

DEVELOPMENT OF MODEL CHECKING RULES FOR VALIDATION AND CONTENT CHECKING

SEBASTIAN SEIß
Bauhaus Universität, Germany

ABSTRACT

The digitalization of the construction industry increases continuously. The focus of this digitalization is on building information modeling (BIM). Due to the growing usage of BIM, the requirements on the models are increasing, professional costumers, especially, require visualizations, cost estimation and scheduling, as well as an implementation of product data. Furthermore, costumers demand model checking, to see if a model fulfills the required guidelines, standards or product specifications. At this point in time, model checking is mostly manual and does not use the automation potentials of model checking with BIM. This paper focuses on the development of model content and validation checking. The aim is to define structures and methods to check models in the special case of professional project developers. Based on this, the development starts with an analysis of predefined building descriptions and Level of Development definitions. Afterward, an overview of the current state of the art and scientific knowledge in model checking is given. After that, the paper practices the content and validating checking based on an example model. The realization of the content checking is done by Solibri Model Checker and DesiteMD. The checking of the building description is difficult to realize because no current software solution can check all the requirements of a building description. Therefore, it was necessary to develop customized model-checking rules in DesiteMD. The functionality of the model-checking rules is proven by an example model. The advantages and disadvantages of the developed checking rules and methods are analyzed, as well as the database which provides the model checking content of the rules. Based on the results, recommendations for further research were given.

Keywords: model checking, content checking, validation checking, BIM.

1 INTRODUCTION

The digitalization of society is progressing continuously and is finding its way into all areas of life. The development and implementation of BIM is a key issue in the digitization of the construction industry. BIM has already been adapted in many areas of the construction industry and is enjoying continued growth.

It is obvious that the use of automated checks allows to reduce time-consuming and costly manual inspections. Potential profiteers in this context are building owners. They can expect faster building permit procedures, improved control and observation of planning as well as simplified verification of their requirements [1], [2].

Although various research approaches have already been carried out in this area, these have focused on the goal of testing directives, standards or laws in a model. Contractual regulations, such as the tenant building description, have so far gone unnoticed. Especially for the owners, the benefits of BIM have not yet been apparent. For this reason, owners are hesitant to push the use of BIM, which considerably slows down the widespread implementation of BIM. The testing of model qualities and contents is an enrichment for the contracting authority. This enables the owner to control the planning at an early stage of the project. Due to the increasing complexity of buildings and model size, automated model checking becomes more and more necessary.

The aim of this work is to enable the content checking of models as well as the model checking of tenant building specifications for professional project developers, to promote the use and increase the request of the BIM method for professional owners.



For this reason, this work concentrates on project developers with a special focus on commercial real estate development. Especially in this area, highly professionalized tenants place high requirements on the project developers. The tenants usually bundle their versatile requirements in a tenant building description, which makes a significant impact on the design of the building. In order to meet these requirements, an automated planning control is a great advantage, according to time, cost and quality.

2 ANALYSIS OF THE CHECKING TASK

2.1 Content checking

For the development of a model checking procedure, the understanding and precise definition of the contents to be checked is essential. For this reason, it is needed to explain the object of research in more detail. Content checking deals with the question, if all necessary object information's are in the model [3]. The definition of the model content or also called model qualities takes place through the contractual description of the BIM process. Here, the objectives and use cases to be pursued as well as the contents are determined by the employer, in the Employer's Information Requirements (EIR). The description of the BIM deliverables provides a basis for modelling and for quality control. The EIR does not only define the services to be provided, but also define the quality in which the model contents must be available. This means that the employers define, which geometric and alphanumeric information must be contained in each object of a model. The definition is usually done according to the Level of Development (LOD). The LOD describes the design of a model component using the Level of Geometry (LOG) and Level of Information (LOI). At this point of time, there are several LOD standards. The used standard is up to the employer. In further consideration, the LOD definition of the BIM-Forum is used [4].

2.2 Validation checking

The tenant building description represents, like the building description, a detailed specification of the construction target. However, it should be distinguished that the tenant defines the exact construction task for the landlord. This means that the tenant has a significant influence on the property. For commercial tenants, for example supermarket operators, who have no interest in owning the building, are such contractual regulations widespread.

A look at the contents of building descriptions on supermarkets has shown, that the structure and the LOD of the requirements described are highly variable. This results in more or fewer freedoms of the future landlord in the design of the project. In addition, it becomes clear that not all characteristics are defined in the tenant description and will be determined by the tenant during the planning or execution process. Furthermore, tenants define minimum requirements for building components or specify certain design variants.

In addition to the content consideration, the structuring of the requirements is also important for the validation checking. Three structural factors were identified in the building descriptions. These structural elements can be subdivided into different levels (Fig. 1), which makes it possible to guarantee an exactly structured, object-related description of the building. Fig. 1 shows that the contents of the building description are described by a task and then by a element-room or room-element relation. This means, that first the task like brickwork and then the wall-room or room-wall relation will be described.

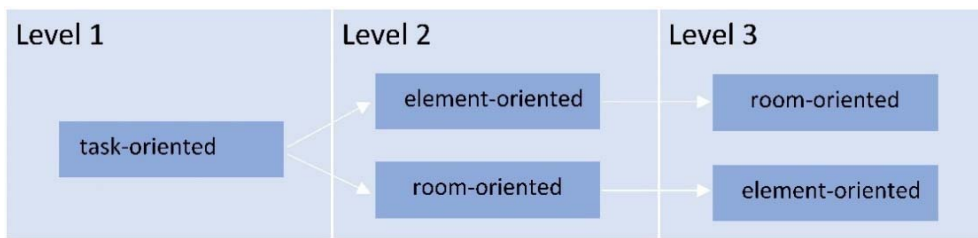


Figure 1: Structural elements of a tenant-based building description ordered in layers.

3 STATE OF THE ART

3.1 Model checking

The checking of models can be divided into three domains. One of these areas is the test software, which is used to make the building information accessible, to execute rule sets and to transfer the test results to the user. The building model contains the necessary information's of the building, which are checked by the rules executed by the software. The rules represent the algorithm to be executed by the software. They are distributed with the applications and can be combined, modified or extended according to the requirements of the project. In addition, it is possible to develop new rules, but these must be supported by the functions of the software. It becomes clear that the development of the model check depends on the available building information, the supported functions of the software as well as the available rules. The kind of rules which will be performed can be divided into content, validation and adaptive checking. Adaptive checking describes objects which must be modelled according to their constrains [3].

The rules can be stored using a black- or white-box method. The black-box method is not open and non-transparent, because the rule is not accessible, so the user does not know how the rule works. Instead, the white-box method gives the user full access to the code, which defines the rule. The analysis of available checking software solutions on the market has shown that the rules are mainly implemented as a black box method. This means that the user has no possibility to understand the content and process of the rule, not to mention to create new rules. The application of the rules is therefore limited to the customization of checking parameters [5]. For this reason, the checks of validation and tenant building description were tried to cover with existing checking rules from different software solutions. Table 1 gives a brief overview of available model checking solutions and a software called BIMQ for LOD definitions. The table shows that the possibility of changing and creating new rules is very limited. Out of this reason, the program DesiteMD was used to create new rules, if the checking was not possible with available rules. This software solution provides a programming interface (API) which makes it possible to implement and execute user-specific checking rules. So, the created checking rules are open, comprehensible for the user and adaptable according to a white-box method.

3.2 Data exchange

The structuring of a tenant building description is similar to the model-based approach of BIM, because in a model, the structure is created on the basis of a component-space relationship. The components contained in the model form spaces; spaces in turn define the

Table 1: Software overview based on Eilif [3].

	Software	Solibri Model Checker	dRofus	BIMserver	Revit Model Checker	Desite MD	BIMQ
Functions	Clash detection	X	-	X	-	X	-
	Content Checking	X	X	X	X	X	Development of Content
	Validation Checking	X	X	X	X	X	-
	Visualization of checking results	X	X	X	-	X	-
	BCF	X	-	-	-	X	-
Rule sets	Rule sets implemented	X	X	X	X	X	-
	Modification of existing rules	X	X	X	X	X	-
	Combination of existing rules	X	-	X	X	X	-
	Development of new rules	Coming next	-	X	-	X	-
Data exchange	Exchange formats	IFC	IFC	IFC	RVT, IFC	IFC, CPI-XML	-
	Direct Exchange links between Software	-	Revit u. Archi-CAD	-	-	Revit	-
	Adaptive Checking	-	Revit Plug-in	-	-	-	AIA Editor

function and the associated space-forming components. This creates a relationship between space and component, which is a cornerstone of model-based work. This scheme is also used in the IFC. The order system of the information provided by the IFC is subdivided from the building to a floor to a room. By these aggregation relations the data model of the IFC reflects the space program of the building description one by one [5], [6]. Therefore the IFC describes the contents of the tenant building description sufficiently exactly. In addition, the application of a neutral data format makes it possible to use BIM across different data interfaces, in the sense of OpenBIM, and to apply the checking procedures to all projects.

4 IMPLEMENTATION OF THE CONTENT CHECKING

The aim of this content check is the verification of a previously defined component related LOD. To perform the content checking, the language driven LOD concept was converted into a database and executed by an automated rule set. The outcomes can be divided into three results: Model components are below, equal to or exceed the defined LOD. The checking of the LOG as well as the LOI is implemented by querying object properties for their definition. For example, the rule will check, if the thickness of a door leaf is defined and if it has a value between 0.1 and 5 cm. The example shows that the usefulness of the checked property value can also be demonstrated within the quality check by limiting the value range or name space. The LOG is not checked directly on the basis of the geometry. Direct access to the LOG via the geometric description of the element would be possible, but has not been pursued further, due to its lack of feasible results. Out of this the LOG is checked on implicitly derived information. For instance, if a wall has layers, reinforcement, openings or a window lintel. These checks of model elements are done by checking the connection between the component. For example, openings and possibly windows should be integrated in walls of

the LOG 300 or, for the LOD 350, fasteners such as reinforcing irons or an offset brick. The check is performed by retrieving property relationships of the elements, by calculating a contact or a collision between the elements and write this contact or relation into the properties. Afterwards a normal content check can be done [4].

The first attempts to test the LOD were made with BIMQ and SMC. In BIMQ, the LOD requirements to comply with were defined and imported into the predefined rule by SMC. The result was that the implementation for checking of geometric properties and relations was difficult and the previously defined checking contents could not be completely covered. Since the model check with BIMQ and Solibri Model Checker proved to be insufficient, the check was implemented in DesiteMD. The access to the API in DesiteMD made it possible to compensate for missing functions of the program as well as to exceed them by SMC. Thus, a collision or contact check could be done to establish a link between wall and fall or offset brick. The quality properties to be checked can be performed on the example model (Fig. 2) and stored in a table (Fig. 3). For test purposes, the sample model consists of four walls, each with different levels of detail. On the basis of this examination it could be shown that a quality examination with reference to the LOD is possible.

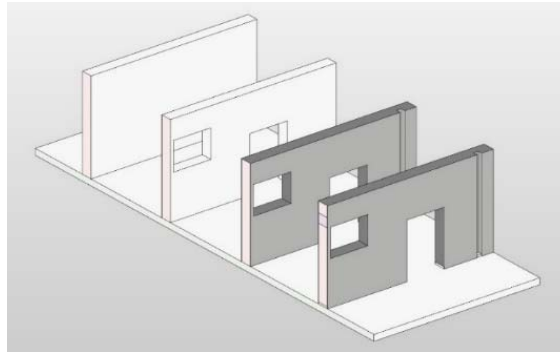


Figure 2: Deferent LOD-definition of a wall according to Reinhardt and Bedrick [4].

Vorbedingungen		Attributregeln		
Name	Datentyp	Fall 'undefiniert'	Bedingung (Muster)	
LOD 300				
Abmessungen:Volumen	xs:double	OK	>0,0001	
Abhängigkeiten:Abhängigkeit oben	xs:string	OK	*	
Abhängigkeiten:Abhängigkeit unten	xs:string	OK	*	
ifcRepresentationType	xs:string	OK	*	
cpOpening	xs:string	OK	*	
cpSchicht	xs:string	OK	*	
cpID	xs:string	OK	*	
cpName	xs:string	OK	"vorhanden"	
cpName	xs:string	OK	!"unnamed"	
cpMaterialName	xs:string	OK	*	

Figure 3: Definition of the checked attributes in the rule.

5 IMPLEMENTATION OF THE VALIDATION CHECKING

Based on the results of the content checking, the check of the tenant building description can be done. So, the content checking is always done before the validation checking to ensure that all necessary properties are included. The checking procedure is shown in Fig. 4. The first steps are used to check project-, component- and room-related properties. If the test of the characteristics is passed, the model can be checked for geometrical relations, for component space relations as well as for general building requirements, such as the length of the building or rental space.

How far the objects correspond to the respective predefined properties is neglected in the checking of relations. This is the reason, why object properties must be checked in advance. This procedure ensures that the relations checked in the check correspond to those of the required objects in the tenant building description. A supermarket store was created as an example model for the practical testing of the model checking procedure.

In the implementation of the content checking procedure, it became clear that checking parts can be done by existing software solutions, but a continuous checking in one software solution is not possible. For this reason, a consistent solution for checking the tenant building description in DesiteMD was developed. In a first step, checking rules were created for the test procedure shown in Fig. 4. The focus of this development is based on the checking of component space relationships. For this kind of checking it was essential to create relations of objects by geometric operations, like clash detection. Because relations in the IFC can be

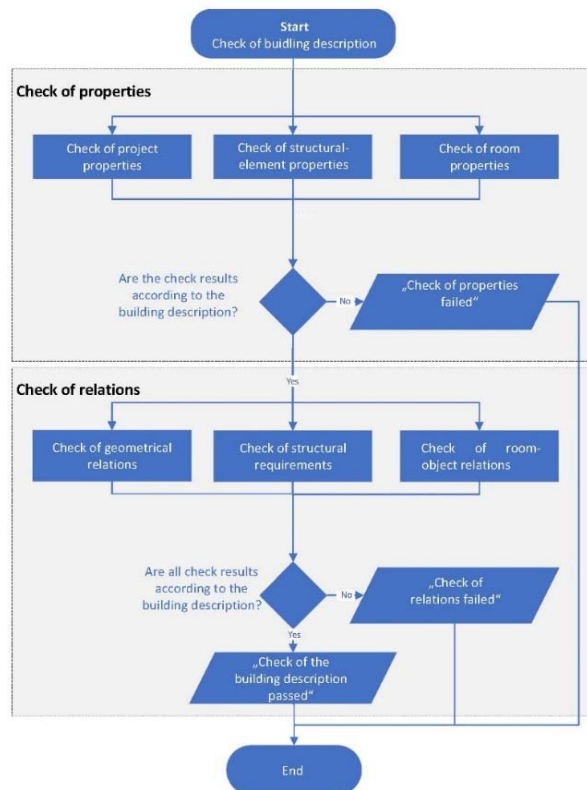


Figure 4: Sequence of the checking algorithm/procedure in a flow chart.



Figure 5: Extract of the data model to describe the relation of the room and wall.

missing or don't match with the model, it is safer to recreate the relations. In order to follow the idea of the BIM method further, all model requirements were stored in a relational database. This enables all project participants to access to the contents of the tenant description and to use them for checking or for modelling which is called adaptive checking. Fig. 5 shows an example of a database model in which the relationship between the component and the room is represented by the mapping of wall types to room types.

The requirements stored in the database can be exported at any time as checking parameters and loaded into the prepared rules. In addition to the checking rules and parameters, the user must be provided with an intuitive user interface and the option of plausibility checking. For this purpose, the option for creating own forms provided by DesiteMD was used (Fig. 6). It is not enough to find planning errors within the inspection process; planning errors must also be communicated. The BIM Collaboration Format (BCF) could be used for this purpose. BCF makes it possible, to link faulty objects with an image and a comment to communicate the mistakes with planners. The software provides natively a formula for BCF communication as shown in Fig. 7.

Figure 6: Implemented checking form in DesiteMD.



Figure 7: Visualization of the checking result in the BCF form provided by DesiteMD.

6 CONCLUSION

It could be proved that the examination of the model qualities as well as the building description can be implemented with simple methods. As a result, a flexible and practical solution could be created for the entire planning process. In addition, a plausibility check can be integrated, and results and comments can be exchanged via the BCF. The application of this new developed model checks shows that automated model checking provides a faster and safer solution than a manual check. The storage of the test parameters in a database is another strength of this method. The database helps to ensure the flexibility, completeness and traceability of the requirements to be checked. The database also enables the user to adapt the object properties to be checked to the specific planning phase or BIM use case. Finally, it is to say that the checking of the LOG is still very theoretical and not used today. Other ways of defining the LOG could be predefined objects with an included LOG property. But for this solution, the modelling in BIM authoring tools will be not flexible enough. So, the research for checking the LOG has still potentials. The validation checking of the LOI as well as the checking of content, is very common. But if this content is getting to complex or out of the software functionality, like in dRofus or Solibri, API's are necessary to enable the users the full functionality of the software. Furthermore, the user will be able to understand and to customize rules, so that a black box method becomes a white box method. In further research the implementation of standardised classification and characteristic properties systems will be analysed. In Germany this could be the DIN SPEC 91400 which is based on the STL-Bau.

REFERENCES

- [1] Borrmann, A. & König, M., Building Information Modeling. *Wendehorst Bautechnische Zahlentafeln*, vol. 36, ed. U. Vismann, Springer Vieweg: Wiesbaden, pp. 1475–1485, 2018.
- [2] Eastman, C., Lee, J.-M., Jeong, Y.-S. & Lee, J.-K., Automatic rule-based checking of building designs. *Automation in Construction*, **18**(8), pp. 1011–1033, 2009.
- [3] Eilif, H., BIM-based Model Checking (BMC). *Building Information Modeling. Applications and Practices*, pp. 33–61, 2015.
- [4] Reinhardt, J. & Bedrick, J., Level of development specification part I. *BIM FORUM (2019)*, Online. <https://bimforum.org/wp-content/uploads/2019/04/LOD-Spec-2019-Part-I-and-Guide-2019-04-29.pdf>. Accessed on: 15 Sep. 2019.
- [5] Borrmann, A. et al., *Building Information Modeling – Technology Foundation and Industry Practice*, Springer Vieweg, pp. 369–380, 2018.
- [6] Schiller, K. & Faschingbauer, G., *Die BIM-Anwendung der DIN SPEC 91400 1*, Auflage, Beuth Verlag GmbH: Berlin, Wien und Zürich, p. 82, 2016.

