues before exchanging them, as BCF is time-consuming. - not working on the original file (for instance, RVT) but working on IFC, which makes some rules challenging to check.
Opportunities: - Using BCF for exchange issues with the possibility of using VPL to semi-automate the solving process. - using SMC data templates and the possibility of linking them with company templates (such as naming convention system).	Threats: - black-box approach (using rule templates based on hard-coded and some inflexible rules).

Continued: Table 11: BMC workflows SWOT analysis

Workflow C – OpenBIM	
Strengths: - white-box approach using openBIM formats with no flexibility to use any software/vendor. - IDS is both computer interpretable and human-readable file, which makes the rule interpretation phase clear. - great functionalities to check IFC attributes (`"Name"`, `"Description"`), `IFCtype`, properties (`"Pset"`), quantity, classifications, composition, and materials. - supporting regexes.	Weaknesses: - Complex rules cannot be implemented. - not possible to do geometrical checks. - Exchange of issues using BCF is not supported in some openBIM software. - not working on the original file (for instance, RVT) but working on IFC, which makes some rules challenging to check.
Opportunities: - Many software vendors will be supporting IDS, giving the chance to generalise the use of IDS for all BIM uses.	Threats: - Poor information exchange quality (in exporting IFC from the repository) may lead to information loss and wrong results.

6.3 Software Review

Throughout the study, various software /tools from various vendors were used to explore BIM uses and implement the whole process of BMC, including rules interpretation that was created and defined based on the information requirements, model preparation, rule execution, and reporting. The case study showed in-depth implementation challenges during BMC's steps using those software/tools.

6.3.1 Autodesk Interoperability Tools Review

Autodesk Interoperability Tools are free tools for Autodesk Revit users, aiming to help with different BIM workflows. Within the repository, two noteworthy tools, the Autodesk Model Checker Configurator and Autodesk Model Checker for Revit, were instrumental in implementing Workflow A. These tools feature an intuitive user interface, offering users the flexibility to employ pre-built checks or customise rules through the use of wizard checks and advanced checker builders. They include a repository of libraries, such as Revit Best Practices, Penn State (PSU) - BIM Standards, and Dutch Revit Standards (NLRS), which can serve as valuable references.

These tools demonstrate exceptional prowess in conducting quality assessments and assessing file health, including verifying Revit parameter existence (defined or undefined) and their corresponding values, with the option to utilise regexes. However, it is important to note that each check can accommodate only a single rule. For example, when examining the presence of a shared parameter such as WBS Property, as outlined in Scenario 1(Scenario explanation in Pg.34), two distinct approaches can be adopted:

The first approach involves consolidating multiple rules into a single check, evaluating property existence, property set, data type, presence of a value, and the correctness of the value (matching the regex). The outcome is a binary pass or fail result with textual description, without detailed information on which specific rule(s) failed. Alternatively, the second approach defines each rule as a separate check, creating five distinct checks for each parameter. While this approach provides granular insights into the exact nature of the issue, it may lead to issues being duplicated, potentially resulting in an increased number of redundant notifications.

Additionally, the model checker tools are limited in geometrical checks. They are not designed for clash detection but are useful in alphanumeric information assessments. A notable advantage of these tools lies in their capacity to execute checks, or portions thereof, across multiple RVT files without opening each file individually. For instance, when opening an architectural file, the checks can be applied to all other project files, including structural and MEP, streamlining the process of evaluating overall quality and adherence to standards even though many projects simultaneously can be checked.

6.3.2 SMC Software Review

As one of the pioneering software applications developed for model-checking processes, SMC is a robust and reputable tool renowned for its versatile capabilities in conducting various BMC processes. SMC has earned widespread recognition for its proficiency in enforcing many rules of accessibility, clearance, revision, and information checking. It finds extensive utility in tasks such as code validation, clash detection, QTO, quality assurance checks, and other model compliance checks against national and international standards.

In the case study context, SMC facilitated essential intersection geometrical and alphanumeric information checks. Notably, SMC exhibited exceptional functionality in conducting geometrical checks, leveraging predefined rules for BIM coordination that delineate intersections within disciplines (e.g., Arc vs. Arc) and intersections spanning different disciplines (e.g., Arc vs. Structure). The software also offers the flexibility of customising rules, thus enabling the implementation of a project-specific clash matrix.

The alphanumerical information checks showed more powerful results, as many rules of property existence, property set, data type, and value can be defined in the same check with the feature to get where the problem existed. Through the same example used in the Revit interoperability tool review, explained in Pg. 80, the rule is defined as one check, and the result will show if the problem is the absence of the property, in the wrong property or has incorrect value and so on. To be mentioned, SMC does not support regexes fully (only `*` and `?` are allowed). The rule SOL/21 has better support but not the same advantages of using regexes.

To achieve interoperability with other software, SMC adopts the principles of openBIM. It seamlessly imports building models from various BIM software products using the IFC interface. Furthermore, SMC enables the efficient exchange of results, problems, and issues by offering export options in various formats, including PDF, Excel, and RTF, or by utilising BCF files, which can be seamlessly transferred to the central repository. Consequently, SMC's workflow is not constrained to a specific repository, enabling harmonious interoperability among diverse software applications. This flexibility empowers stakeholders to collaborate effectively across different platforms, fostering a more efficient and streamlined exchange information process.

6.3.3 Bexel Manager Software Review

Bexel Manager Property Check Add-In was employed to verify alphanumeric information within the project. This tool operates on an Excel template that comprehensively outlines information for each project element, including the properties, properties set, data type and value validation using regexes. The tool's versatility lies in its ability to define information for each project phase. Moreover, it offers the flexibility to customise checks specifically tailored to the requirements of each project phase. This adaptability ensures that the tool remains effective and adaptable throughout the project's lifecycle.

In addition to its property check functionality, Bexel Manager mainly serves various purposes, such as clash detection and information take-off. Its distinctive feature lies in its seamless integration of cost and time, enabling the generation of analytical reports to assess the project's status comprehensively.

6.3.4 Blender Software Review

Blender software is an open-source software with a free license, making it accessible for various purposes, including educational and commercial use. In the case study context, the BIM Add-In within Blender is used for the BMC stage of rules execution within Workflow C, which is utilised to validate IDS against IFC. While the BIM Add-In within Blender offers tools for loading and creating BCF files, the integration between the IDS validator and the BCF creator is currently lacking.

Consequently, the reporting output is generated as an HTML report. The HTML report might be used with Python scripts to facilitate the navigation of reported problems and issues within the authoring modelling tool.

Blender exhibits significant potential due to its diverse free tools, encompassing tasks such as IFC creation, clash detection, and BCF generation. Its strength lies in its status as open-source design software, providing users with considerable flexibility and the opportunity to address the challenges effectively.

6.3.5 ACCA Tools Review

ACCA software offers a range of solutions for different BIM uses. In the case study, the usBIM.platform served as the Common Data Environment (CDE), containing IFC and IDS files. Additionally, the study employed usBIM.IDS Editor and Validator tools for editing IDS files and conducting validation checks against the corresponding IFC files.

The usBIM.IDS Editor tool boasts a user-friendly interface, simplifying the creation of rules in a human-readable format. This functionality enhances the flexibility to incorporate more applicability and requirements constraints for each check. Furthermore, users have the option to export the human-readable version in addition to the XML format of the IDS.

The usBIM.IDS Validator is responsible for validating the IDS file against the IFC file, identifying and reporting any problems or issues, and presenting them directly within the BIM model. Notably, the tools offer the valuable functionality of enabling users to rectify the issues within the IFC model in real-time using usBIM tools or to export a PDF or Word report for further analysis. The research recommends the investigation of using usBIM.bcf to enhance interoperability with the modelling authoring platform.

6.3.6 Software General Comparison

As a general observation, each software has its strengths and weaknesses according to the usage of the software. On the project and organisational levels, the criteria for choosing the designated software to implement the checking process are not limited to the software's capability only. The criteria of chosen software could include the cost, the functionality, the team's capability to use the software, and interoperability with other project stakeholders; as concluded in Table 12, a brief comparison is conducted to the Software used to implement the BMC process in term of its other functionalities, cost, scalability, stability and integration with the main repository (in case of the case study project is Autodesk Revit).

Table 12: Used software review

		Inside Autodesk Revit		**Solibri Model Checker (SMC)**	**Bexel Manager**	**Blender**	**ACCA**
		Autodesk Interoperability Tools	**Dynamo**				**USBIM IDS Editor and Validator**
Functionality	BMC -Information Consistency	√	√	√	√	√	√
	BMC- Code Validation	Limited	√	√	-	Limited	-
	BMC - Design Checks	Limited	√	√	-	√	-
	BMC - Rule Customisation	Limited	√	Using rule templates	√	-	√
	BMC -Data templates	-	√	√	-	-	√
	QTO	-	-	√	√	-	-
	Cost Estimation	-	-	-	√	-	-
	Clash Detection	-	-	√	√	-	-
	Schedule management	-	-	-	√	-	-
	Facility management	-	-	-	√	-	-
Friendly user Interface		√	√	√	√	√	√
Cost	Type of License	Free (shall have valid Revit License)		Annually	Annually	Free	Free
	Cost of license per user			€ 1,870	€ 2,400		
Scalability		Scalable	Scalable	Scalable	Scalable	Non-Scalable	Non-Scalable
Vendor Reputation		Promising	Well-known	Well-known	Well-known	Promising	Promising
Feedback and reviews		Emails (Responsive)	Forums	SMC Forum	Emails	Open Forums	Official Website

6.4 BMC Workflows Integration

Each BMC workflow was executed independently of other Workflows throughout the study to identify their limitations. However, it is essential to note that integrating these workflows could significantly enhance the overall capabilities of the BMC processes. As previously elucidated, all the workflows share similar procedural steps of Rule Interpretation, Model Preparation, Rule execution and Reporting. Consequently, there exists a notable opportunity to leverage each workflow for more tailored rule applications.

To illustrate this point, Workflow A exhibits strength in conducting general quality assessments, enforcing naming conventions, and fulfilling basic information requirements, which is beneficial for the design team during the early project phases. For more intricate geometrical checks requiring specialised software and tools, Workflow B is equipped to address these needs, offering the added benefit of double-checking for each submission. Additionally, in scenarios involving the definition of general information requirements spanning various file repositories such as Revit and Tekla, adopting openBIM formats for information exchange is recommended. Using Workflows B and C to meet these demands effectively.

7 CONCLUSION

The research addressed the critical necessity of defining information requirements within the BIM information exchange process context. Employing the Level of Information Need framework has significant potential for standardising the information management process. In such a scenario, the BMC adapted process with the proposed workflow, designed to verify the defined information requirements, can automate the entire process.

7.1 Theoretical and Practical Contributions

The research delved into prior theoretical and practical endeavours related to BMC processes. Various BIM uses were explored, encompassing code validation, QTO, cost estimation, and clash detection, detection as the common BIM uses where BMC could be used. Irrespective of the specific BIM use, the research emphasised that information requirements shall be derived and defined, whether they pertain to geometric or alphanumeric information. Moreover, it emphasised the importance of considering the relationships between BIM elements and other elements. For example, element clearance cannot be solely determined by the geometry of an element; it must also take into account its interaction with surrounding elements.

Subsequently, the research introduced an adaptation to the BMC process by incorporating an additional stage into the conventional steps. The proposed BMC process encompasses five distinct stages: (1) Rule Interpretation, (2) Model Preparation, (3) Rule Execution, (4) Reporting Check Results, and (5) Automated/Semi-automated codes execution to modify the information inconsistency or add the missing information.

In line with the research's aim to investigate black-box and white-box BCM approaches, three distinct BMC workflows were introduced to investigate various workflow possibilities and their practical implementation. The workflows are (1) Workflow A – Inside Reprosirty, (2) Workflow B – Standalone Software, and (3) Workflow C – openBIM. The study primarily focused on three distinct information scenarios: (1) the existence of a property along with its value, (2) property duplication, and (3) Element geometry duplication.

Following the methodology outlined, this research adopted a practical approach to investigate the proposed information requirements and BMC workflows in an actual case study. The chosen case study involved the development of an administrative building, a project undertaken by Protim Ržišnik Perc company. The Level of Information Need was aligned with the predefined checking rules for inputting the BMC process. Subsequently, each workflow was thoroughly implemented, allowing the exploration of various software and tools.

7.2 Key Findings

The Level of Information Need framework is a notable evolution, surpassing the earlier LOD framework. It introduces a higher level of precision in information management by tailoring information requirements to the unique needs of each project. Unlike LOD, which primarily focuses on data maturity and detail levels at various project stages, the Level of Information Need framework brings a nuanced approach. It takes into account factors such as project stage, stakeholder demands, and the specific purpose of the information exchange derived from the EIR. This precision ensures that information is not a one-size-fits-all commodity but rather a finely tuned resource designed to meet the distinct requirements of each project component.

Another important aspect as a practice note is the ability of regular expressions (regexes) to implement various information checks and rules. regexes are powerful and flexible tools that can be used to ensure information consistency in multiple conditions. By setting up regex-based rules, it is possible to detect and correct inconsistencies in the information, thus improving the overall quality of the project, specifically in the alphanumerical identifiers naming convention.

However, it is essential to note that conducting the checking process solely inside the BIM repository has limitations. In many cases, it may be necessary to implement other software as part of the proposed Workflow B or C to achieve the desired results. The geometrical checks that can be done inside the authoring tool are limited. For instance, Autodesk Revit's capabilities to make geometrical redundant and clash are limited, while using other standalone software will add more value, such as Navisworks or SMC. It is also essential to consider the different capabilities of each software and choose the most appropriate one for the specific project, considering the other aspects of cost, team capability to use the software and other software features that can be used for other BIM uses.

Finally, using Information Delivery Specifications (IDS) can significantly improve the exchange of information processes between different software and systems. IDS can help ensure that the information is consistent and compatible across other platforms, thus reducing the risk of errors and inconsistencies. Except for the geometrical checks, there is great potential in using the IDS in the different BMC processes of code validation, quality checks, and other specific information required according to the EIR.

7.3 Recommendations for Future Research

Over the past decades, it has become increasingly clear that the topic of BIM-based Model Checking (BMC), especially information requirement checks, is dynamic and continues to garner significant research interest theoretically and practically. Numerous challenges lie ahead, ensuring that researchers will remain deeply engaged in such topics to deal with the challenges of AECO digitisation. The study recommends some points in the same area of the research as follows:

- Geometrical representation adaptation as part of Level of Information Need implementation.
- Automation of BMC step of manipulation resulted information inconsistency and missing information.
- Developing other BMC possible workflows based on the synergy of the proposed workflows.
- Enhancing usage of BMC with different BIM uses.
- Implementation of IDS in the whole project lifecycle based on each project phase's Level of Information Need.

Moreover, the study recommends investigating other topics that may be aligned with the abovementioned topics, enhancing the information requirements process and BIM-based Model Checking. These topics are:

- Automation of international and national standards implementation using Data templates and database systems.
- Developing workflows to enhance openBIM processes that rely on industry standards such as IFC, bSDD, and BCF for all needed BIM uses.
- Analytical research on the current limitations and challenges of available BIM software in the market.

7.4 Final Thoughts

Throughout the research, it became evident that BMC occupies a central role within the broader BIM framework. BCM transcends the mere act of verifying compliance with established rules; rather, it represents a comprehensive methodology that necessitates a holistic approach from all stakeholders involved, including BIM managers, coordinators, and modellers. This methodology revolves around the meticulous allocation of information within the appropriate structure. The success of the BMC process is contingent upon the active engagement of all project parties, including high-level organisational participants, at every stage and throughout the project lifecycle. This engagement encompasses the formulation of rules and their execution and extends to the subsequent stages of reporting and refining BIM models.

8 REFERENCES

Amor, R., & Dimyadi, J. (2021). The promise of automated compliance checking. *Developments in the Built Environment*, *5*. https://doi.org/10.1016/j.dibe.2020.100039

Aydın, M. (2022). A Review of BIM-Based Automated Code Compliance Checking: A Meta-Analysis Research. In *Automation and Control - Theories and Applications*. IntechOpen. https://doi.org/10.5772/intechopen.101690

Bohrer, T. B. (2021). Quality Assurance of BIM Models in Project Management. Master's thesis, University of Ljubljana, Slovenia. https://bimaplus.org/wp-content/uploads/2021/10/2021-TaianeBohrer-Dissertation.pdf (This reference was checked as a similar topic)

Beach, T. H., Hippolyte, J. L., & Rezgui, Y. (2020). Towards the adoption of automated regulatory compliance checking in the built environment. *Automation in Construction*, *118*. https://doi.org/10.1016/j.autcon.2020.103285

Beach, T. H., Rezgui, Y., Li, H., & Kasim, T. (2015). A rule-based semantic approach for automated regulatory compliance in the construction sector. *Expert Systems with Applications*, *42*(12), 5219–5231. https://doi.org/10.1016/j.eswa.2015.02.029

BSI Standards Publication Building Information Modelling-Level of Information Need. (2020).

buildingSMART. (n.d.-a). *Website of buildingSMART International*. Industry Foundation Classes (IFC). Retrieved June 3, 2023, from https://technical.buildingsmart.org/standards/ifc/

buildingSMART. (n.d.-b). *Website of buildingSMART International*. BIM Collaboration Format (BCF). Retrieved June 1, 2023, from https://technical.buildingsmart.org/standards/bcf/

buildingSMART. (n.d.-c). *Website of buildingSMART International*. Information Delivery Manual (IDM). Retrieved May 21, 2023, from https://technical.buildingsmart.org/standards/information-delivery-manual/

buildingSMART. (n.d.-d). *Website of buildingSMART International*. Information Delivery Specification (IDS). Retrieved June 12, 2023, from https://technical.buildingsmart.org/projects/information-delivery-specification-ids/

Chapman, C., & Stolee, K. T. (2016). Exploring regular expression usage and context in python. *ISSTA 2016 - Proceedings of the 25th International Symposium on Software Testing and Analysis*, 282–293. https://doi.org/10.1145/2931037.2931073

Dimyadi, J., Solihin, W., Eastman, C., & Amor, R. (2016). *Integrating the BIM Rule Language into Compliant Design Audit Processes*.

Ding, L., Drogemuller, R., Marchant, D., Robin Drogemuller, csiroau, Rosenman, M., & Gero, J. (2006). Automating code checking for building designs-DesignCheck. *Clients Driving Innovation: Moving Ideas into Practice*, 1–16.

Eastman, C., Lee, J., Jeong, Y., & Lee, J. (2009). Automatic rule-based checking of building designs. *Automation in Construction*, *18*(8), 1011–1033. https://doi.org/10.1016/j.autcon.2009.07.002

Fazeli, A., Dashti, M. S., Jalaei, F., & Khanzadi, M. (2021). An integrated BIM-based approach for cost estimation in construction projects. *Engineering, Construction and Architectural Management*, *28*(9), 2828–2854. https://doi.org/10.1108/ECAM-01-2020-0027

Fenves, S. J., Garrett, J. H., Kiliccote, H., Law, K. H., & Reed2, K. A. (1995). COMPUTER REPRESENTATIONS OF DESIGN STANDARDS AND BUILDING CODES: U.S. PERSPECTIVE. *The International Journal of Construction Information Technology*, *3*(1), 13–34.

Guedes, Í., & Andrade, M. (2019). *Automatic Rule-Based Checking for the Approval of Building Architectural Designs of Airport Passenger Terminals based on BIM*. 333–338. https://doi.org/10.52842/conf.ecaade.2019.2.333

Hjelseth, E. (2009). Foundation for development of computable rules. *Managing IT in Construction/Managing Construction for Tomorrow*, 257–266.

Hjelseth, E. (2015). *Foundations for BIM-based model checking systems Transforming regulations into computable rules in BIM-based model checking systems Prinsipper for BIM basert modell sjekking Omforming av bestemmelser i regelverket til digitaliserte regler i BIM-baserte systemer for modellsjekking*. *2015*, 54.

Hjelseth, E. (2016). Classification of BIM-based model checking concepts. *ITcon*, *21*(CIB W78 2015 Special track on Compliance Checking), 354–369. http://www.itcon.org/2016/23

Hjelseth, E., & Nisbet, N. N. (2011). Capturing normative constraints by use of the semantic mark-up RASE methodology. *CIBW78 2011 28th International Conference-Applications of IT in the AEC Industry*.

Ismail, A. S., Ali, K. N., Iahad, N. A., Kassem, M. A., & Al-Ashwal, N. T. (2023). BIM-Based Automated Code Compliance Checking System in Malaysian Fire Safety Regulations: A User-Friendly Approach. *Buildings*, *13*(6), 1404. https://doi.org/10.3390/buildings13061404

[ISO 19650-1]. (2018). *Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) - Information management using building information modelling - Part 1: Concepts and principles* (Vol. 1). CEN-CENELEC Management Centre.

[ISO 19650-2]. (2018). *Organization and digitization of information about buildings and civil engineering works, including building information modelling (BIM) - Information management using building information modelling - Part 2: Delivery phase of the assets*.

Jadid, M. N., & Idrees, M. M. (2007). Cost estimation of structural skeleton using an interactive automation algorithm: A conceptual approach. *Automation in Construction*, *16*(6), 797–805. https://doi.org/10.1016/j.autcon.2007.02.007

Jovanovic, L., Meng, L., Doherty, B., Gardner, N., Hank Haeusler, M., & Daniel Yu, K. (2022). Automated Code Compliance Checking A computational workflow for verifying model, parameter and regulatory compliance. *ECAADe 40*, *2*, 319–832.

Jubierre, J. R. (2015). Knowledge-Based Engineering for Infrastructure Facilities: Assisted Design of Railway Tunnels Based on Logic Models and Advanced Procedural Geometry Dependencies. *Www.Itcon.Org-Journal of Information Technology in Construction*, *20*. www.itcon.org

Laakso, M., & Kiviniemi, A. (2012). The IFC Standard-A Review of History, Development, and Standardization Platform business model for construction industry View project Requirements management interface to building product models View project. In *Article in Electronic Journal of Information Technology in Construction*. www.itcon.org

Liu, H., Cheng, J. C. P., Gan, V. J. L., & Zhou, S. (2022). A knowledge model-based BIM framework for automatic code-compliant quantity take-off. *Automation in Construction*, *133*. https://doi.org/10.1016/j.autcon.2021.104024

Lopez, R., & Love, P. E. D. (2012). Design Error Costs in Construction Projects. *Journal of Construction Engineering and Management*, *138*(5), 585–593. https://doi.org/10.1061/(ASCE)CO.1943-7862.0000454

Ma, Z., Wei, Z., & Zhang, X. (2013). Semi-automatic and specification-compliant cost estimation for tendering of building projects based on IFC data of design model. *Automation in Construction*, *30*, 126–135. https://doi.org/10.1016/j.autcon.2012.11.020

Pauwels, P., Van Deursen, D., Verstraeten, R., De Roo, J., De Meyer, R., Van De Walle, R., & Van Campenhout, J. (2011). A semantic rule checking environment for building performance checking. *Automation in Construction*, *20*(5), 506–518. https://doi.org/10.1016/j.autcon.2010.11.017

Preidel, C., & Borrmann, A. (2015, June 18). *Automated Code Compliance Checking Based on a Visual Language and Building Information Modeling*. https://doi.org/10.22260/ISARC2015/0033

Preidel, C., & Borrmann, A. (2016). Towards code compliance checking on the basis of a visual programming language. *Journal of Information Technology in Construction (ITcon)*, *21*(CIB W78 2015 Special track on Compliance Checking), 402–421. http://www.itcon.org/2016/25

Preidel, C., Bus, N., Borrmann, A., & Fies, B. (2018). Pre-Processing IFC Building Models for Code Compliance Checking based on Visual Programming. *ICCCBE* .

Regular Expression Language - Quick Reference. (n.d.). Retrieved June 23, 2023, from https://learn.microsoft.com/en-us/dotnet/standard/base-types/regular-expression-language-quick-reference

Schwabe, K., König, M., & Teizer, J. (2016, July 21). *BIM Applications of Rule-Based Checking in Construction Site Layout Planning Tasks*. https://doi.org/10.22260/ISARC2016/0026

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

Solihin, W., Dimyadi, J., Lee, Y.-C., Eastman, C., & Amor, R. (2017). The Critical Role of Accessible Data for BIM-Based Automated Rule Checking Systems. *Lean and Computing in Construction Congress - Volume 1: Proceedings of the Joint Conference on Computing in Construction*, 53–60. https://doi.org/10.24928/JC3-2017/0161

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

Succar, B. (2019). *BIMe Initiative Model Uses List*. https://doi.org/https://doi.org/10.5281/zenodo.3563403

UK BIM Framework. (2020). *Information management according to BS EN ISO 19650*. http://www.bsigroup.com/Shop

[UNI 11337-5]. (2017). *Building and civil engineering works - Digital management of the informative processes - Part 5: Informative flows in the digital processes*.

Website of the Legal Information System of Slovenia. (n.d.). Rules on Minimum Technical Requirements for the Construction of Residential Buildings and Apartments. Retrieved July 22, 2023, from Rules on minimum technical requirements for the construction of residential buildings and apartments

Zhang, S., Teizer, J., Lee, J. K., Eastman, C. M., & Venugopal, M. (2013). Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules. *Automation in Construction*, *29*, 183–195. https://doi.org/10.1016/j.autcon.2012.05.006

9 APPENDICES

Appendix A: Level of Information Need for BIM uses of QTO & cost estimation

This appendix defines Level of Information Need for architectural and structural elements for the purpose of QTO and cost estimation.

Project:				
Purpose: QTO & Cost Estimation				
Actor: Lead Appointed Party- Design Consultant				
1	**Ceiling**			
Model Element		**Ceiling**		
Category		Ceilings		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	2D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Interior (RP-AR-Facade only for elements that are part of the facade)
		Dimension	Material	Shall have material, follow local material list
		Dimension	Area	Parameter Area shall be IsReadOnly: True
		Dimension	Area	Units shall be m2
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
2	**Curtain Walls**			
Model Element		**Curtain Walls**		
Category		Walls		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	2D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Structural	Structural	Shall have Value (Boolean Yes/No)
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Facade/ RP-AR-Interior

		Materials and Finishes	Material	Shall have material, follow local material list
		Dimension	Area	Parameter Area shall be IsReadOnly: True
		Dimension	Area	Units shall be m2
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
3		**Curtain Wall Panels**		
	Model Element	**Curtain Wall Panels**		
	Category	Curtain Panels		
	Information delivery milestone	PZI		
	Family Type	System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		
	Location	Relative		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Interior
		Dimension	Material	Shall have material, follow local material list
		Dimension	Area	Parameter Area shall be IsReadOnly: True
		Dimension	Area	Units shall be m2
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
4		**Curtain Wall Mullions**		
	Model Element	**Curtain Wall Mullions**		
	Category	Curtain Wall Mullions		
	Information delivery milestone	PZI		
	Family Type	System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		
	Location	Relative		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Facade
		Materials and Finishes	Material	Shall have material, follow local material list
		Dimension	Area	Parameter Area shall be IsReadOnly: True
		Dimension	Area	Units shall be m2
		Dimension	Length	Units shall be m
		Dimension	Length	Parameter Area shall be IsReadOnly: True
		WBS	WBSCode	Shall have a WBS Property with Value:

Documentation	**Set of documents**	N/A		
5		**Doors**		
Model Element		**Doors**		
Category		Doors		
Information delivery milestone		PZI		
Family Type		Loadable Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	2D		
	Location	Relative		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Family Name		
		Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Facade
		Identity Data	Mark	Each element shall have a unique Mark
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
6		**Architectural Floor**		
Model Element		**Architectural Floor**		
Category		Floors		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	2D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Structural	Structural	Shall have Value (Boolean: No)
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Interior
		Dimension	Area	Parameter Area shall be IsReadOnly: True
		Dimension	Area	Units shall be m2
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
7		**Furniture**		
Model Element		**Furniture**		
Category		Furniture		
Information delivery milestone		PZI		
Family Type		Loadable Family		
Ge om	**Detail (Using LOD)**	N/A		

	Dimensionality	3D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	Identification	Family Name		
		Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-ST-Concrete_structures
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	Set of documents	N/A		
8		**Equipment**		
Model Element		**Equipment**		
Category		Specially Equipment		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	Detail (Using LOD)	N/A		
	Dimensionality	3D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	Identification	Family Name		
		Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Equipment
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	Set of documents	N/A		
9		**Plumbing Fixtures**		
Model Element		**Plumbing Fixtures**		
Category		Plumbing Fixtures		
Information delivery milestone		PZI		
Family Type		Loadable Family		
Geometrical Representation	Detail (Using LOD)	N/A		
	Dimensionality	3D		
	Location	Absolute/Relative		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	Identification	Family Name		
		Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Equipment
		WBS	WBSCode	Shall have a WBS Property with Value:

Documentation	**Set of documents**	N/A		
10		**Railings, guardrails and handrails**		
Model Element		**Railings, guardrails and handrails**		
Category		Railings		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	1D		
	Location	Absolute/Relative		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Facade
		Dimension	Length	Parameter Area shall be IsReadOnly: True
		Dimension	Length	Units shall be m
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
11		**Roofs**		
Model Element		**Roofs**		
Category		Roofs		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	2D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Interior
		Dimension	Area	Parameter Area shall be IsReadOnly: True
		Dimension	Area	Units shall be m2
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
12		**Finishing Stairs**		
Model Element		**Finishing Stairs**		
Category		Stairs		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		

	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Interior
		Dimension	Volume	Parameter Volume shall be IsReadOnly: True
		Dimension	Volume	Units shall be m3
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
13		**Finishing Stairs Runs**		
Model Element		**Finishing Stairs Runs**		
Category		Stairs: Runs		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		
	Location	Relative		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Interior
		Dimension	Volume	Parameter Volume shall be IsReadOnly: True
		Dimension	Volume	Units shall be m3
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
14		**Finishing Stairs Landings**		
Model Element		**Finishing Stairs Landings**		
Category		Stairs: Landings		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		
	Location	Relative		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Interior
		Dimension	Volume	Parameter Volume shall be IsReadOnly: True

		Dimension	Volume	Units shall be m3
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
15	**Interior Wall (Partition Interior Wall / Finishing Wall/..)**			
Model Element		**Interior Wall (Partition Interior Wall / Finishing Wall/..)**		
Category		Walls		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Family Name		
		Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Structural	Structural	Shall have Value (Boolean No)
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-Interior
		Dimension	Area	Parameter Volume shall be IsReadOnly: True
		Dimension	Area	Units shall be m2
		Dimension	Volume	Parameter Volume shall be IsReadOnly: True
		Dimension	Volume	Units shall be m3
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
16	**Exterior Wall**			
Model Element		**Exterior Wall**		
Category		Walls		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Family Name		
		Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Structural	Structural	Shall have Value (Boolean No)
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Facade
		Dimension	Area	Parameter Volume shall be IsReadOnly: True
		Dimension	Area	Units shall be m2
		Dimension	Volume	Parameter Volume shall be IsReadOnly: True
		Dimension	Volume	Units shall be m3

		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
17		**Windows**		
Model Element		**Windows**		
Category		Windows		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Family Name		
		Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-AR-Interior or/ PR-AR-Facade
		Identity Data	Mark	Each element shall have a unique Mark
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
18		**Structural Columns**		
Model Element		**Structural Columns**		
Category		Structural Columns		
Information delivery milestone		PZI		
Family Type		Loadable Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Family Name		
		Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-ST-Concrete_structures
		Dimension	Area	Parameter Volume shall be IsReadOnly: True
		Dimension	Area	Units shall be m2
		Dimension	Volume	Parameter Volume shall be IsReadOnly: True
		Dimension	Volume	Units shall be m3
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
19		**Structural Framing**		
Model Element		**Structural Framing**		
Category		Structural Framing		

Information delivery milestone		PZI		
Family Type		Loadable Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Family Name		
		Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-ST-Concrete_structures
		Dimension	Area	Parameter Volume shall be IsReadOnly: True
		Dimension	Area	Units shall be m2
		Dimension	Volume	Parameter Volume shall be IsReadOnly: True
		Dimension	Volume	Units shall be m3
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
20		**Slabs**		
Model Element		**Slabs**		
Category		Floors		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	2D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Structural	Structural	Shall have Value (Boolean: Yes)
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-ST-Concrete_structures
		Dimension	Area	Parameter Area shall be IsReadOnly: True
		Dimension	Area	Units shall be m2
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
21		**Structural Foundation**		
Model Element		**Structural Foundation**		
Category		Structural Foundation		
Information delivery milestone		PZI		
Family Type		Loadable Family		
Geometrical	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		
	Location	Absolute		

<table>
<tr><td></td><td>Appearance</td><td colspan="3">No Colour</td></tr>
<tr><td></td><td>Parametric behaviour</td><td colspan="3">Explicit geometry</td></tr>
<tr><td rowspan="10">Alphanumerical Information</td><td rowspan="2">Identification</td><td colspan="3">Family Name</td></tr>
<tr><td colspan="3">Type Name</td></tr>
<tr><td rowspan="8">Information content</td><td>Property Set</td><td>Property</td><td>Requirement / Value</td></tr>
<tr><td>Identity Data</td><td>Type Mark</td><td>Shall have a Type Mark with the regular expression defined in the naming convention</td></tr>
<tr><td>Identity Data</td><td>Workset</td><td>RP-ST-Concrete_structures</td></tr>
<tr><td>Dimension</td><td>Area</td><td>Parameter Volume shall be IsReadOnly: True</td></tr>
<tr><td>Dimension</td><td>Area</td><td>Units shall be m2</td></tr>
<tr><td>Dimension</td><td>Volume</td><td>Parameter Volume shall be IsReadOnly: True</td></tr>
<tr><td>Dimension</td><td>Volume</td><td>Units shall be m3</td></tr>
<tr><td>WBS</td><td>WBSCode</td><td>Shall have a WBS Property with Value:</td></tr>
<tr><td>Documentation</td><td>Set of documents</td><td colspan="3">N/A</td></tr>
<tr><td>22</td><td colspan="4">Shear/ Retaining Walls</td></tr>
<tr><td colspan="2">Model Element</td><td colspan="3">Shear/ Retaining Walls</td></tr>
<tr><td colspan="2">Category</td><td colspan="3">Walls</td></tr>
<tr><td colspan="2">Information delivery milestone</td><td colspan="3">PZI</td></tr>
<tr><td colspan="2">Family Type</td><td colspan="3">System Family</td></tr>
<tr><td rowspan="5">Geometrical Representation</td><td>Detail (Using LOD)</td><td colspan="3">N/A</td></tr>
<tr><td>Dimensionality</td><td colspan="3">3D</td></tr>
<tr><td>Location</td><td colspan="3">Absolute</td></tr>
<tr><td>Appearance</td><td colspan="3">No Colour</td></tr>
<tr><td>Parametric behaviour</td><td colspan="3">Explicit geometry</td></tr>
<tr><td rowspan="12">Alphanumerical Information</td><td rowspan="2">Identification</td><td colspan="3">Family Name</td></tr>
<tr><td colspan="3">Type Name</td></tr>
<tr><td rowspan="10">Information content</td><td>Property Set</td><td>Property</td><td>Requirement / Value</td></tr>
<tr><td>Structural</td><td>Structural</td><td>Shall have Value (Boolean Yes)</td></tr>
<tr><td>Identity Data</td><td>Type Mark</td><td>Shall have a Type Mark with the regular expression defined in the naming convention</td></tr>
<tr><td>Identity Data</td><td>Workset</td><td>RP-ST-Concrete_structures</td></tr>
<tr><td>Structural</td><td>Structural Usage</td><td>shear</td></tr>
<tr><td>Dimension</td><td>Area</td><td>Parameter Volume shall be IsReadOnly: True</td></tr>
<tr><td>Dimension</td><td>Area</td><td>Units shall be m2</td></tr>
<tr><td>Dimension</td><td>Volume</td><td>Parameter Volume shall be IsReadOnly: True</td></tr>
<tr><td>Dimension</td><td>Volume</td><td>Units shall be m3</td></tr>
<tr><td>WBS</td><td>WBSCode</td><td>Shall have a WBS Property with Value:</td></tr>
<tr><td>Documentation</td><td>Set of documents</td><td colspan="3">N/A</td></tr>
<tr><td>23</td><td colspan="4">Bearing Walls</td></tr>
<tr><td colspan="2">Model Element</td><td colspan="3">Bearing Walls</td></tr>
<tr><td colspan="2">Category</td><td colspan="3">Walls</td></tr>
<tr><td colspan="2">Information delivery milestone</td><td colspan="3">PZI</td></tr>
<tr><td colspan="2">Family Type</td><td colspan="3">System Family</td></tr>
<tr><td rowspan="3">Geometrical</td><td>Detail (Using LOD)</td><td colspan="3">N/A</td></tr>
<tr><td>Dimensionality</td><td colspan="3">3D</td></tr>
<tr><td>Location</td><td colspan="3">Absolute</td></tr>
</table>

	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Family Name		
		Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Structural	Structural	Shall have Value (Boolean Yes)
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-ST-Concrete_structures
		Structural	Structural Usage	Bearing
		Dimension	Area	Parameter Volume shall be IsReadOnly: True
		Dimension	Area	Units shall be m2
		Dimension	Volume	Parameter Volume shall be IsReadOnly: True
		Dimension	Volume	Units shall be m3
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
24		**Structural Stairs**		
Model Element		**Structural Stairs**		
Category		Stairs		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		
	Location	Absolute		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	**Identification**	Type Name		
	Information content	**Property Set**	**Property**	**Requirement / Value**
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-ST-Concrete_structures
		Dimension	Volume	Parameter Volume shall be IsReadOnly: True
		Dimension	Volume	Units shall be m3
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	**Set of documents**	N/A		
25		**Structural Stairs Runs**		
Model Element		**Structural Stairs Runs**		
Category		Stairs: Runs		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	**Detail (Using LOD)**	N/A		
	Dimensionality	3D		
	Location	Relative		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alpha	**Identification**	Type Name		

	Information content	Property Set	Property	Requirement / Value
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-ST-Concrete_structures
		Dimension	Volume	Parameter Volume shall be IsReadOnly: True
		Dimension	Volume	Units shall be m3
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	Set of documents	N/A		
26	Structural Stairs Landings			
Model Element		Structural Stairs Landings		
Category		Stairs: Landings		
Information delivery milestone		PZI		
Family Type		System Family		
Geometrical Representation	Detail (Using LOD)	N/A		
	Dimensionality	3D		
	Location	Relative		
	Appearance	No Colour		
	Parametric behaviour	Explicit geometry		
Alphanumerical Information	Identification	Type Name		
	Information content	Property Set	Property	Requirement / Value
		Identity Data	Type Mark	Shall have a Type Mark with the regular expression defined in the naming convention
		Identity Data	Workset	RP-ST-Concrete_structures
		Dimension	Volume	Parameter Volume shall be IsReadOnly: True
		Dimension	Volume	Units shall be m3
		WBS	WBSCode	Shall have a WBS Property with Value:
Documentation	Set of documents	N/A		

Appendix B: Protim Ržišnik Perc BIM model rules and metrics

This appendix encompasses the existing and new textual general, geometrical and alphanumerical information for Protim Ržišnik Perc company projects.

Checking	Category	Concept	Rule	Source
General Quality Rules	File	Size	The Revit file is smaller than 400 MB	exist
	File	Warnings	The Revit file must have no warnings/ as few as possible	new
	File	Purgeable Elements	The Revit file must have unneeded elements that increase the model size	new
	File	Non-built-in Objects	A large number of Non-built-in Objects may be indicative of an imported CAD file. Importing CAD files is not recommended unless it is part of the Drafting view	new
	File	Model Groups	Usage of groups should be avoided; create them only if necessary.	exist
	File	Detail Groups	Details groups should be created only if necessary.	new
	File	In-Place Families	The use of Models In-Place Families should be avoided. Exception: Use them if you do not need to calculate their area and volume or tag them. If you need to model such a family, write in the name prefix “IPF-”.	exist
	External Files	Revit Links	The Revit Links are correctly organised, and the insertion method is correct. Links shall be pinned in place	exist
	External Files	CAD Links	Importing CADs into model views should be avoided. Exception: You can link CAD drawings into the view with the option “visible on current view only”.	exist
	External Files	CAD Links	Import CAD is allowed to draft views (reusing details from AutoCAD) or as a 2D family only.	exist
	File	source of coordinates, grids, and levels	The main source of coordinates, grids and levels is the Coordination file; all discipline models obtain coordinates, grids, and levels according to it. The coordination file is stored in the 99_BIM folder.	exist
	Base Point	Location	The project base point shall be located at the intersection of grids one and A.	exist
	Levels	Naming	The Levels' names according to RP Revit Standard Levels & Grids Chapter	exist
	Grids	Grids Naming	Vertical grids are marked with consecutive numbers (1, 2, 3…), and horizontal grids are marked with consecutive capital letters (A, B, C, D…). The grids’ starting point is on the bottom left of the building.	exist
	Rooms	Unplaced Rooms	There are NOT unplaced, unclosed, and redundant rooms	exist
	Rooms	Rooms	Rooms are represented correctly and are tagged in drawings. No text allowed	exist
	Rooms	Rooms	Every Room Number is unique, and they have the right Room Name	exist
	Rooms	Rooms	Every Room is set with the proper boundary height	exist
	Views	Views Template	Each view should be assigned to a view template.	new
		Navisworks Export View	There should be a 3D view labelled with the word "Navisworks" for export to Navisworks.	new

	Parameters	Project Info	Project Information filled with all the relevant and up-to-date data	exist
Geometrical Requirements	All	No duplicates	No duplicated elements	exist
	All	General Rule	Refer to Company LOD specification for each project Phase	exist
	Walls, Structural Columns, Columns	Model breakdown	The Walls and Columns are modelled according to the constructive requirement. As a rule, the elements that go from the bottom of the building to the top (walls, columns, facades) should be separated by levels. Exception: In the case of prefabricated modular elements, they should be modelled in one piece.	exist exist
	Ceilings	Model breakdown	Ceilings are modelled correctly and include a Revit pattern, not lines, aligned to texture	exist
	Curtain Walls	Curtain Walls	Curtain walls and their elements are placed correctly (doors, panels, profiles, etc.)	exist
Information Requirements	All	General Rule	Refer to the Level of Information Need Defined for each BIM use.	New
	All	Schedules	All the necessary data for the current project phase is in the Roombook schedule.	exist
	Worksets	Workset Names	Workset names according to RP Revit Standard Workset chapter	exist
	All	Materials	Material names are logical, consistent, and named according to RP Revit Standard Materials. Materials assigned to all elements starting from the IDP phase	exist
	All	Families and Types	There are no duplicate parameters or unnecessary parameters	exist
	All	Schedules	All tables are generated through Revit Schedules. Some exceptions apply.	exist
	All	Tags	All notes are inserted as Tags and not as Text. Exceptions can be considered for fabrication or installation details.	exist
	All	Schedules	All tables are generated through Revit Schedules. Some exceptions apply.	exist
		Tags	All notes are inserted as Tags and not as Text. Exceptions can be considered for fabrication or installation details.	exist

Appendix C: Regular expression definition

This appendix defines regular expressions for architectural and structural elements. Also, it includes regular expressions of the material list.

<table>
<tr><th>Element(s)</th><th>Identifier(s)</th><th>Regular Expression (s)</th><th>Example(s)</th></tr>
<tr><td>Ceiling</td><td>Type Name</td><td>RP-AR-(STCL|SSCL)-(GB|MS|WD|PS|TI|PI)(_MDF|_ACS|_AQU|_XPS|_EPS|_MNW|_WTR)??-[0-9]*mm</td><td>RP-AR-SSCL-GB-600x1200-50mm</td></tr>
<tr><td>Curtain Walls</td><td>Type Name</td><td>RP-AR-CURW(-(GL||MM|PL|AP|H|TX))??((-|_)[0-9]*x(GL||MM|PL|AP|H|TX))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PIR|_MDF|_ACS|_ACM|_BTM|_STF|_PVC|_AQU|_VIS|_WTR|_HPL)??(-|_)[0-9]*mm</td><td>RP-AR-CURW-150mm</td></tr>
<tr><td>Curtain Panels</td><td>Type Name</td><td>RP-AR-CURP-(PL|NA|WD|AL|GL|VOID)(_(PL|NA|WD|AL|GL|VOID))??(-M3-R2[0-5])??</td><td>RP-AR-CURP-AL_GL</td></tr>
<tr><td>Curtain Mullions</td><td>Type Name</td><td>RP-AR-CURM-(PL|NA|WD|AL)(-[0-9]*x[0-9]*mm)??(-[A-Za-z0-9])??</td><td>RP-AR-CURM--150x50mm</td></tr>
<tr><td rowspan="2">Doors</td><td>Family Name</td><td>RP-AR-(SGLD|DBLD|TRPD|REVD|PASS|RLRD|SNPD|DBPD|OHSD|OHRD)-(Swing|Sliding|Swing_Asim|Swing_Pivot|Sliding_Surface|Sliding_Telescopic|Sliding_Pocket|Double_Acting|Sliding_Bi_Parting|NA)(-(Emb|Cor|Cen|NA|Emb_Rvs))??(-(Full|Glass|Mesh)-(w_Toplight|w_Sidelight|w_Sidelight_Toplight|w_Vision_Panel|w_Inactive_Leaf|w_Pass_Door))??(-M3-R2[0-5])??</td><td>RP-AR-DBLD-Swing-Emb-Full-w_Inactive_Leaf-M3-R20</td></tr>
<tr><td>Type Name</td><td>(WD_GL|AL_AL|PL-PL|WD_WD|ST-ST)(-[0-9]*x[0-9]*mm)??(-[A-Za-z0-9]*)??</td><td>AL_AL-2000x2100mm-EI60</td></tr>
<tr><td>Architectural Floors</td><td>Type Name</td><td>RP-AR-(FLRS|LOAD|LNDG|ROAD|TRRN|ROOF|RFLR|DRPN|RFSL|PVNG|SLTH|FOOT|FLRM|PATH|FLRI|HIBD|CMFL|FNDB|SLED|GRNR)(-(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))??((-|_)[0-9]*x(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PRQ|_CMP)??(-|_)[0-9]*mm</td><td>RP-AR-LOAD-PQ-150mm</td></tr>
<tr><td>Structural Floors (slabs)</td><td>Type Name</td><td>RP-ST-(FLRS|LOAD|LNDG|ROAD|TRRN|ROOF|RFLR|DRPN|RFSL|PVNG|SLTH|FOOT|FLRM|PATH|FLRI|HIBD|CMFL|FNDB|SLED|GRNR)(-(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))??((-|_)[0-9]*x(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PRQ|_CMP)??(-|_)[0-9]*mm</td><td>RP-ST-FLRS-RC-200mm</td></tr>
</table>

<table>
<tr><td>Furniture</td><td>Family Name</td><td>RP-AR-Furniture-[A-za-z]*-.*</td><td>RP-AR-Furniture-Table</td></tr>
<tr><td></td><td>Type Name</td><td>[A-za-z]*??-??[0-9](x[0-9]*){0,2}mm</td><td>400*500mm</td></tr>
<tr><td>Equipment</td><td>Family Name</td><td>RP-AR-SPEQ-.*</td><td>RP-AR-SPEQ-External_Curtain</td></tr>
<tr><td></td><td>Type Name</td><td>[A-za-z]*??-??[0-9](x[0-9]*){0,2}mm</td><td>1200*1000mm</td></tr>
<tr><td>Plumbing Fixtures</td><td>Family Name</td><td>RP-AR-PLUM-*</td><td>RP-AR-PLUM-Sink</td></tr>
<tr><td></td><td>Type Name</td><td>[A-za-z]*??-??[0-9](x[0-9]*){0,2}mm</td><td>1200*1000mm</td></tr>
<tr><td>Exterior Railings, guardrails, and handrails</td><td>Type Name</td><td>RP-AR-RALG-(PL|NA|WD|AL|GL)(_(PL|NA|WD|AL|GL))??-([0-9]*x[0-9]*mm|[0-9]*mm)??(-[A-Za-z0-9])??(-M3-R2[0-5])??</td><td>RP-AR-RALG-AL_GL-40mm</td></tr>
<tr><td>Roof</td><td>Type Name</td><td>RP-AR-ROOF(-(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))??((-|_)[0-9]*x(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PRQ|_CMP)??(-|_)[0-9]*mm</td><td>RP-AR-GB-20mm</td></tr>
<tr><td>Structural Stairs</td><td>Type Name</td><td>RP-ST-STAR(-(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TIT|FO|CO))??((-|_)[0-9]*x(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PRQ|_CMP)??(-|_)[0-9]*mm</td><td>RP-ST-RC-1800x300x166mm</td></tr>
<tr><td>Structural Stairs Run</td><td>Type Name</td><td>RP-ST-RUN(-(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))??((-|_)[0-9]*x(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PRQ|_CMP)??(-|_)[0-9]*mm</td><td>RP-ST-RUN-RC-40mm</td></tr>
<tr><td>Structural Stairs Landings</td><td>Type Name</td><td>RP-ST-LNDG(-(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))??((-|_)[0-9]*x(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PRQ|_CMP)??(-|_)[0-9]x[0-9]*x[0-9]*mm</td><td>RP-ST-LNDG-RC-40mm</td></tr>
<tr><td>Finishing Stairs</td><td>Type Name</td><td>RP-AR-STAR(-(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))??((-|_)[0-9]*x(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PRQ|_CMP)??(-|_)[0-9]x[0-9]*x[0-9]*mm</td><td>RP-AR-RUN-CR-1700x2800x150mm</td></tr>
<tr><td>Finishing Stairs Run</td><td>Type Name</td><td>RP-AR-RUN(-(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))??((-|_)[0-9]*x(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PRQ|_CMP)??(-|_)[0-9]*mm</td><td>RP-AR-RUN-CR-40mm</td></tr>
<tr><td>Finishing Stairs Landings</td><td>Type Name</td><td>RP-AR-LNDG(-(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|G</td><td>RP-AR-LNDG-CR-40mm</td></tr>
</table>

		V|SC|TR|TI|CT|FO|CO))??((-|_)[0-9]*x(RC|AC|EP|CR|VI|PQ|PV|TX|AS|GR|PT|WC|GV|SC|TR|TI|CT|FO|CO))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PRQ|_CMP)??(-|_)[0-9]*mm	
Structural Columns	Family Name	RP-ST-STCL-(Rectangular|Round|IPE|HEA)(-(Forks|Drop_Panel))??(-M3-R2[0-5])??	RP-ST-STCL-Rectangular-M3-R20
	Type Name	(WD|RC|ST-HEA|ST-IEP)-[0-9]*x[0-9]*mm	RC-40x60mm
Structural Framing	Family Name	RP-ST-STFR-(Rectangular|L|I|T|Inverted_T|Aastho|IPE|HEA)(-(Tapered|Varying_SectionDapped_End))??(-FB)??(-M3-R2[0-5])??	RP-ST-STFR-Rectangular-M3-R20
	Type Name	((Beam|Tie_Beam|Bond|Console)-)??(WD|RC|ST-HEA|ST-IEP)-[0-9]*x[0-9]*((x[0-9]*mm)|mm)	Bond-RC-150x150mm
Structural Foundation	Family Name	RP-ST-(PFND|TRNS)(-(Precast_Footing|Precast_Pocket|Precast_Wall))??-(Simple|Sloped|Single|Double)(-M3-R2[0-5])??	RP-ST-PFND-Precast_Pocket-Simple-M3-R20
	Type Name (System Family)	RP-ST-FNDS-RC-([0-9]*mm|[0-9]*x[0-9]*x[0-9]*mm)	RP-ST-FNDS-RC-500mm
	Type Name (Loadable Family)	RC-([0-9]*mm|[0-9]*x[0-9]*x[0-9]*mm)	RC-2500x2500x1100mm
Shear/ Retaining Walls	Type Name	RP-ST-(WALL|DRYW|PEDW|PARW|WCOV|WFIN|WINS|CURW|STFC|WFAC|FCOV|CLNW|CLNC|CASC|LUVR)(-(RC|CB|AC|WD|SN|GB|BR|PT|GL|CR|MM|PL|TI|FO|FI|PS|AP|HI|PI|TX))??((-|_)[0-9]*x(RC|CB|AC|WD|SN|GB|BR|PT|GL|CR|MM|PL|TI|FO|FI|PS|AP|HI|PI|TX))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PIR|_MDF|_ACS|_ACM|_BTM|_STF|_PVC|_AQU|_VIS|_WTR|_HPL)??(-|_)[0-9]*mm	RP-ST-WALL-RC_PRE-200mm
Bearing walls	Type Name	RP-ST-(WALL|DRYW|PEDW|PARW|WCOV|WFIN|WINS|CURW|STFC|WFAC|FCOV|CLNW|CLNC|CASC|LUVR)(-(RC|CB|AC|WD|SN|GB|BR|PT|GL|CR|MM|PL|TI|FO|FI|PS|AP|HI|PI|TX))??((-|_)[0-9]*x(RC|CB|AC|WD|SN|GB|BR|PT|GL|CR|MM|PL|TI|FO|FI|PS|AP|HI|PI|TX))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PIR|_MDF|_ACS|_ACM|_BTM|_STF|_PVC|_AQU|_VIS|_WTR|_HPL)??(-|_)[0-9]*mm	RP-ST-PEDW-RC-250mm
Interior Wall (Partition Interior Wall / Finishing Wall/..)	Type Name	RP-AR-(WALL|DRYW|PEDW|PARW|WCOV|WFIN|WINS|STFC|WFAC|FCOV|CLNW|CLNC|CASC|LUVR)(-(RC|CB|AC|WD|SN|GB|BR|PT|GL|CR|MM|PL|TI|FO|FI|PS|AP|HI|PI|TX))??((-|_)[0-9]*x(RC|CB|AC|WD|SN|GB|BR|PT|GL|CR|M	RP-AR-PARW-20xPS_100xAC_20xPS-240mm

		M|PL|TI|FO|FI|PS|AP|HI|PI|TX))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PIR|_MDF|_ACS|_ACM|_BTM|_STF|_PVC|_AQU|_VIS|_WTR|_HPL)??(-|_)[0-9]*mm	
Exterior Walls	Type Name	RP-AR-(WALL|DRYW|PEDW|PARW|WCOV|WFIN|WINS|STFC|WFAC|FCOV|CLNW|CLNC|CASC|LUVR)(-(RC|CB|AC|WD|SN|GB|BR|PT|GL|CR|MM|PL|TI|FO|FI|PS|AP|HI|PI|TX))??((-|_)[0-9]*x(RC|CB|AC|WD|SN|GB|BR|PT|GL|CR|MM|PL|TI|FO|FI|PS|AP|HI|PI|TX))*?(_PRE|_CLT|_XPS|_EPS|_MNW|_PIR|_MDF|_ACS|_ACM|_BTM|_STF|_PVC|_AQU|_VIS|_WTR|_HPL)??(-|_)[0-9]*mm	RP-AR-WINS-MW-150mm
Windows	Family Name	RP-AR-(SGLW|SNRW|SNPW|SKLT|DBLW|DBCW|TRPW|LVRW)-(Swing|Sliding|Still|Hung|Swing_Pivot|Awning)-(Cen|NA|Cor)-??(w_Toplight|w_Sidelight|w_Undelight|w_Sidelight_Underlight)??(-M3-R2[0-5])??	RP-AR-SGLW-Swing-Cen-w_Toplight-M3-R22
	Type Name	(PL|NA|WD|AL)(-[0-9]*x[0-9]*mm)??(-[A-Za-z0-9])??	WD-900x900mm
Material List		RP-(Beton|Izolacija_Toplotna|Site|Plošča|Jeklo)(-(Armirani|XPS|Mineralna_Volna|Zazelenitev|Mavčnokartonska| Konstrukcijsko))??(-[0-9]*mm)??	