In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

Chapter 24
BIM-based Code Compliance Checking

Cornelius Preidel, André Borrmann

Abstract In the construction industry, a large number of codes and guidelines define technical specifications and standardized requirements to ensure a building's structural stability, accessibility, and energy efficiency, among others. Today, checking the compliance with the applicable guidelines is an iterative, manual process which is based to a large extent on 2D drawings. In consequence, this process is cumbersome, time-consuming and error-prone. With the increasing adoption of digital methods in the construction industry, most importantly Building Information Modeling (BIM), new technologies are available to improve and partially automate this process. In a BIM-based construction project, digital models that include 3D geometric as well as semantic information comprehensively describe the building to be erected across the different involved disciplines. This rich information provides an excellent basis for automating the code compliance checking process. With Automated Code Compliance Checking, not only a higher degree of compliance with the different regulations can be achieved, but also a significant reduction of effort is possible. The chapter first discusses the major challenges of Automated Code Compliance Checking. Subsequently, representative available software solutions are presented and current research activities are discussed. Finally, an outlook for the development of code compliance checking in the construction industry is given.

24.1 Introduction

In the construction industry, a large number of codes and guidelines define technical specifications and standardized requirements to ensure a building's structural stability, accessibility, and energy efficiency, among others. There is a large variety of

Cornelius Preidel · André Borrmann
Technical University of Munich, Chair of Computational Modeling and Simulation, Arcisstr. 21, 80333 Munich, Germany
e-mail: cornelius.preidel@tum.de, andre.borrmann@tum.de

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

regulations covering different life-cycle phases and disciplines within a construction project. Depending on the respective international, national or regional legislation, the regulations must not only be followed by the architects and planning consultants in the course their design activities, but also be checked by the building authority officers when granting the building permission (Hjelseth, 2015).

In general, a code or guideline describes a certain discipline-related application context and requests the compliance with a number constraints. These conditions and constraints can be represented in different ways, ranging from running text over graphical representations to parameter tables. In Fig. 24.1, a selection of different ways for representing conditions and requirements is depicted. So far, the checking of the building design for compliance with the guidelines is an iterative and largely manual process, characterized by high effort, costs and error-rates. In the conventional procedure, 2D drawings are checked manually by the building authority for their compliance with the applicable guidelines and codes. Whenever a change in the building design comes into effect, the checks for the affected elements have to be repeated. In consequence, the manual checking process demands not only an advanced level of knowledge and experience regarding the appropriate guidelines but also a high degree of skill and care of the responsible planner.

With the introduction of digital methods, in particular *Building Information Modeling (BIM)*, and the development of data standards for digital models in the construction industry (e.g., IFC), new tools become available which provide a very suitable basis for improving and optimizing of this process (Nisbet et al., 2009; Hjelseth, 2015). During the different phases of a BIM-based construction project, models are created by the various stakeholders resulting in a comprehensive digital representation of the building. It is straight-forward to use the available high-level

The access route for pedestrians / wheelchair users shall not be steeper than 1:20. For distances of less than 3 metres, it may be steeper, but not more than 1:12. The access route shall have clear width of a minimum of 1,8 m and obstacles shall be placed so that they do not reduce that width. Maximum cross fall shall be 2 %. The access route shall have a horizontal landing at the start and end of the incline, plus a horizontal landing for every 0,6 m of incline. The landing shall be a minimum of 1,6 m deep. Minimum clear height shall be 2,25 m for the full width of the defined walking zone of the entire access route including crossing points.

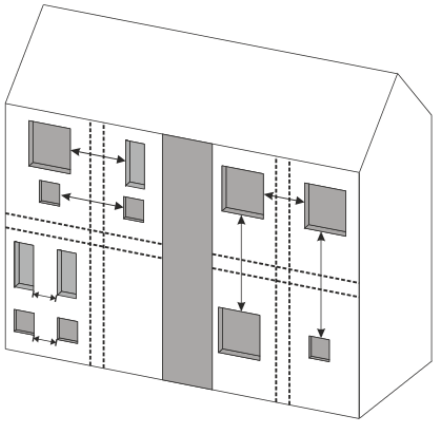

Fig. 24.1 Selection of usual representation styles of requirements within guidelines left – Extract of a guideline for accessibility, Norwegian Standard NS 11001-1 (2009) right – Restrictions for the placement of openings in exterior walls for fire safety, UK Fire Code Part B4 (U. K. Building Regulations, 2007)

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

information for a semi or even full automation of the checking processes, resulting in Automated Code Compliance Checking.

24.2 Challenges of Automated Code Compliance Checking

In order to discuss the major challenges of Automated Code Compliance Checking the common structure and basic components of the process will be presented first. Eastman et al. (2009b) divide the overall process into four components: Translation of the Rules in a Machine-Readable Language, Preparation of the Building Model Data, Execution of the Checking Process and, finally, the Preparation and Representation of the Checking Results. The resulting structure is shown in Fig. 24.2.

The translation of the contents of the codes and guidelines into a machine-readable language represents the starting point and is therefore the core task of an Automated Code Compliance Checking. Two essentially different approaches can be distinguished here:

The significantly easier way of translation is based on the direct transfer of the checking process in hard-coded program routines or methods. This means that the digitization of the contents of a code or guideline focuses on the definition of machine-readable algorithms, which are usually hidden from the user. Therefore, the readability of the translated rules for the user is limited and an involvement of a user in the encoding process is disregarded. As a result, the execution of the checking process is a hidden procedure, in which the user does not have an insight. Also, extensions and modifications are only possible by incorporating the software vendor. Such a process, which makes only the ingoing and outgoing information visible, but not the processing procedure itself, is called the *Black-Box* method (see Fig. 24.3) according to the general system theory (Von Bertalanffy, 1972). The major advantage of this method is the comparatively low error rate of the overall process because of the closedness and the direct access to the internal data structures of the code checking system.

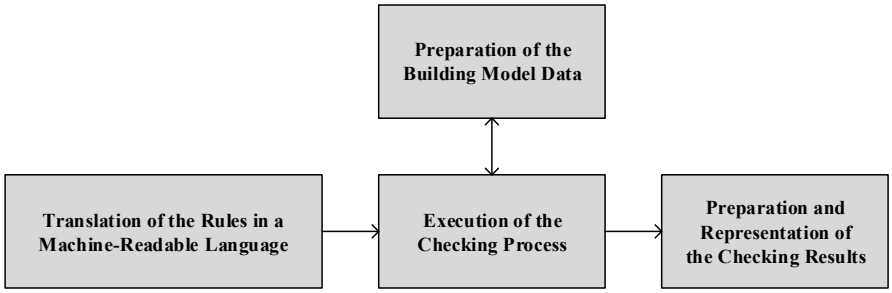

Fig. 24.2 Common Structure of an Automated Code Compliance Checking, inspired by Eastman et al. (2009b)

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

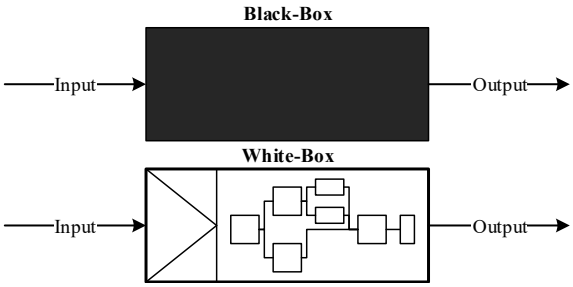

Fig. 24.3 Schematic Representation of a Black-Box and White-Box-Method

In contrast to these hidden procedures, *White-Box* methods make the internal processing steps visible and therefore comprehensible for the user. To achieve this transparency, the contents of the translated guideline or code must be readable not only by the machine but also by the user. The rules must be translated based on a code representation system (a language), which is a system of symbols and rules. These elements can be used for the sufficient description of objects, methods and relationships. So the major target is not only to cover all kinds of information a code or guideline could contain, but also to enable the user to understand and retrace information at any time and follow the progress step of the checking procedure. Although the development and implementation of such a system requires significantly more effort compared to the closed checking approach, it has major advantages for the execution of a checking task.

The transparency solves one of the major problems within the automation of processes: Despite the growing degree of digitization and automation in the construction industry, the planning engineers or the regulatory authorities, respectively, remain in charge of the outcome of each process step. This responsibility cannot be handed over to a machine or software application due to legal restrictions. Results of an automated process need to be treated with caution and in case of doubt must be checked manually. It is common practice in the *AEC* industry to check results periodically, e.g., by a manual calculation or comparison with rules of thumb. Such checks for plausibility must also be enabled within an automated framework for code checking, but this requires transparent and observable single processing steps. Since planning consultants are usually non-programmers, these checks cannot be combined with *Black-Box* methods due to the lack of transparency. According to Gross (1996), hidden procedures easily lead to a lack of trust of the results. The *White-Box* approach, on the other hand, represents an acceptable compromise in order to fulfill the major requirements. This raises the question if a full automation without involvement or feedback by the user makes sense at all. Since semantics in guidelines may be ambiguous, they need to be interpreted by a human, who has the necessary experience, knowledge and responsibility. Accordingly, it is advisable to implement a semi-automated approach.

After translating the contents of a guideline into a machine-readable language, an interpreting instance has to execute the directives. This processing is closely related to the contents of the building information model, since information must be ac-

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

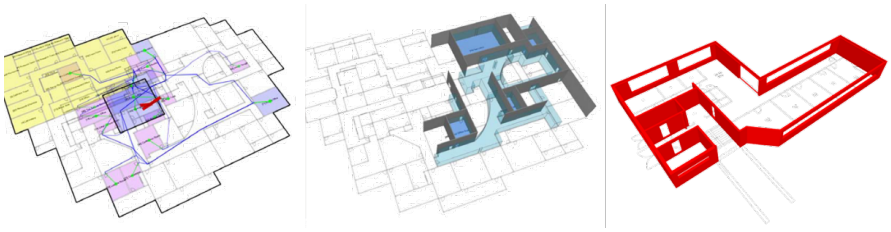

Fig. 24.4 Geometric representation of model subsets for different purposes; Left: Fire Escape Routing; Middle: Accessibility & Circulation; Right: Fire-Safety related building components; inspired by Solihin & Eastman (2015)

cessed, retrieved or derived. The accuracy, correctness and consistency of the building model is a basic prerequisite for the following checking process and therefore a basic condition in order to produce resilient results (Kulusjärvi, 2012). Although there is a continuous development of non-proprietary and open data standards, especially the IFC standard, Beetz et al. (2009) point out that a complete correctness of the data standard can only be achieved by providing a formal rigid data structure. Therefore, the generally valid formulation of a checking process for a specific data standard is quite difficult and can only be realized by a preprocessing step, which checks and prepares the data model. The correctness of building model information is outlined in the following Sect. 24.3.

Since a rule usually applies only to a certain subset of data, it is recommended to create and prepare this subset before the rule is checked (Solihin & Eastman, 2015). In Fig. 24.4 different subsets and derivations of a model for different purposes are shown.

At this point, there is a high demand to implement solutions which enable quality and consistency checks for preparing the models for the subsequent Code Compliance Checking.

As a last step, the results of a checking process must be reported so that the responsible person can understand the intended meaning of the detected problem to be able to initiate the correct post-process, i.e. solve the detected noncompliance. Therefore, the detected problems should be presented as a written report or, better, digitally communicated to the responsible person, e.g. using the *BIM Collaboration Format (BCF)* (buildingSMART, 2016a) (see Chap. 15).

24.3 Formal and content-related correctness of building models

As described in Sect. 24.2 the results of a checking process are highly dependent on the correctness and availability of the information in the underlying BIM model. Since a process cannot produce correct results based on incorrect information, this correctness of the digital building model is an essential prerequisite for following code checking processes. The necessity of corresponding checking processes were

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

introduced by various BIM guidelines, such as the *Singapore BIM Guideline* (BCA Singapore, 2013), *COBIM* (Kulusjärvi, 2012) or the *PAS 1192-2 (2013)*. According to these guidelines all kind of models have to pass quality gates at certain milestones (e.g., when they are exchanged or submitted) in order to preserve the overall quality.

Although the overall correctness of a model is often subsumed under the general term data quality, it can be divided in two parts: the formal part and the content-related correctness of a BIM model.

First of all, the information provided by a BIM model must fulfill formal criteria, which means that the information follows defined "grammar rules." Usually these rules are defined by the syntax of the data model, e.g., IFC, which is used for the representation of information. These requirements can be extended by further project-wide requirements, which the project participants contractually agree on in the beginning of a building project. Such requirements can contain individual constraints which information should fulfill in order that all the stakeholders have a unified interpretation of the contents of a BIM model, such that the information is interpreted in the same way. Templates for project-wide requirements can be found in several guidelines, such as in the *GSA BIM Guide* (GSA, 2007). Usually these requirements are written down in modeling requirements and provided as part of the *BIM Execution Plan*. Since formal criteria are straight-forward rules (e.g., checking the availability or the data type of certain attributes), these conditions are easy to check. Unlike the formal criteria, the content-related criteria are significantly more complicated to check, since they require the interpretation of the information. Content-related checks include, e.g., the compliance with reasonable boundaries and the consistency of the provided information.

Next to the correctness of single BIM models, the validity of multiple composed models must also be taken into account. According to the *Federated Model Approach*, each stakeholder, who is responsible for discipline-specific model contents according to the requirement specifications, has to submit a BIM model when reaching specific milestones (BCA Singapore, 2013). These submitted models result in a composed overall model, which finally describes a comprehensive description of the building to be constructed. Required information for model checking often refers to different discipline models and so not only the quality of a single model, but the quality of the overall model must be taken into account. This applies particularly to intersecting building components, redundant or contradictory information.

24.4 Selected software products

In the past decades, a number of commercial software tools for Automated Code Compliance Checking have been introduced. A selection of these products is depicted in the chronological diagram in Fig. 24.5 next to some of the research approaches which will be discussed in Sect. 24.5. The number of available approaches shows the continuously growing significance of this application area during the last years.

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

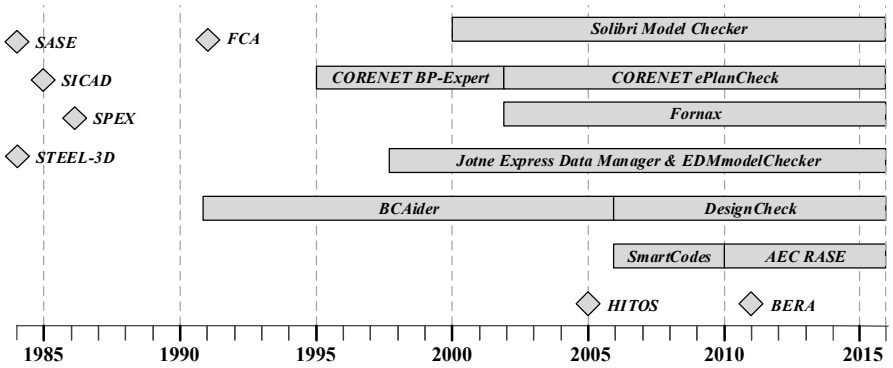

Fig. 24.5 Chronology of selected research approaches as well as commercial products, which focus on the Automated Code Compliance Checking, inspired by (Dimyadi & Amor, 2013)

In the following subsections, selected software solutions and applications for Automated Code Compliance Checking will be discussed.

24.4.1 CORENET

In 1995 the *Building Construction Authority (BCA)* in Singapore started the platform *CORENET* as a first representative of a common national submission platform. The basic intention of this platform is to collect all kinds of information related to a construction project and optimize the processes with the help of digital methods and tools. One of these tools is the application *CORENET BP-Expert*, which aims to check the compliance of digital 2D-based drawings with regulations regarding accessibility and fire safety. In 1998 *CORENET* was enabled to work with the IFC standard and therefore extended by 3D compliance checking. The current version of the tool was first published in 2002 as *CORENET e-Plan Check* and provides a code compliance checking feature of a digital building model regarding a large extent of the Singaporean regulations in terms of building control, accessibility, fire safety as well as environmental healthcare (Dimyadi & Amor, 2013).

The checking processes within *CORENET* are based on hard-coded routines and therefore the algorithms, process steps and methods are not transparent for the user. The overall process is structured into three basic phases. In a first step, the model information is checked for availability of the information in the required form to be processed. Subsequently, in a second step, the model is searched for the missing information in underlying information layers. If the missing information cannot be found here, it is created in a last step with the help of information derivation (Eastman et al., 2009b).

In order to enable such a preparation of the BIM model, the company *novaCITYNETS* developed a *C++* library of hard-coded methods, called *FORNAX*. These methods contain routines which are able to represent semantic objects and map this

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

kind of information onto the IFC data schema. With these methods, not only data preparation routines, but also code checking processes may be defined. These defined routines can be stored directly in the model (Eastman et al., 2009b).

The development of *CORENET* and *FORNAX* represents one of the earliest, but even nowadays one of the most advanced approaches for the automation of code compliance checking. In 2008 *CORENET* covered almost 92% of the Singaporean Guideline *Integrated Building Plan* and 77% of the *Integrated Building Service*. The *CORENET* platform is used by approximately 2500 companies in the AEC sector (Eastman et al., 2009b). The basic principle of the *FORNAX* objects and methods provided also the basis for several other approaches aimed at the digital representation of code checking processes, such as discussed by Xu, Solihin & Huang (2001).

24.4.2 Jotne Express Data Manager

Besides the developments in Singapore, in 1998 the Norwegian technology company *Jotne EPM Technology (2016)* published the collaboration platform *Express Data Manager (EDM)*. This platform is built upon an object-oriented database, which makes direct use of the EXPRESS data modeling language (ISO 10303-11) – the basis of the IFC data model. The platform is intended to manage product model data of various engineering domains but is especially focused on the AEC industry.

The data modeling language *EXPRESS* is used within the *EDM* to achieve a high degree of flexibility for the handling of the information in the data models, since it enables the user to perform queries and derivations of information. *EXPRESS* is a part of the *Standard for the exchange of product model data* (STEP, ISO 10303) and provides EDM platform compatibility with a large number of different data model formats (cf. Chap. 6). Since this data model language is also the base for the IFC data modelling, EDM is particularly compatible with this format. For the conversion of the data, EDM provides an integrated conversion tool, the *EDMmodelConverter*. EDM is often used as a basis for the development of import/export routines of third-party applications(Wix & Espedokken, 2004).

For model checking, *EDM* provides the *EDMmodelChecker* tool which can be used for the definition and formulation of rules. Rules and guidelines must be translated into *EXPRESS* checking routines and can be applied afterwards on BIM models. Since the formulation of these processes with the data modeling language demands significant programming skills, it is only of limited suitability for non-programmers, such as architects or engineers.

24.4.3 BIM Assure

BIM Assure is an online platform for model checking and was released by the company *Invicara* in 2016. This platform is one of the first representatives of an on-

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

line checking system, but marks a general trend in the construction industry toward increasing adoption of cloud-based systems. Currently, several software vendors develop solutions for managing building models in a Common Data Environment (CDE) according to the British specification PAS 1192-2 (2013). Since model checking is an important step of the CDE process, which affects all discipline models and must be performed frequently, a cloud-based solution seems to be a reasonable approach.

The *BIM Assure* platform retrieves the required model information via an add-in plugin for Autodesk Revit, which means that it currently works only for Revit models. Apart from a basic project and user management, *BIM Assure* also provides checking functionalities as a main feature. As a first step, all retrieved information from the Revit model is "normalized", which means that the information is categorized according to a library maintained by *Invicara*. Elements, which could not be detected, and therefore not categorized, can be adjusted manually by a custom mapping. For the actual checking, the user can compose a specific checking process by choosing analysis templates, which can be adjusted by a number of parameters. Detected non-geometric problems, e.g., wrong attributes, can be immediately fixed within the BIM Assure environment.

Even if *BIM Assure* does not provide as rich functionalities as comparable desktop applications so far, the software represents a first commercial approach for taking the model checking process to the cloud. Since the information of the BIM model as well as the definition of the check routines are stored online, the execution of the rules can be conducted without specific local hard- or software. The execution of the checks can even be carried out automatically and become part of an automated workflow.

24.4.4 Solibri Model Checker

The *Solibri Model Checker (SMC)* is a software system for BIM-based model quality and code checking, which was published by the Finnish company *Solibri (2016)* in 2000. All processes within this application are based on information contained in IFC models, which themselves are mapped onto an internal data model. Although the IFC format is standardized, the available BIM authoring tools often export IFC data in a slightly individual way. Therefore, SMC's data mapping is based on hard-coded routines adjusted to the individual export mechanisms of the authoring tools. In this manner, SMC is able to read and correctly interpret IFC files exported from a wide range of BIM authoring tools.

In order to harmonize and bring the data of different discipline models and authoring tools together, the SMC makes extensive use of a classification approach. A classification represents a categorization of building components based on specific information found in the data model. With this classification mechanism, information from different building models can not only be filtered but also prepared for the following checking process.

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

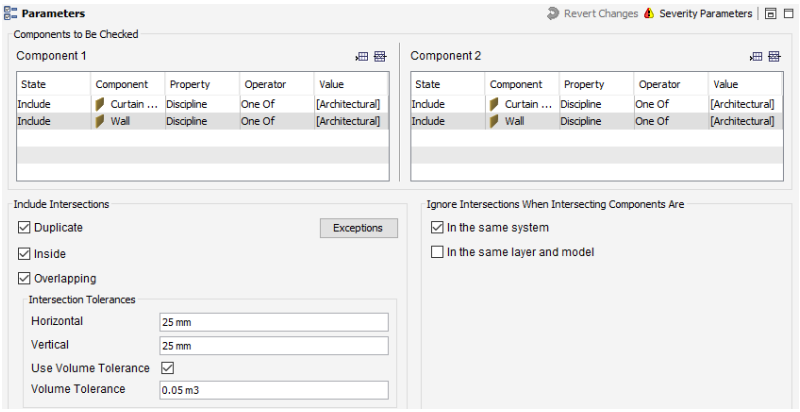

Fig. 24.6 Interface for adjusting a rule template in the SMC (Solibri, 2016)

The core of the SMC's code compliance checking routines is the Ruleset Manager, which provides a basic library of 42 single rule templates. Such a rule template represents a hard-coded standard checking procedure, which can be adjusted by a limited number of given parameters. These rule templates can be composed or adjusted by the user regarding his individual requirements. In order to exchange and share such defined rules, the rule composition can be stored as rule sets. As an example, the interface for adjusting a rule for checking intersections between defined architectural building components is shown in Fig. 24.6.

However, since the composition of such rules demands not only deep knowledge regarding the rules themselves but also expertise in data modeling and IFC structures, this feature is mostly used by expert users and not the planning consultants themselves. Large parts of the user interface of SMC are based on a line-by-line composition of requirements and adjustments, which can be very complicated for non-expert users. Therefore, most of the users currently use the predefined rules provided by SMC and focus on basic architectural checks, such as the completeness of information or the intersection of building components (Fig. 24.7). More specific rulesets like COBie compliance, ADA/ABA accessibility or fire escape routing are available as fee-based extensions.

Since SMC focuses on the quality check of building model information, it is not applicable for modifying and writing IFC data. However, results of checking routines may be exported and published as PDF or BCF reports.

For specific applications, *Solibri* provides an Application Programming Interface (API), which can be used for the composition as well as definition of rules, but is not publicly available. Based on this API and in close cooperation with the Georgia Institute of Technology, the *Design Assessment Tool* was developed, which can be used for checking requirements by the U.S. General Services Administration for courthouses. These requirements are defined by the U.S. Courts Design Guide, a guideline that includes spacing, safety, environmental as well as building service requirements (Eastman, 2009a).

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

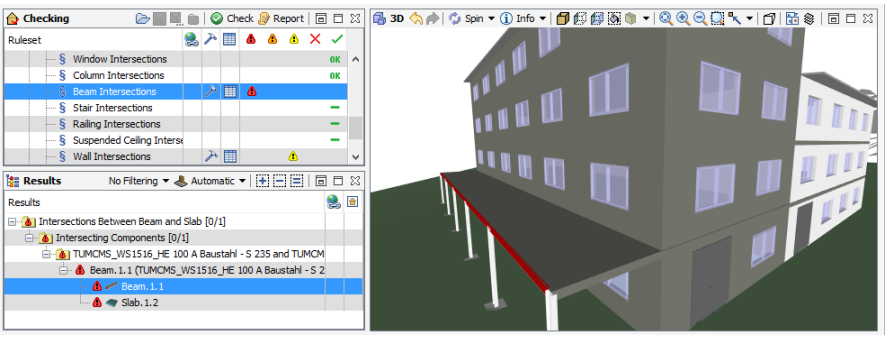

Fig. 24.7 User interface of the Solibri Model Checker (Solibri, 2016)

24.5 Current research

Given the high potential benefits of automated code compliance checking on the one hand, and the numerous challenges involved on the other hand, code checking is an intensively investigated research subject (Navari, 2018). Among the most challenging issues is the transformation of human-readable text into computer-interpretable code. A significant number of current research projects follow the white-box approach by defining a proprietary computer language that is close to the domain concepts and comparatively easy to use and thus allows domain experts to manually encode the regulations into a computer-processable representation.

A representative of these research approaches is the *SMARTCodes* project, which was initiated in 2006 by the *International Code Council (ICC)*. The *SMARTCodes* represent in principle a data exchange protocol used for standardizing and unifying common elements of a regulation. The defined elements are supposed to be provided in a library. Although the developments within this research project were discontinued by the ICC in 2010 due to lack of funding, the approach was continuously investigated by the companies AEC3 and DigitalAlchemy (Dimyadi & Amor, 2013).

For the formalization of guidelines, the *SMARTCodes* uses the *RASE* syntax. With this tool, all elements of a regulation rule can be categorized into the four different classes ***R**equirement, **A**pplicability, **S**elect and **E**xceptions*. In this way, even complex contents of codes and guidelines may be formalized and divided into these basic components. The result of this categorization can be illustrated and marked within the regulations' running text (Hjelseth & Nisbet, 2011). It must be noted, however, that this markup procedure can only be seen as a preprocessing step as the result cannot be directly interpreted by the computer. An exemplary application of the markup language on a regulatory text is shown in Fig. 24.8.

Another language-based approach is the *Building Environment Rule and Analysis (BERA) Language* (Lee (2011)). It provides a much more powerful tool than RASE, as it is able to provide not only a descriptive categorization but an algorithmic implementation of regulatory rules. On the one hand, BERA is supposed to meet the

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

<R>**Standard NS 11001-1, Clause: 5.2 Dimensioning an <a>access route to a building**

<R> The <a>access route for <s>pedestrians</s> <s>wheelchair users</s> shall <r>not be steeper than 1:20 </r>. <E>For <a>distances of less than 3 metres, it may be steeper, but <r>not more than 1:12</r>.</E> </R>
<R>The <a>access route shall have <r>clear width of a minimum of 1,8 m</r> and <r>obstacles shall be placed so that they do not reduce that width </r>.<r>Maximum cross fall shall be 2 %.</r> </R>
<R>The <a>access route shall have <r>a horizontal landing at the start and end of the in-cline<r>, plus <r>a horizontal landing for every 0,6 m of incline</r>. <r>The landing shall be a minimum of 1,6 m deep.</r> </R>
<R> <r>Minimum clear height shall be 2,25 m </r>for the full width of the defined walking zone of the entire <a>access route including crossing points. </R> </R>

Fig. 24.8 Formalization of the Norwegian code NS 11001-1:2009 according to the RASE syntax (Hjelseth & Nisbet, 2011)

high requirements regarding the handling of building model data and, on the other hand, to enable the formulation of rules and guidelines. The design of *BERA* is inspired by popular languages known from data base management and handling. A special feature of *BERA* is the relatively easy readability by humans and the direct access to the contents of a building information model.

To automate the translation of human-readable regulations into computer code, a number of research projects have investigated the application of natural-language processing (NLP) (Zhang and El-Gohary, 2015; Salama and El-Gohary, 2016; Zhang and El-Gohary, 2016; Uhm et al., 2015; Lee et al., 2016). Although the results seem to be promising, it is important to note that comparatively straight-forward and well-defined regulations have been used for the conducted cases studies.

An approach which aims for an increased involvement of the domain experts in the translation process is the development of a *Visual Programming Language (VPL)* for Code Checking. The basic idea of this approach is that a checking procedure can be represented as a visual, graph-based flow of information. For the definition of this graphical representation, a *VPL* uses visual instead of textual elements, so that also non-programmers like engineers and architects can understand the intended meaning. In this way, the visualization of the process is used as a visual assistance and supporting system for the user, who is able to adjust the checking procedure according to his requirements (Preidel & Borrmann, 2015, 2016).

An entirely different approach is to enable the machine-readability of codes and guidelines in general. This requires that all guideline documents maintained by the various international, national and regional boards must be rewritten, so that they are readable for humans as well as machines. Usually today's codes and guidelines are not written with the intention to be translated into a machine-readable language. Although there are no specific approaches for the guidelines in the construction industry, there are available mechanisms to enable such a two-way readability, e.g. the programming language *Inform* (Graham, 2005). Though the benefits of a formally defined and computer-processable regulations are clear, this shift will require

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

extensive resources and it will thus take a number of years until such solutions are conceivable.

To cover the various approaches undertaken so far, buildingSMART (2016b) has initiated the *Regulatory Room*, which aims to provide a platform for an open discussion for regulators, researchers, developers as well as end users. In this way, the platform strives for an open BIM-based approach, which will lead to common mechanisms, templates or definitions supporting building regulations.

24.6 Summary

In the AEC domain, there exists a large number of regulations and guidelines which must be fulfilled by the building design. Today, the checking process is conducted mostly manually in a laborious and error-prone process. BIM provides an excellent basis for automating Code Compliance Checking as the digital building model provides in principle the required geometric and semantic information. In many cases, however, a pre-processing of the BIM model is required, as information needed for checking some regulations (e.g. excavation routes) are not directly provided by the model and must be computed or derived beforehand. The bigger challenge, however, lies in the transformation of the regulatory texts, which are written to be understood by humans, into computer-processable formats. In many cases, contextual expert knowledge is required to a large extent in order to interpret a rule correctly. Very often, today's regulations contain soft or even ambiguous expressions that require the careful interpretation of a regulatory expert. In consequence, approaches for a full automation of code compliance are still in an early research stage. More promising solutions rely on a semi-automated translation of the rules under the guidance and supervision of a domain expert.

From a general perspective, it is important to distinguish white-box solutions, which implement an open, transparent approach for rule representation based on an accessible rule repository, from black-box solutions, which typically rely on hidden hard-coded implementations of specific regulations. Whereas white-box solutions are open to verification, modification and extension by domain experts, black-box solutions are currently more powerful as they can make direct use of internal algorithms and data structures of the code checking system.

The software solutions discussed in Sect. 24.4 show that most of the commercial tools focus on black-box approaches. They are able to to perform basic model checks regarding element classifications, attribute provision and collision detection, but can also provide advanced code compliance checking for specific national or international regulations in the domains of fire safety, accessibility or escape routing, for example. However, these solutions only provide limited room for customization and flexibility for the user, i.e. domain experts are typically not able to alter the rule implementation or create new ones. In consequence, both planning consultants and construction authorities are forced to use the predefined sets of rules available. Nevertheless, the currently available solutions document the enormous potential of

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

Automated Code Compliance Checking, which reduce the effort of the checking processes significantly.

References

BCA Singapore (2013). *Singapore BIM Guide – Version 2*. Singapore: Building and Construction Authority Singapore.

Beetz, J., van Leeuwen, J. & de Vries, B. (2009). IfcOWL: A case of transforming EXPRESS schemas into ontologies. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 23*, pp 89–101.

buildingSMART (2016a). *BCF Introduction*. Retrieved from http://www.buildingsmart-tech.org/specifications/bcf-releases (Accessed 13.05.2016).

buildingSMART (2016b). *Regulatory Room*. Retrieved from http://buildingsmart.org/standards/standards-organization/rooms/regulatory-room/ (Accessed 25.08.2016).

Dimyadi, J. & Amor, R. (2013). *Automated Building Code Compliance Checking – Where is it at?* Proceedings of CIB WBC 2013, pp 172–185.

Eastman, C. (2009a). Automated Assessment of Early Concept Designs. *Architectural Design, 79*, pp 52–57.

Eastman, C., Lee, J., Jeong, Y. & Lee, J. (2009b). Automatic rule-based checking of building designs. *Automation in Construction, 18*(8), pp 1011–1033.

Graham, N. (2005). *Natural Language, Semantic Analysis and Interactive Fiction*. St Anne's College, Oxford.

Gross, M. D. (1996). Why can't CAD be more like Lego? CKB, a program for building construction kits. *Automation in Construction, 5*, pp 285–300.

GSA (2007). *BIM Guide for Spatial Program ValidationGSA BIM Guide Series 02*. Retrieved from http://www.gsa.gov/bim/ (Accessed 26.08.2016).

Hjelseth, E. (2015). *Public BIM-based model checking solutions: lessons learned from Singapore and Norway*. 149, pp 421–436. Retrieved from http://library.witpress.com/viewpaper.asp?pcode=BIM15-035-1 (Accessed 07.01.2018).

Hjelseth, E. & Nisbet, N. (2011). *Capturing normative constraints by use of the semantic mark-up RASE methodology*. Proceedings of the 28th International Conference of CIB W78, pp 26–28.

Invicara (2016). *BIM Assure*. Retrieved from http://bimassure.com/ (Accessed 24.08.2016).

Jotne EPM Technology (2016). *About Jotne IT*. Retrieved from http://www.epmtech.jotne.com/about-jotne-it (Accessed 26.08.2016).

Kulusjärvi, H. (2012). *Common BIM Requirements (COBIM) – Part 6 Quality Assurance*. Retrieved from http://www.en.buildingsmart.kotisivukone.com/.

Lee, H., Lee, J. K., Park, S., & Kim, I. (2016). Translating building legislation into a computer-executable format for evaluating building permit requirements. *Automation in Construction, 71*, pp. 49–61

Lee, J. K. (2011). *Building Environment Rule and Analysis (BERA) Language*. PhD, Georgia Institute of Technology.

Nawari, N. O. (2018). *Building Information Modeling: Automated Code Checking and Compliance Processes*. CRC Press.

Nisbet, N., Wix, J. & Conover, D. (2009). *The Future of Virtual Construction and Regulation Checking*. Virtual Futures for Design, Construction & Procurement. Blackwell Publishing Ltd.

NS 11001-1 (2009). *Universal design of building works – Part 1: Buildings open to the public*. Norwegian Standards.

PAS 1192-2 (2013). *Specification for information management for the capital/delivery phase of construction projects using building information modelling*. British Standards Institution.

Preidel, C. & Borrmann, A. (2015). *Automated Code Compliance Checking Based on a Visual Language and Building Information Modeling*. ISARC. Oulu, Finland.

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

Preidel, C. & Borrmann, A. (2016). *Integrating Relational Algebra into a Visual Code Checking Language for Information Retrieval from Building Information Models*. Proc. of the International Conference on Computing in Civil and Building Engineering (ICCCBE). Osaka, Japan.

Salama, D. M., & El-Gohary, N. M. (2016). Semantic Text Classification for Supporting Automated Compliance Checking in Construction. *Journal of Computing in Civil Engineering, 30*(1).

Solibri (2016). *Solibri Model Checker*. Retrieved from http://www.solibri.com/solibri-model-checker-v9-1-available-today/ (Accessed 08.03.2016).

Solihin, W. & Eastman, C. (2015). Classification of rules for automated BIM rule checking development. *Automation in Construction, 53*, pp 69–82.

U. K. Building Regulations (2007). *UK Fire Code Part B4: Fire safety*.

Uhm, M., Lee, G., Park, Y., Kim, S., Jung, J., & Lee, J. K. (2015). Requirements for computational rule checking of requests for proposals (RFPs) for building designs in South Korea. *Advanced Engineering Informatics, 29*(3), pp. 602–615

Von Bertalanffy, L. (1972). The History and Status of General Systems Theory. *Academy of Management Journal, 15*, pp 407–426.

Wix, J. & Espedokken, K. (2004). *Building code and code checking developments in the UK and Norway*.

Xu, R. (2001). *Code Checking and Visualization of an Architecture Design*. IEEE Visualization. Austin, TX, USA.

Zhang, J., & El-Gohary, N. M. (2015). Automated Information Transformation for Automated Regulatory Compliance Checking in Construction. *Journal of Computing in Civil Engineering, 29*(4).

Zhang, J., & El-Gohary, N. M. (2016). Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking. *Journal of Computing in Civil Engineering, 30*(2).

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

In: Borrmann, A.; König, M.; Koch, C.; Beetz, J. (Eds): Building Information Modeling - Technology Foundations and Industry Practice, Springer, 2018, DOI: 10.1007/978-3-319-92862-3

Index