TECHNISCHE UNIVERSITÄT MÜNCHEN

LEHRSTUHL FÜR COMPUTERGESTÜTZTE MODELLIERUNG UND SIMULATION

DEVELOPMENT OF METHODS FOR MODEL-BASED QUALITY ASSURANCE IN RAILWAY INFRASTRUCTURE DESIGN

Marco Häußler

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der
Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Prof. Dr.-Ing. Stephan Freudenstein
Prüfer der Dissertation:	1.	Prof. Dr.-Ing. André Borrmann
	2.	Prof. Ghang Lee (Yonsei University, Seoul, Korea)

Die Dissertation wurde am 11.05.2021 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 12.08.2021 angenommen.

Für meine liebe Maria, die mich in der Zeit der Promotion wundervoll unterstützt hat. Danke!

Abstract

Through the introduction of modern, digital technologies to the process of executing construction projects, the construction design industry aims to achieve various goals, among them to increase efficiency, to improve quality, and to achieve more reliable cost estimation. Digitization in construction design is primarily associated with Building Information Modeling (BIM). In the BIM method, a digital, three-dimensional building model is created and enriched with information concerning the construction component or project. The resulting digital twin is used for various aspects, for example to conduct evaluations, analyses, and simulations over the life cycle – i.e., the design, construction, and operation phases – of the structure.

This dissertation focuses on the design process of railway structures. A significant portion of this work is still undertaken in a more or less conventional manner using 2D CAD applications. The drawings that are produced contain information that cannot be used intelligently. BIM aims to address this shortcoming. Designing in the railway sector is subject to a large number of regulations that have to be observed and is characterized by a high degree of complexity due to the large number of interdependent subsections, e.g., superstructure, civil engineering, control, communication and safety system or catenary systems.

Practical experience shows that the design quality is made only partially more error-free through the application of BIM. In fact, the method can introduce new sources of error, which can lead to acceptance problems due to the inexperience of many users. In addition, the large amount of available data and information that a building model may typically contain makes manual checking more difficult. However, all data and information is then available digitally and can be evaluated accordingly with the help of so-called automatisms.

This dissertation examines which sources of error arise in connection with the BIM method. This is then used to systematically develop automatisms to support the checking of models from the field of designing railway structures. This serves, on the one hand, to facilitate the systematic verification of models and on the other to assist in keeping the very large amounts of data related to models manageable.

A quality assurance concept is presented which covers the domains construction design, clashes, semantics, construction process as well as quantities and costs. The concept is modular, so that extensions, but also simplifications are possible. Different routines are developed for each domain, which illuminate the construction model to be tested from different perspectives.

A central element of this work is the combination of the three procedures ‘Business Process Model and Notation’, ‘Decision Model and Notation’ and BIM. The ‘Business Process Model and Notation’ is a graphical modeling language that can be used to visualize process flows using standardized symbols. The ‘Decision Model and Notation’ is also standardized and enables the simple and clear creation of decision rules within business processes. This approach is used both to check the model for compliance with guidelines (Code Compliance Checking) and to create models in compliance with guidelines (Knowledge Based Engineering).

The methods developed were verified using several real-world case studies. The approaches for Code Compliance Checking and Knowledge Based Engineering were also evaluated with regard to their applicability. Using the ‘Business Process Model and Notation’ and ‘Decision Model and Notation’, 68% of the rule sets examined can be represented digitally. The approach was also evaluated against the acceptance criteria for KBE applications compiled in the literature. A corresponding analysis reveals that the approach fulfills 73 % of the weighted criteria. Both evaluation results are considered good and promising for the present work. A further aspect is the possible time savings in checking or creating models that this method promises, which measurements show amount to between 75% and 97% compared to manual processing.

Zusammenfassung

Die Bauplanungswirtschaft verfolgt mit der Einführung moderner, digitaler Technologien bei der Bearbeitung von Bauprojekten verschiedene Ziele. Hierzu gehören unter anderem: Effizienzsteigerung, Qualitätssteigerung und Erhöhung von Kostensicherheit. Die Digitalisierung in der Bauplanung wird in erster Linie mit Building Information Modeling (BIM) in Zusammenhang gebracht. Bei der BIM-Methode wird ein digitales, dreidimensionales Bauwerksmodell erstellt und mit Informationen, z.B. zum Bauteil oder Projekt, angereichert. Der dabei entstehende digitale Zwilling wird für verschiedene Aspekte, z.B. Auswertungen, Analysen, Simulationen etc., über den Lebenszyklus des Bauwerks – Planung, Bau und Betrieb – genutzt.

Die vorliegende Dissertation fokussiert auf den Planungsprozess von Eisenbahnbauwerken. Dieser wird in weiten Teilen noch auf mehr oder minder konventionelle Weise mit 2D-CAD-Anwendungen geführt. Im Ergebnis werden Pläne gefertigt, deren Informationen nicht intelligent genutzt werden können. Diesem Schwachpunkt der Planung soll mit BIM begegnet werden. Planungen im Eisenbahnwesen unterliegen einer Vielzahl von zu beachtenden Regelwerken und zeichnen sich aufgrund der Vielzahl untereinander abhängiger Gewerke, wie z.B. Oberbau, Ingenieurbau, Leit- und Sicherungstechnik und Oberleitungsanlagen, durch eine hohe Komplexität aus.

Praktische Erfahrungen zeigen, dass die Qualität von Planungen allein durch die Anwendung von BIM nur bedingt fehlerfreier wird. Vielmehr entstehen neue Fehlerquellen, die aufgrund der Unerfahrenheit vieler Anwender zu Akzeptanzproblemen der Methode führen können. Die Vielzahl an verfügbaren Daten und Informationen, die am Bauwerksmodell mitgeführt werden, erschwert darüber hinaus die manuelle Prüfung. Allerdings stehen sämtliche Daten und Informationen digital zur Verfügung und können mit Hilfe von Automatismen entsprechend ausgewertet werden.

In der vorliegenden Dissertation wird untersucht, welche Fehlerquellen im Zusammenhang mit der BIM-Methode existieren. Auf dieser Basis werden systematisch Automatismen entwickelt, die die Überprüfung von Modellen aus dem Bereich der Planung von Eisenbahnbauwerken unterstützen. Hierdurch soll einerseits eine systematische Prüfung von Modellen erreicht werden. Andererseits sollen die Automatismen dazu dienen, die sehr großen Datenmengen in Zusammenhang mit Modellen handhabbar zu halten.

Es wird ein Qualitätssicherungskonzept präsentiert, das die Domänen Konstruktion, Kollisionen, Semantik, Bauablauf sowie Mengen und Kosten umfasst. Das Konzept ist modular aufgebaut, so dass Erweiterungen, aber auch Vereinfachungen möglich werden. Zu jeder Domäne werden verschiedene Routinen entwickelt, die das zu prüfende Bauwerksmodell aus unterschiedlichen Perspektiven beleuchten.

Ein zentrales Element der Arbeit ist die Kombination der drei Verfahren ‚Business Process Model and Notation', ‚Decision Model and Notation' und BIM. Die ‚Business Process Model and Notation' ist eine grafische Modellierungssprache, mit der Prozessabläufe unter Verwendung von standardisierten Symbolen visualisiert werden können. Die ‚Decision Model and Notation‘ ist ebenso standardisiert und ermöglicht eine einfache und übersichtliche Erstellung von Entscheidungsregeln innerhalb von Geschäftsprozessen. Dieser Ansatz wird sowohl dafür eingesetzt, das Modell auf Richtlinienkonformität zu prüfen (Code Compliance Checking) als auch Modelle richtlinienkonform zu erstellen (Knowledge Based Engineering).

Die entwickelten Methoden wurden an Hand mehrerer Fallbeispiele aus der Praxis verifiziert. Die Ansätze zum Code Compliance Checking und Knowledge Based Engineering wurden außerdem in Hinblick auf ihre Anwendbarkeit bewertet. Mit Hilfe der ‚Business Process Model and Notation‘ und ‚Decision Model and Notation‘ sind 68 % der untersuchten Regelsätze digital darstellbar. Der Ansatz wurde ebenso gegen die in der Literatur zusammengetragenen Akzeptanzkriterien für KBE-Anwendungen bewertet. Es konnte herausgearbeitet werden, dass der Ansatz 73 % der gewichteten Kriterien erfüllt. Beide Bewertungsergebnisse werden für die vorliegende Arbeit als gut und vielversprechend eingestuft. Ergänzt wird dieses Bild durch die mögliche Zeitersparnis bei der Prüfung bzw. Erstellung von Modellen. Die gemessene Zeitersparnis liegt zwischen 75 % und 97 % im Vergleich zur manuellen Bearbeitung.

Contributions

This cumulative dissertation is based on three published, peer-reviewed research papers, which are presented in chapters 3 to 5.

Paper I

Marco Häußler, André Borrmann
Model-based quality assurance in railway infrastructure planning
Automation in Construction, Volume 109, 2020, 102971, ISSN 0926-5805
DOI: https://doi.org/10.1016/j.autcon.2019.102971.

Contributions:

Marco Häußler developed a concept for the quality assurance of railway models and developed methods to check railway models in four of the five given domains. André Borrmann supervised this study and reviewed the manuscript. All authors approved the final version.

Paper II

Marco Häußler, Sebastian Esser, André Borrmann
Code compliance checking of railway designs by integrating BIM, BPMN and DMN
Automation in Construction, Volume 121, 2021, 103427, ISSN 0926-5805
DOI: https://doi.org/10.1016/j.autcon.2020.103427.

Contributions:

Marco Häußler developed and analyzed a method to represent railway guidelines in a digital manner and to automate the checking process of railway models. Sebastian Esser contributed knowledge about IFC and self-developed software tools to analyze IFC data. André Borrmann supervised this study and reviewed the manuscript. All authors approved the final version.

Paper III

Marco Häußler, André Borrmann

Knowledge-based engineering in the context of railway design by integrating BIM, BPMN, DMN and the methodology for knowledge-based engineering applications (MOKA).
Journal of Information Technology in Construction (ITcon), Vol. 26, pg. 193-226,
DOI: 10.36680/j.itcon.2021.012

Contributions:

Marco Häußler developed a method to automate railway model generation based on the digital representation of railway guidelines and the methodical knowledge of domain experts. André Borrmann supervised this study and reviewed the manuscript. All authors approved the final version.

Table of Contents

Abbreviations

2

2D	two dimensional

3

3D	three dimensional

4

4D	four dimensional

5

5D	five dimensional

A

ACC	Automated Code Checking
AEC	Architecture, Engineering and Construction
AEG	Allgemeines Eisenbahngesetz (General Railway Act)
AI	Artificial Intelligence
AKFF	Anforderungskatalog Feste Fahrbahn (catalogue of requirements for slab tracks)
API	Application programming interface

B

BIM	Building Information Modeling
BImSchV	Bundes-Immissionsschutz-Verordnung (Federal Immission Control Ordinance)
BPMN	Business Process Model and Notation

C

CAD	Computer Aided Design
CityGML	City Geography Markup Language
CPIxml	Construction Process Integration markup language
CSV	Comma-Separated Value

D

DB AG	Deutsche Bahn AG
DGUV	Deutsche Gesetzliche Unfallversicherung (German accident insurance)
DIKW	Data, Information, Knowledge, Wisdom
DIN	Deutsches Institut für Normung
DMN	Decision Model and Notation
DSM	Design Structure Matrix

E

EBA	Eisenbahnbundesamt (Federal Railway Administration)
EN	European Norm
EU	European Union

F

FG	Fertigstellungsgrad (degree of completion)

G

GIS	Geographic Information System

H

HC	Hard Clash
Hg	Höchstgeschwindigkeit (max. velocity)

HOAI	Honorarordnung für Architekten und Ingenieure (fee structure for architects and engineers)
HTML	Hypertext Markup Language
I	
IC	Irrelevant Clash
ICARE	Illustrations, Constraints, Activities, Rules and Entities
ICE	Intercity Express
ICT	information and communications technology
IDM	Information Delivery Manual
IDMVU	Infrastructure Data Management
IEC	International Electrotechnical Commisssion
IFC	Industry Foundation Classes
ISO	International Organization for Standardization
IT	Information technology
K	
KBE	Knowledge Based Engineering
KBS	Knowledge-based system
KM	Knowledge Management
L	
LoD	Level of Detail
LOD	Level of Development
LOG	Level of Geometry
LOI	Level of Information
Lt	Lasttonnen (load tons)
M	
MDG	Modelldetaillierungsgrad (Level of Development)
MOKA	Methodology for Knowledge-Based Engineering Applications
mvdXML	Model view definition extensible markup language
N	
NLP	Natural Language Processing
O	
OKSTRA	Objektkatalog für das Straßen- und Verkehrswesen (object catalog for roads and traffic)
OOBB	optimized oriented bounding box
P	
PDF	Portable Document Format
Q	
QL4BIM	Query Language for Building Information Modeling
R	
railML	rail markup language
RASE	Requirement, Applicability, Selection, Exception
RDF	Resource Description Framework
S	
SC	Soft Clash
SMC	Solibri Model Checker
SPARQL	SPARQL Protocol And RDF Query Language
SQL	Structured Query Language

T	
TSI	Technical specification interoperability
U	
UIC	International Union of Railways
UML	Unified Modeling Language
V	
VPL	Visual Programming Language
VzG	Verzeichnis zulässiger Geschwindigkeiten (directory of allowed velocity)
X	
XML	Extensible Markup Language

1 Introduction

1.1 Motivation

The decades-old process of planning railway structures is currently undergoing a digital transformation. The introduction of Building Information Modeling (BIM) in the railway sector aims to achieve various goals, among them to increase efficiency, to improve quality, and to achieve more reliable cost estimation. In the BIM method, a digital, three-dimensional building model is created and enriched with information concerning the construction component or project. The resulting digital twin is used for various aspects, for example to conduct evaluations, analyses, and simulations over the life cycle of the structure or building. A commonly cited major advantage of BIM is that it avoids and minimizes the occurrence of errors through the networked use of building models (see for example [1,2]). *Eastman* states, for example, that “design and coordination with 2D systems is error-prone, labor intensive and relies on long cycle-times. BIM addresses these problems […]. The benefits of BIM for subcontractors and fabricators include: […] reduced cycle-times for detailed design and production; elimination of almost all design coordination errors; lower engineering and detailing costs […]” [1]. Media articles on BIM even suggest that the creation of three-dimensional models alone, including their further use in construction process simulations and quantity and cost calculations, will increase the quality of construction planning (see [2,3]).

The planning of railway structures is heavily regulated by laws, guidelines and standards, among other things. On the one hand, this is due to the legally prescribed safety requirements in railway operations, and on the other, a consistently usable transport network can only be created with the help of defined specifications. Compliance with the specifications must be ensured throughout the entire life of a project – from design to construction. To this end, quality assurance teams are deployed in practice, usually as a downstream instance, and thus check prepared designs for conformity at the end of a work phase or at defined points in time during the project.

In practice, quality assurance measures are carried out to varying degrees in the context of BIM. These range from simple clash checks of the 3D model to extensive property checks and checks for compliance with guidelines. The latter are mainly found in the area of building design (see [4]). There is no comprehensive, generally applicable concept for quality assurance and thus no generally applicable definition of when a model is of high quality. While clash checks as well as simple property checks are now easily possible with the available software products, checks for semantic-geometric coherence, for example, are comparatively uncommon but also important. Checks for compliance with guidelines for infrastructure design are difficult to implement, as there are currently two major hurdles:

(1) Availability of standardized data schemes (see also Section 2.4)

The development of the Industry Foundation Classes (IFC) for infrastructure are currently still at the draft stage. A parameter-oriented, software-neutral data exchange format will not be available until projects have been completed. Currently, only software-specific data schemas are commonly used for the parameter-based description of infrastructure models. The development effort for a checking tool increases insofar as the technical procedures for checking models are to be used for all available software products.

(2) Availability of machine-readable rules for compliance checking

“In the Architecture, Engineering, and Construction (AEC) industry, building projects must be checked against numerous building codes for compliance. They are allowed to be executed only when compliance with all applicable rules of the building code has been guaranteed” [5]. Digital, machine-readable mapping of guidelines is an essential component for automated model checking. Currently machine-readable formats of the guidelines relevant to design are not available.

The introduction of BIM aims to increase the quality of planning. There is no standardized approach to model checking. The focus of the work presented here is on the development of digital procedures for checking and complying with specifications and guidelines in the design of railway structures. The related topic, known in academia as ‘code compliance checking’, is an essential part of the dissertation presented here. It is important to create and provide machine-readable rules. By solving this, it is also possible to incorporate design-relevant specifications already at the time of 3D model creation. This is the core content of the investigations on ‘Knowledge Based Engineering’, which aim to combine methodical expert knowledge as well as normative and fact-based knowledge digitally in such a way that 3D models can be created automatically. This in turn makes it possible to consider quality assurance as an integral part of design.

The dissertation is written under the aspect that the developed methods should be comprehensible for as large a circle of users as possible so that the methods can be adopted as widely as possible. As the research literature shows, user acceptance of automatism of design and checking tasks will be reached by using transparent coding approaches. Therefore, the checking routines in the topic area of code compliance checking and the approaches investigated in the area of knowledge-based engineering are designed to be more comprehensible with the help of visual programming languages.

1.2 Scope

The present dissertation is limited to railway construction and here specifically to the design of the associated structures. These are produced by different disciplines. In general, the main planning disciplines and their associated railway structures are as follows:

- **Traffic facilities**: alignment, superstructure, civil engineering, cable civil engineering, stations
- **Engineering structure**: bridges, tunnels, retaining structures, noise barriers, other load-bearing structures
- **Technical equipment**: catenary systems, control and safety systems, other electrotechnical systems

The disciplines of surveying, subsoil and environment also contribute significantly to the planning of railway structures and there are interdependencies between the individual disciplines and structures that must be considered both in testing and in the creation of models.

The work in this dissertation concerns primarily traffic facilities and engineering structures.

Numerous regulations govern the design, construction and operation of railway infrastructure. In Germany, this is due, among other things, to the national legal requirement under §4 of the General Railway Act (Allgemeines Eisenbahngesetz), which stipulates that railway infrastructures and vehicles must meet the requirements of public safety during construction and operation [6]. Railway companies are obliged to mitigate against hazards using all available and reasonable technical possibilities [7]. To meet this requirement, various regulations exist at both national and European levels. The following list applies to infrastructure facilities but there are further requirements for the operation of railways and for the personnel employed.

- laws (e.g., AEG)
- by-laws (e.g., EBO, BImschV)
- standards (e.g., DIN, DIN EN)
- technical specification (e.g., TSI)
- guidelines (e.g., of Deutsche Bahn AG or EBA)
- Catalog of requirements (e.g., AKFF)
- other regulations (e.g., DGUV regulation 78)
- “DB standards” (DBS)

Within the scope of this dissertation, the complexity of the regulations cannot be considered or mapped in its entirety. For our purposes, the guidelines set out by Deutsche Bahn AG are the most authoritative set of regulations for German railway infrastructure. They were “introduced by the Federal Railway Authority as a technical construction regulation by the building authorities and as a recognized rule of technology [...]” [8] and bring together the planning and construction requirements for the railway system, refer to other applicable types of regulations, such as laws and standards, or implicitly map the requirements of other types of regulations.

1.3 Approach

The approach taken in this thesis is to investigate the entire model creation process in the context of designing railway structures and to identify resulting sources of error. There are two main aspects:

Approach 1 – Model checking

The first part of the thesis examines the model checking process. In current planning practice, checking designs and the associated models for compliance with specifications with regard to the semantics of models and guidelines (code compliance checking) requires a significant amount of effort.

Approach 2 – Model creation

The model creation process is currently a predominantly manual process in practice, where different regulations and guidelines have to be taken into account. The second part of the thesis therefore examines the extent to which the 3D model creation process can already be automated. The use of machine-readable guidelines and expert knowledge leads to higher quality of 3D models (knowledge-based engineering).

In both cases, the objective is to use automation to support the user and perform routine tasks. This allows the quality assurance process to be designed efficiently.

Based on the state of the art presented in Chapter 2 and the finding that multiple evaluation criteria need to be considered in order to assess model quality, a quality assurance concept was developed as shown in Figure 1. This focuses on approach 1 and defines the five domains considered in this thesis, each of which requires different checks to be made. The domains are Construction Design, Clashes, Semantics, Construction sequence (4D) as well as Quantities and costs (5D). In addition, a standardized evaluation procedure for the test results was developed to make it possible to compare the quality achieved across models and projects.

Due to the currently still limited availability of a fully software-neutral data exchange format for railway structures (compare Section 2.4), both the software-neutral IFC format in the current state of development and proprietary data formats for the exchange of model information are used.

For this dissertation, the basic understanding of different levels of model detail as described in Section 2.3 is helpful. However, the methods developed follow a generic approach and are not limited to a specific project phase or level of detail.

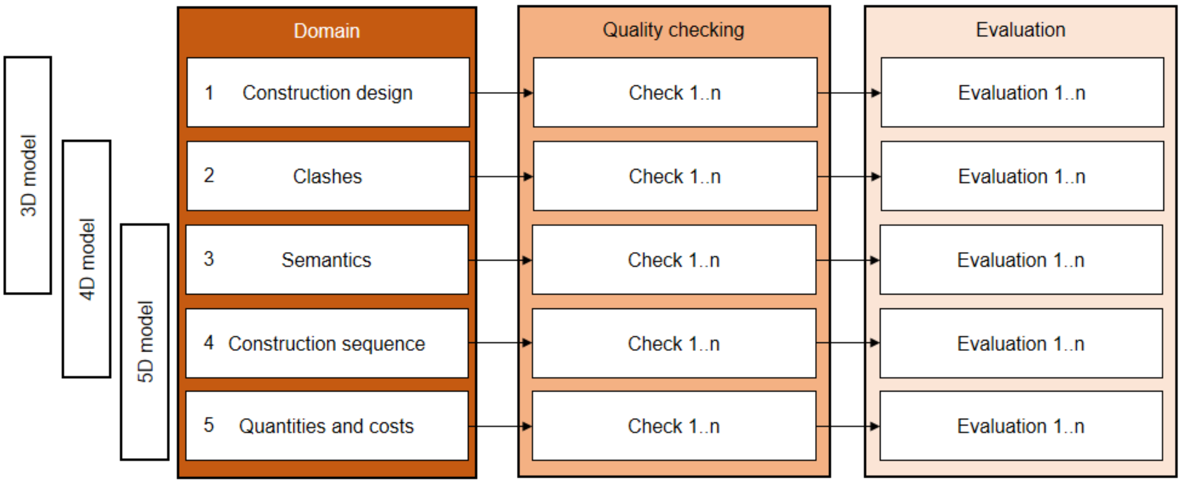


Figure 1: Quality assurance concept for model checking

The considerations presented in Chapter 3 deal with domains 2 to 5. Domain 1, which is located in the topic area 'Code Compliance Checking', is examined in more detail in Chapter 4. In Chapter 5, the developed methods in the domain 'Knowledge Based Engineering' are described in the context of model creation.

Regardless of the approach considered, the following principles are defined for the development of the technical methods:

(1) The methods must be applicable across all disciplines as far as possible.
(2) It must be possible for non-IT-savvy users to apply and evaluate the methods.
(3) The formalization of existing design rules is preferably carried out with a visual, software-neutral modeling language.
(4) If possible, the methods are managed using software-neutral data exchange formats.
(5) The methods are validated using practical examples.

In domains 2 to 5, simple if-then queries predominate, and these data-intensive test methods were developed with text-based source code.

To represent machine-readable rules in the context of 'Construction Design', the three methods 'Business Process Model and Notation' (BPMN), 'Decision Model and Notation' (DMN) and BIM are combined as a central element of the work. BPMN, standardized with ISO/IEC 19510:2013, has been developed as a generic approach for modeling business processes. BPMN is designed as a graphical modeling language, which makes it easy to represent the logic of a process in an understandable way. DMN was developed as a complementary language for the representation of decision rules. Both notations can be combined, allowing complex process and decision logics to be developed. The introduction of the generically designed notations into the discipline of railway design is intended to give due consideration to principles 1 through 3. This approach is used both to check the model for code compliance and to create models that conform to guidelines.

1.4 Research questions and hypotheses

Based on the aforementioned boundary conditions, the following questions arise, which will be addressed in this dissertation:

(1) Which errors can occur within the individual domains, but also across domains, and what are the sources of these errors?
(2) How can the contents of rules and requirements be formalized in a way that is machine-readable and thus digitally evaluable?
(3) Which possibilities, but also limitations, do the currently available data formats offer for the transfer of infrastructure models? What are the differences between explicit and implicit geometry descriptions?
(4) What hurdles exist for non-IT-savvy users in the context of model-based quality assurance?
(5) How can different types of checks be evaluated according to a standardized scheme?
(6) Which possibilities exist to automate the model creation process?

This leads to the following hypotheses for the dissertation:

(1) Several domains must be considered when checking models.
(2) Checking of model semantics is a central aspect.
(3) Using implicit geometry (see Section 2.4) description allows easier compliance checking.
(4) Compliance checking methods of parameterized models can also be used for model creation.

1.5 Structure of the dissertation

Chapter 2 presents the current state of the art in the context of this dissertation. It begins with a discussion of the international status of BIM in the railway sector. Then, the terms “quality” and “level of development” are examined, and the current state of data exchange formats, model checking, and automated model generation is presented.

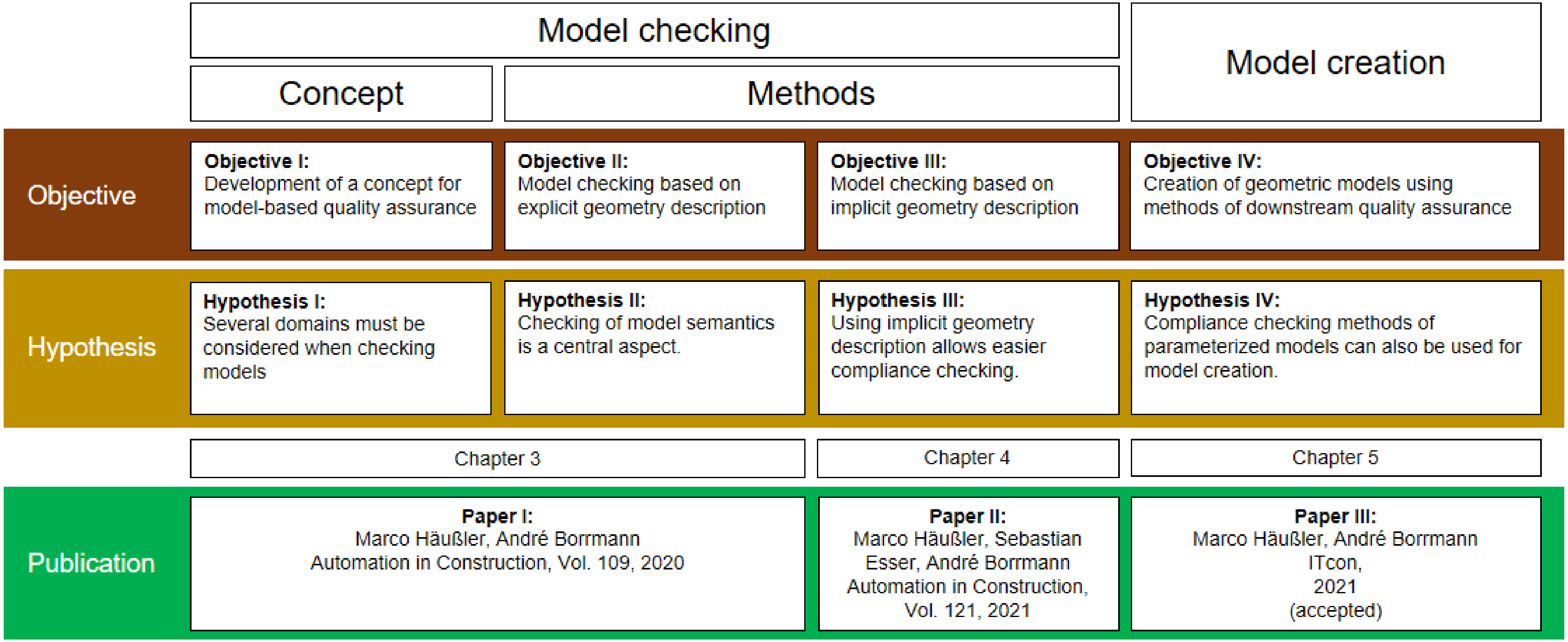
Figure 2 shows the remaining structure of the dissertation:

Chapter 3 presents the quality assurance concept of the work in general and the necessary methods for testing railway models for four of the five included domains. The methods presented are related to clash detection, checking of the construction process, quantity and cost determination, and verification of semantics from different perspectives. The testing methods are based on volumetric models.

Chapter 4 takes a closer look at the fifth domain, “Construction Design.” The checking of models in the context of code compliance checking is performed using the Business Process Model and Notation (BPMN) standardized by ISO/IEC 19510 in combination with the Decision Model and Notation (DMN). For this purpose, selected guidelines of Deutsche Bahn AG are analyzed and examined to determine the extent to which guideline contents can be mapped with the help of BPMN and DMN.

The methods for checking models for compliance with guidelines are adapted in **Chapter 5** with the aim of supporting the process of creating models from the outset. The examinations from the field of Knowledge Based Engineering aim to digitally combine methodical expert knowledge as well as normative and fact-based knowledge in such a way that 3D models can be created automatically.

The assignment of the objectives and hypotheses to the individual chapters is also shown in Figure 2.



	Model checking			Model creation
	Concept	Methods		
Objective	**Objective I:** Development of a concept for model-based quality assurance	**Objective II:** Model checking based on explicit geometry description	**Objective III:** Model checking based on implicit geometry description	**Objective IV:** Creation of geometric models using methods of downstream quality assurance
Hypothesis	**Hypothesis I:** Several domains must be considered when checking models	**Hypothesis II:** Checking of model semantics is a central aspect.	**Hypothesis III:** Using implicit geometry description allows easier compliance checking.	**Hypothesis IV:** Compliance checking methods of parameterized models can also be used for model creation.
	Chapter 3		Chapter 4	Chapter 5
Publication	**Paper I:** Marco Häußler, André Borrmann Automation in Construction, Vol. 109, 2020		**Paper II:** Marco Häußler, Sebastian Esser, André Borrmann Automation in Construction, Vol. 121, 2021	**Paper III:** Marco Häußler, André Borrmann ITcon, 2021 (accepted)

Figure 2: Structure of the dissertation

2 State of the art

This chapter provides an overview of the current state of the art in the main topics. A detailed presentation is given in chapters 3 to 5.

2.1 BIM in railway design

The BIM Roadmap [2] introduced in 2015 by the Federal Ministry of Transport and Digital Infrastructure marked the beginning of the stepwise introduction of BIM methodologies in the infrastructure sector in Germany. According to the Roadmap, BIM will be introduced in three phases (preparatory phase, extended pilot phase, BIM Niveau I) over a period from 2015 to 2020. All new infrastructure measures starting in 2020 are to be implemented using the BIM method.

The DB Station & Service AG – the operating company of the stations on the network of DB Netz AG – already began to introduce BIM in all new projects [9] as a mandatory part of the project from early 2017 onwards. In this context, specifications for the application of the BIM methodology were defined, which included statements on objectives, use cases, processes and the expected results. After successful completion of the monitoring of 13 pilot projects as part of the BIM4RAIL project [10] in 2019, the DB Netz AG – the operator of the German railway network – began implementing the BIM method in new projects in 2020 [11]. Initial company standards were likewise developed for 3D models, Employer's Information Requirements, BIM Execution Plans, collaboration processes and virtual design reviews.

BIM in railway construction is not a German phenomenon. The use of BIM is also being pushed internationally. The China Railway BIM Alliance, for example, is significantly involved in the development of the IFC standard for the railway sector [12,13]. Major projects are also being implemented using the BIM method, like the Beijing-Zhangjiakou high-speed railway [14] or the high-speed railway Changhua station [15]. Similar to the German Roadmap, South Korea has adopted the ‘Rail BIM 2030 Roadmap’ [16,17]. The goal of the plan is to implement artificial intelligence based design, engineering, and model quality checking by 2030. BIM is also being implemented by a variety of railway companies, including French railways [18], Swiss railways [19], Italian railways [20], Russian railways [21], Swedish, Danish, Norwegian and Finnish railways [22,23]. Internationally known railway projects that are implemented using the BIM method include the Doha Metro in Qatar [24], the HS2 in United Kingdom [25] or the project Rail Baltica [26].

The design and associated modeling of railway structures differs significantly from the design of buildings. Buildings are plane-oriented. The individual floors of a building are divided into rooms or areas (see Figure 3). Buildings can have regular, grid-like structures, as shown by *Gan et al.* [27] and *Sharma et al.* [28], or they can be based on freeform curves, see *Pottman et al.* [29].

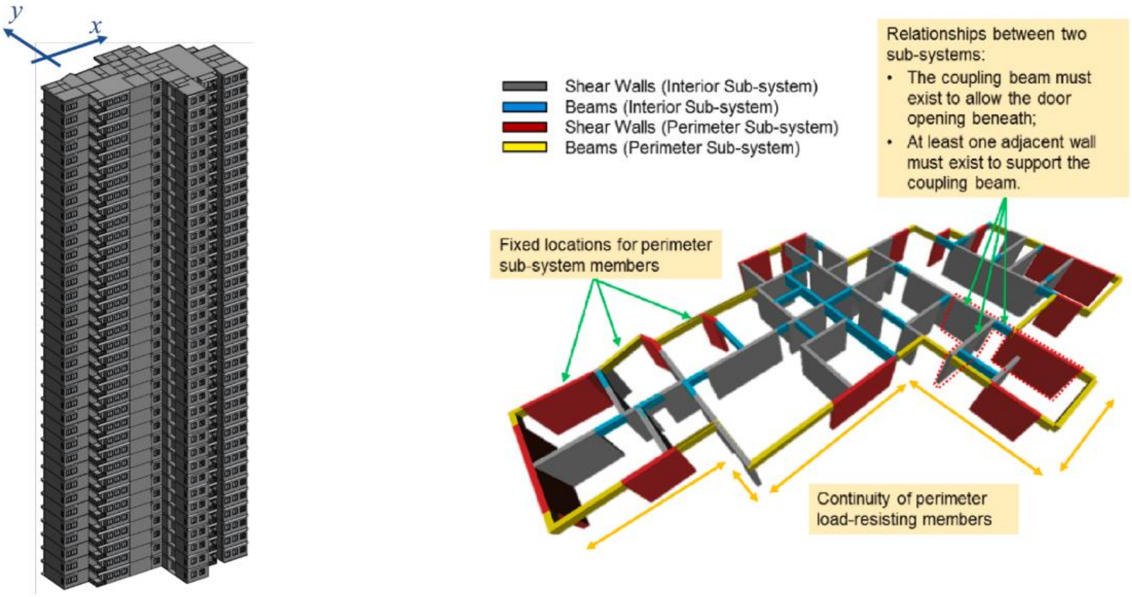


Figure 3: High-residential building and its separation into levels, rooms and spaces (image from Gan et al. [27]*)*

In contrast to buildings, infrastructure structures are linear structures that are related to the track alignment. The horizontal alignment represents the position information of a structure. The vertical alignment provides the associated height information. By superimposing horizontal and vertical alignment, the 3D space curve can be determined. By defining transverse profiles at discrete stations or parameter sets from which transverse cross sections can be calculated, a 3D model can be computed (see *Amann* [30], *Galiläer* [31] and Figure 4). An essential aspect in the modeling of infrastructure structures is the approach of parametric modeling (see *Borrmann et al.* [32] and *Vilgertshofer et al.* [33]). „The core concept is not to store the final outcome of the construction process, that is, an explicit geometric model, but instead the history of the individual construction operations" [32].

This philosophy is applied in a number of specialist applications. From the field of track design, the following tools are relevant: Civil 3D [34], OpenRail Designer [35], card_1 [36], Vestra Rail [37], ProVI [38] or Ferrovia [39]. In interaction with other disciplines, software solutions specific to building construction are sometimes integrated into the modeling workflow, as shown for example by *Pasetto et al.* [40]. The standards of DB Station & Service AG also accommodate the use of software (Allplan [41] und Revit [42]) originally designed for building projects. These and other software solutions are also used in the design of bridges and tunnels, e.g. Allplan Bridge [43], Infraworks 360 [44], Siemens NX [45] or CATIA [46]. Functioning interfaces that enable data exchange are crucial for the interaction of the various software products (see Section 2.4).

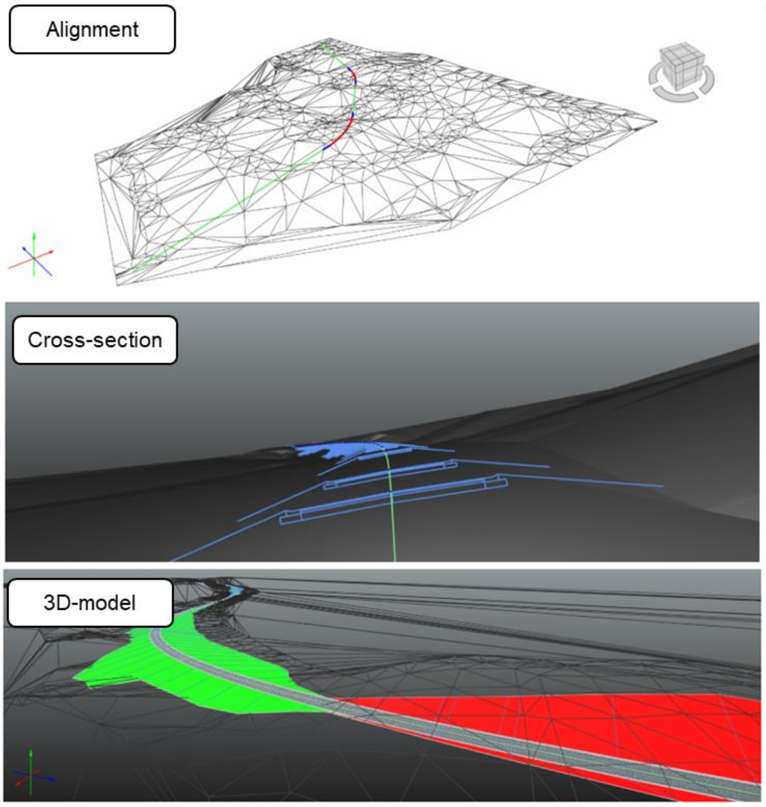


Figure 4: 3D-alignment based on horizontal and vertical alignment, cross-sections oriented along alignment, generated 3D-model (figure based on Amann [30]*)*

2.2 Quality

ISO 9000:2015 defines ‘quality’ as the “degree to which a set of inherent characteristics of an object fulfils requirements”. The term ‘requirement’ is defined as a “need or expectation that is stated, generally implied or obligatory” while “a specified requirement is one that is stated, for example in documented information” [47]. ISO 9000:2015 speaks of non-compliance or errors if the given requirements are not met.

Preidel [4] distinguishes between three different categories of quality in the context of BIM. While data quality is understood as ensuring the correct functioning of interfaces, compliance with modeling guidelines and agreements on model content represents quality of content. The correct implementation of standards and guidelines is understood as quality of design. The categories and the relationship are shown in Figure 5.

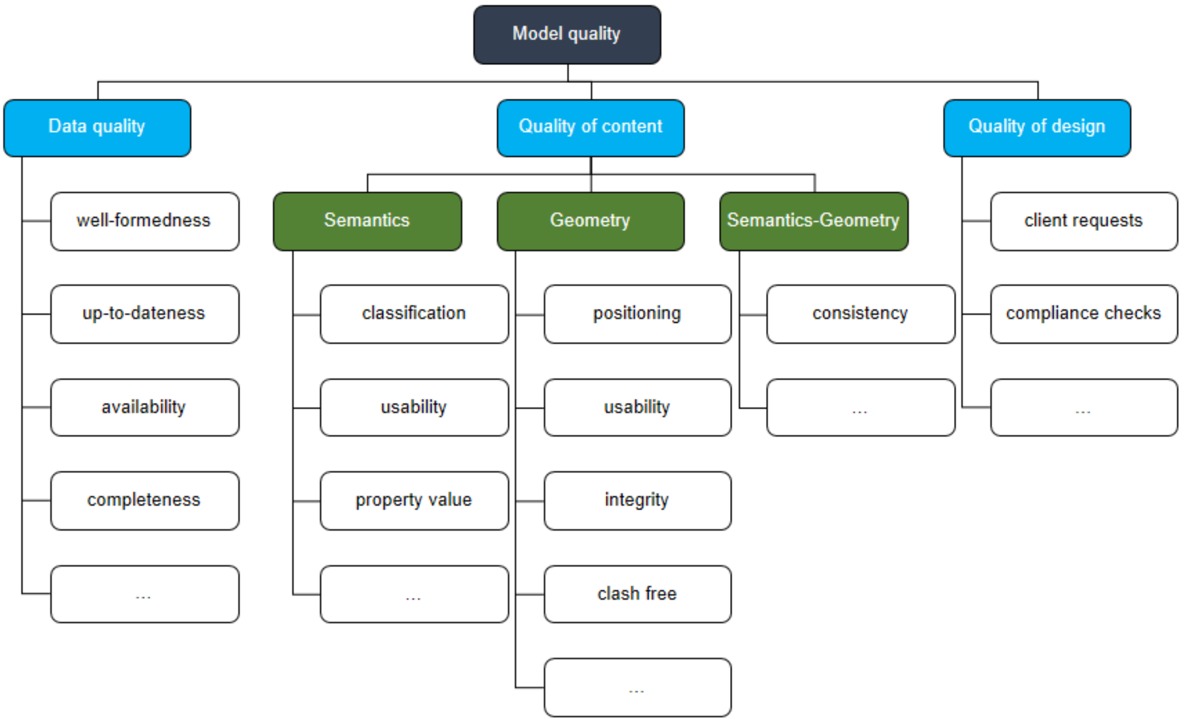


Figure 5: Categories of model quality, based on Preidel [4]

According to Kulusjärvi [48], two methods exist to ensure the quality of models, namely checking and analysis. “Checking refers to a method whereby the correctness of the information contained in a BIM file is verified. To determine the correctness of any information, it must be possible to compare or measure it against some reference information. [..] Analysis, on the other hand, produces information refined from the BIM, making it easier to interpret and assess the quality and correctness of the information” [48].

Solihin & Eastman [49] define the following rule categories for model checking:

- Checks for well-formedness of a building model, i.e., the syntactic properties of the digital model
- Building regulatory code checking
- Specific client requirements
- Constructability and other contractor requirements
- Safety and other rules with possible programmed corrective actions
- Warrantee approvals
- BIM data completeness for handover to the facilities management

This shows that similar, but not standardized criteria for model quality exist. “In order to bring these elements together in a construction project, it makes sense to implement a strategy for quality assurance. In this way, maximizing the quality of model content throughout the project becomes a core objective” [4].

In this thesis, methods are investigated that focus on quality of content and design. Data-technical checks according to Preidel’s definition are a secondary aspect.

2.3 Level of Detail and Level of Development

Analogous to the internationally common sequential project phases, which provide for increasing detail with increasing project progress (see [50–52]), it is also common practice to adapt the model content to the respective project phase. How detailed a model should be in the respective project phase is generally not standardized in German-speaking countries. Furthermore, different definitions are used for the level of development. The following paragraphs provide an overview of the term detail in the context of models.

"Digital building models consist of individual model elements to which alpha-numerical information can be added. The combination of geometric level of information (LOG) and alpha-numeric level of information (LOI) results in the **Modelldetaillierungsgrad** (MDG, German translation of level of development) of the model at defined performance times” [53]. The individual levels of development are described in [53] in short texts and sketches, the respective level of development is roughly outlined, but concrete content specifications for the geometric or semantic model are not included. The descriptions are predominantly oriented towards buildings.

The term “**Level of Development**” was developed by the American Institute of Architects and defined as follows: “The Level(s) of Development (LOD) describes the level of completeness to which a model element is developed” [54]. The system has been refined/interpreted by the BIMFORUM. The Level of Development is divided by the AIA into five levels (100, 200, 300, 400, 500). An additional level (LOD 350) has been introduced by the BIMFORUM, see [55]. The “Level of Development Specification” makes component-related statements about the requirements for detailing. The catalog is mainly filled with components from the fields of architecture, structural engineering, technical building equipment and civil engineering. However, there are also some placeholders for the infrastructure sector, especially for the road construction sector. Furthermore, bridge girders made of steel and concrete for roads and railway systems are described in the four detail levels, as, for example, shown in Figure 6.

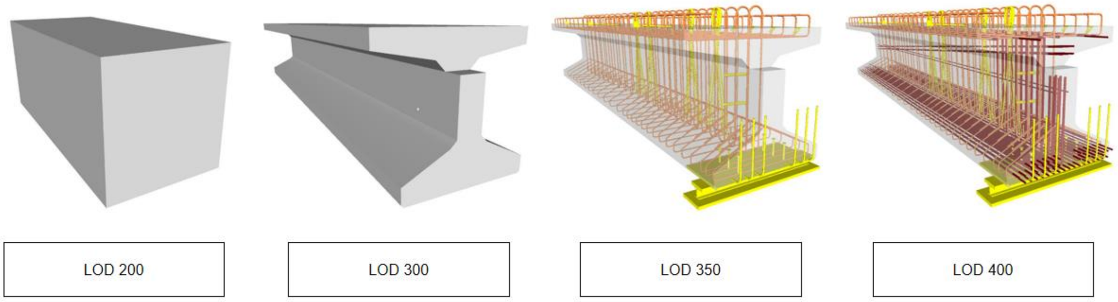


Figure 6: Level of Development of Railway Bridge Precast Structural I Girder (see [55]*)*

Mini adopts the term “**Level of Development”** and equates it with the terms “degree of completion”, “degree of maturity” or “degree of elaboration” [56]. *Mini* expands the definition of the level of development by establishing a link to the project phases according to HOAI. The assignment of the level of geometry to a project phase is based on the drawing scale of the deliverable from the 2D design approach. With regard to information content, only a few general statements are made. Here, reference is made to the project dependency.

The term **“level of definition”** originates from the United Kingdom and is understood as a blend of the terms “level of model detail” and “level of model information” [57]. A distinction is made between seven “levels of definition”, which are (1) Brief, (2) Concept, (3) Definition, (4) Design, (5) Build and Commission, (6) Handover and closeout, (7) Operation.

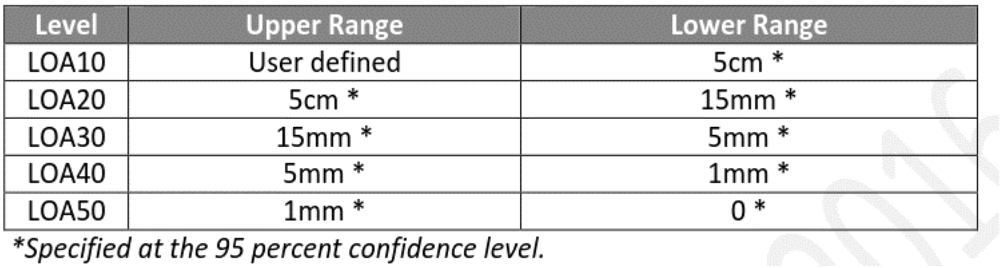
According to the “Guidelines for the Application of the BIM Methodology” by the Deutsche Bahn AG, the “**Level of Detail** (LoD) [...] defines the geometric level of detail of 3D components in the respective design phases” [9]. DB Station & Service AG equates the term Level of Detail with the requirements for geometry (Level of Geometry). The Level of Detail and Level of Information are considered separately from each other and refer to the component. DB Station & Service AG distinguishes between five different LoDs. In the published specifications, the individual components are shown with a picture and description for better user-friendliness. However, all components reach the highest accuracy level with LoD 200. LoDs 300 to 500 always correspond to LoD 200. Similarly, component-related, non-graphical information (properties) is defined and assigned to the individual Levels of Information. This provides the user with explicit specifications that can be implemented, checked and thus quality assured during project processing.

The term “**Fertigstellungsgrad”** (English: “degree of completion”) is given in ZukunftBAU [58]. “The degree of completion of the discipline-specific building models represents the required degree of detailing of the modeling and depends on the project phase and the discipline. [...] As a first approximation, the completion levels of the models are comparable to the 2D design scales in the different project phases, and therefore the expression of the models in the initial phases is still less detailed and will then develop in more detail as the design progresses.”

The term “**Level of Information Need**” is defined in EN 17412-1:2021-06, which describes concepts to define the depth of information requirements in context of BIM and different levels for sharing information throughout the lifecycle of buildings and structures. The Level of Information Need distinguishes between geometric and alphanumeric information.

The term “**Level of Accuracy”** originates from the field of surveying. Basically, it is necessary to distinguish between the terms “precision” and “accuracy” in the sense of exactness. Precision is understood as the degree of consistency in multiple measurements. Accuracy, on the other hand, describes how precisely a point or object to be recorded was actually recorded (see [59,60]). Five levels of accuracy and one upper and one lower limit for the measure of accuracy are specified as shown in Table 1.

Table 1: Levels of Accuracy [59]



Level	Upper Range	Lower Range
LOA10	User defined	5cm *
LOA20	5cm *	15mm *
LOA30	15mm *	5mm *
LOA40	5mm *	1mm *
LOA50	1mm *	0 *

**Specified at the 95 percent confidence level.*

In the following, the different definitions are compared in relation with the project phases according to HOAI. The term “level of definition” is not included, as it is not possible to draw a clear conclusion with regard to the HOAI.

Table 2: Comparison of different definitions in relation to the project phases according to HOAI

Project phase		MDG	FG	LoD * (BIMFORUM)	LoD (Mini)	LoD (DB AG)
1	Basic evaluation	010	-	-	100	100/ 200
2	Preliminary design	100	100	100	200	min. 100
3	Basic design	200	200	200	300	200
4	Approval planning	210	300	-	300	200
5	Detailed design	300	400	300	400	300
6	Tender phase	310	-	-	-	-
7	Support of tender phase	320	-	-	-	-
8	Site management Factory design	400	500	400	450	400
8	Site management As-built model	500	-	500	500	500
9	Supervision of facility for warranty purposes	510	-	-	-	-
-	Facility Management	600	-	-	-	-

* based on comparisons given in [53]

There are various terms and definitions of the level of development of building models, both internationally and on the German market, and the multitude of different definitions can lead to confusion. The acronym LoD in particular is used several ways in practice. What all definitions have in common is that a distinction must be made between the geometric model and the semantic model when describing detailing needs or expectations. Exact definitions of geometric accuracy are only given by [9,55].

2.4 Data exchange formats

“Today, the main obstacles are to be seen in the use of different data formats and the non-standardized level of detail of information in models. But already today automatic model checking can be applied very effectively and objectively if the specific use cases and corresponding modeling requirements are carefully defined upfront” [61].

In addition to the standardized description of components and structures, it is necessary to exchange data and models between different software as part of the BIM methodology. Various data schemas are available for this purpose, which are examined in more detail below.

Borrmann et al. [62] basically distinguishes between explicit and implicit methods for describing 3D objects. In the explicit method, the volumetric object is described as a surface. “The implicit method, by contrast, is a procedural approach that describes the history of the creation of the modeled body” [62]. The parametric modeling of parts is used in modern CAD applications to reuse them, but at the same time to be able to adapt them, see [62,63]. The distinction between explicit and implicit geometry description is essential for the consideration of data exchange formats. An example of an explicitly described 3D model is shown in Figure 7.

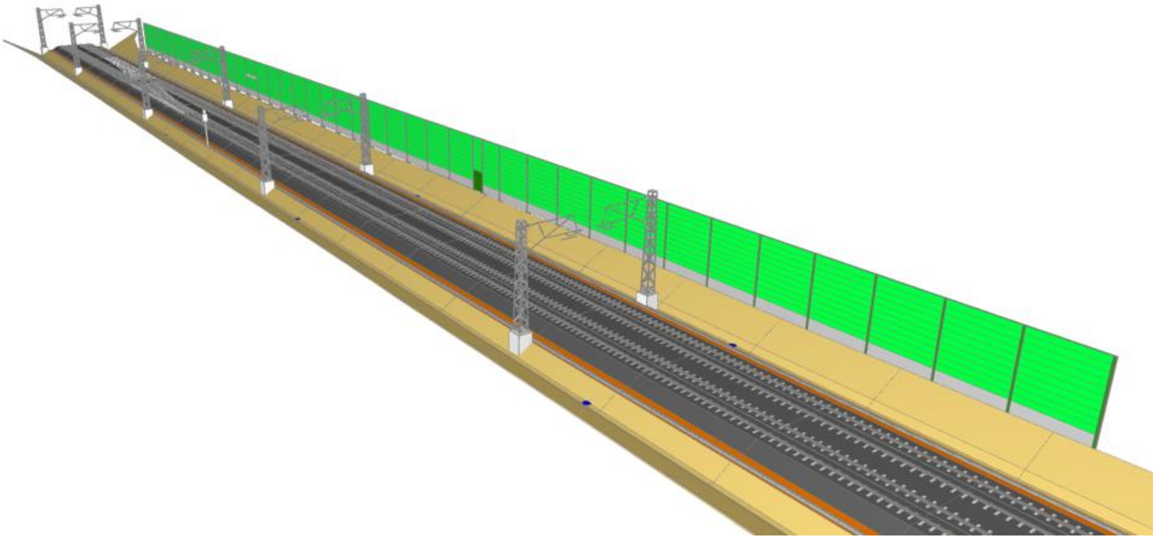
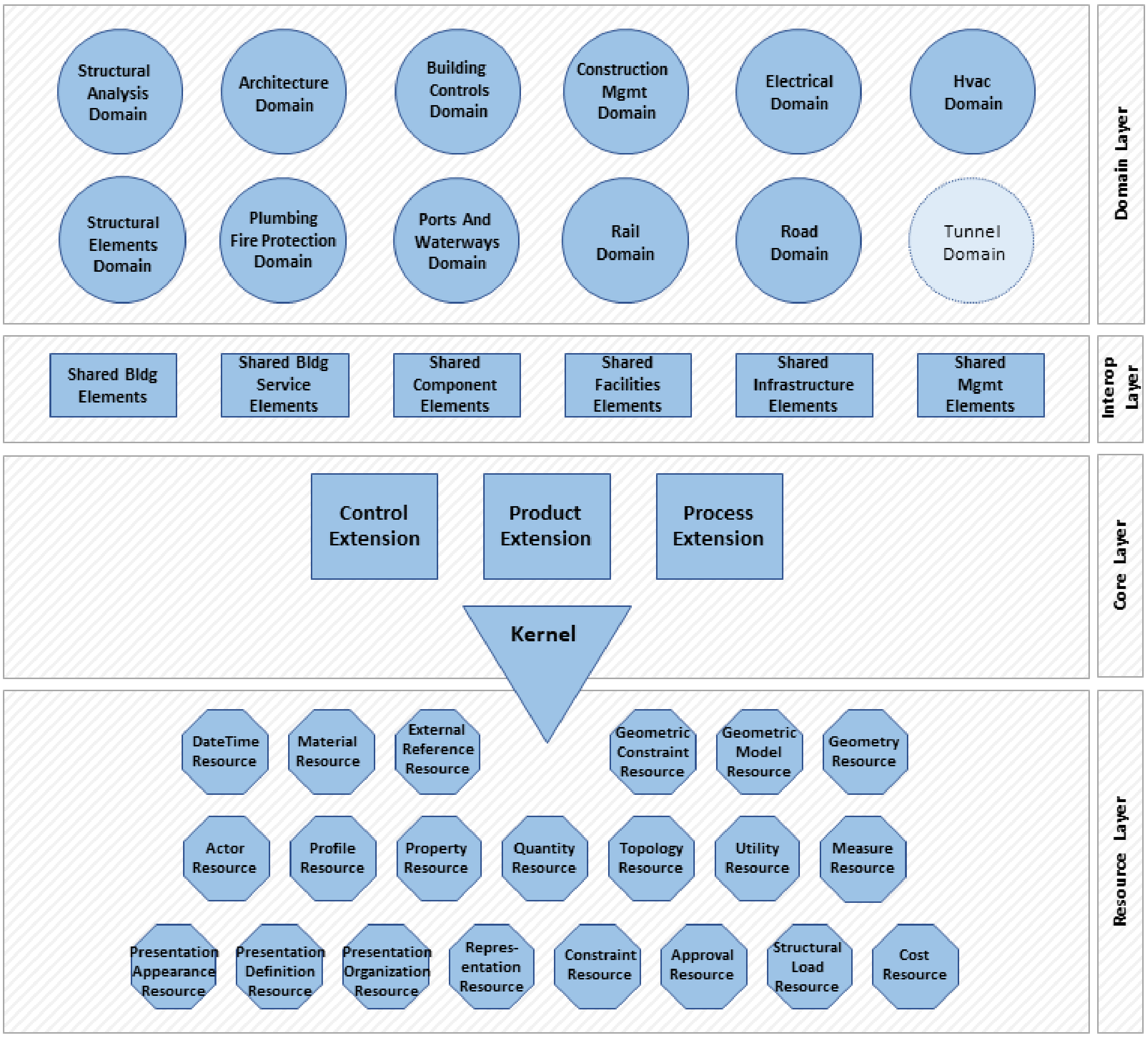


Figure 7: Explicit description of a 3D-model

“Both methods are used in BIM software and in the corresponding data exchange formats, and both are part of the IFC specification” [62]. IFC is the acronym for Industry Foundation Classes. ”IFC is a standardized, digital description of the built asset industry. It is an open, international standard (ISO 16739-1:2018) and promotes vendor-neutral, or agnostic, and usable capabilities across a wide range of hardware devices, software platforms, and interfaces for many different use cases” [64].

IFC is developed within the framework of buildingSMART International. Currently, version 4x1 is the published version and contains the structures for terrain models as well as horizontal and vertical alignment that are relevant for infrastructure. Version 4x2, which contains extensions for bridge models, is currently still in draft status. Development on both the IfcRoad and IfcRail projects has reached so-called candidate status with version 4x3 [65]. The IFC data model is divided into four concept layers (resource, core, interoperability, domain). In version 4x3, separate domain layers are provided for the rail and road domains (see Figure 8). In addition to the methods for describing geometry and semantics, the IFC schema also contains concepts for describing schedules, quantities, and costs. This is illustrated by the resource layers “DateTime”, “Quantity”, and “Cost” (see also [66]).



Industry Foundation Classes version 4.3.x Architecture overview
© buildingSMART International Ltd.

Figure 8: Architecture of IFC version 4x3, original from [65]

Next to IFC the exchange format CPIxml (construction process integration markup language) exists, which is a proprietary interface by the company RIB AG. It was developed to make model information from RIB's CAD systems available to the iTWO quantity takeoff and cost calculation tool. "Analogous to the IFC standard, geometric objects were defined for road and civil engineering projects, among others, which enables the exchange of planning models in 3D" [67]. The implementation of the interface on a practical example is described in [68] and the interface is now offered in numerous software products.

In context of roads, there is another exchange format available in the German-speaking road construction sector which is called "object catalog for roads and traffic" (OKSTRA). It represents a comprehensive standard for "all areas from road design and as-built documentation to the recording of traffic data" [69]. OKSTRA is used to describe and exchange a wide range of data, e.g., design data for roads, cadastral data, accident data or road network and road condition data (list not complete). In the field of road design, OKSTRA can be used to exchange horizontal and vertical alignment as well as cross-section data, for example. Digital terrain models and classical 2D data such as points, lines, and surfaces can also be exchanged. Figure 9 shows an example of road design data in the OKSTRA schema. The data are transmitted as implicit, parametric geometry descriptions, divided into site plan, cross section and longitudinal section. According to a proposed amendment by *Tulke*, the OKSTRA scheme will in future be developed from a two-dimensional to a three-dimensional scheme for the description of volumetric components, see [70].

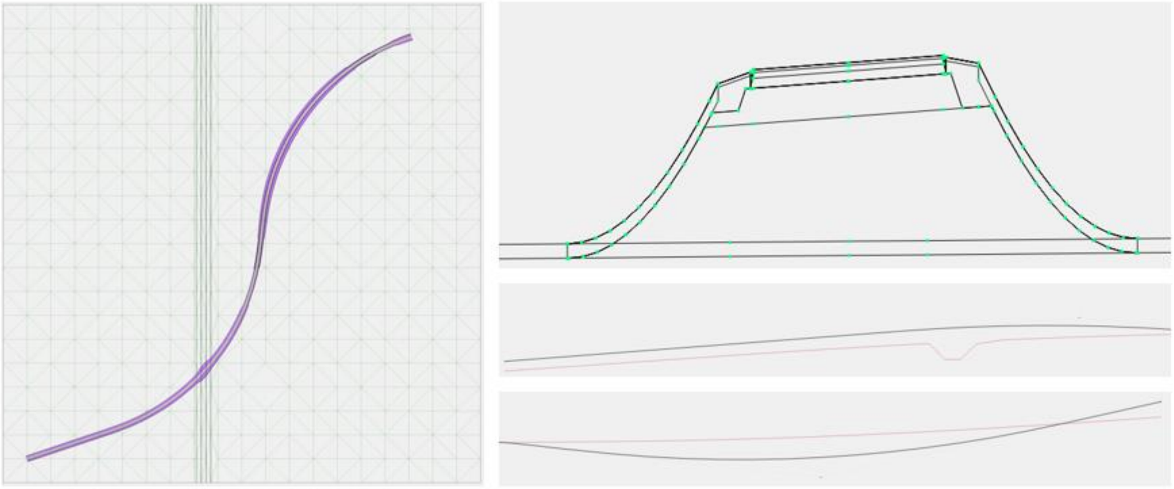


Figure 9: Representation of OKSTRA test data, map view, cross section, longitudinal section, data available on www.okstra.de

The data models and data exchange formats available in the railway sector are considered by *Wunsch et al.* [71]. In the following, the formats IDM^{VU} (Infrastructure Data Management) and railML are described in more detail.

IDM^{VU} was developed based on the OKSTRA standard. The class library of OKSTRA was extended accordingly by the requirements of IDM^{VU}. "The IDM^{VU} data model describes all infrastructure objects, inclusive of their attributes and data types, as well as their relation to one another in the field of infrastructure data management of transportation companies active in the public transport and rail freight sector. Put simply, all ‘real’ components that are mentioned in e.g. as-completed documents or design plans are called infrastructure objects. It is a uniform, system-neutral basic data structure for the qualified documentation and provision of data within the scope of the life cycle process (planning, building, operation, maintenance and disposal) of the infrastructure of transportation companies.” [72]. The data model is distinguished into several TOP-levels:

> Network model, track system, power supply, control and safety technology, stations, structures, cables and lines, telecommunications, real estate, depots, emergency facilities, general objects, condition data, operational data, commercial data.

IDM^{VU} enables the data exchange of the implicit data model and the objects contained therein and arranged relative to each other between different modeling software, analogous to OKSTRA. Figure 10 schematically illustrates the possibilities for exchanging elements of the track body. It is possible to describe railway specific objects like rails, sleeper and the cross section type.

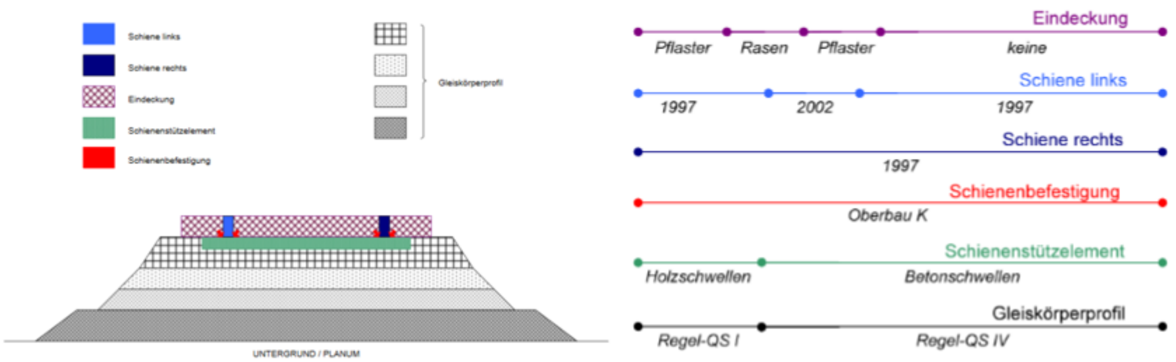


Figure 10: Graphical representation of IDMVU, based on [72]

Practical experience shows that IDM^{VU} is not used in Deutsche Bahn projects. Typically the data exchange is based on IFC for 3D models. In context of alignment several formats are used, like Verm.ESN or landXML.

In contrast to the detailed description of infrastructure elements like rails or sleepers, there are high level exchange formats available which focus on railway operations. “Many different topological infrastructure data models and interfaces have been created over the years, either to fulfil national railway needs or to support EU directives” [73]. “RailTopoModel is a logical infrastructure data model aimed at international standardization for the description of railway systems (network topology, infrastructure and the whole lifecycle of operations). It is the result of continuous collaborative work of the UIC RailTopoModel Expert Group, in which various infrastructure Managers from all around Europe take part” [74]. During the development of the RailTopoModel, various national and European models for the description of railway operating data were examined. The investigation has shown that 95% of the respective model features are compatible with each other, which is due to the similarity of railway infrastructure in the various nations. It was also found that different levels of detail have to be taken into account (see Figure 11). Four levels of detail are distinguished: microscopic, mesoscopic, macroscopic and corridor. Furthermore, it becomes clear that the microscopic level is also structured as a node-edge model, which can be sufficient for the operation of railway infrastructure (see also [75]). The detailed description of structures, as is possible with IFC, OKSTRA, IDMVU, cannot be defined by this. RailML is a data exchange format for railway infrastructure data that has been closely related to RailTopoModel since version 3 [76]. It is structured as an independent markup language. With railML, timetable, vehicle, infrastructure data and interlocking logic can be transmitted in a standardized way [75,77]. railML aims to standardize the exchange of data between different software in the area of railway operations planning. An application example for the use of railML is shown by *Rahmig*. This refers to the exchange of schematic track plans on a microscopic level [78].

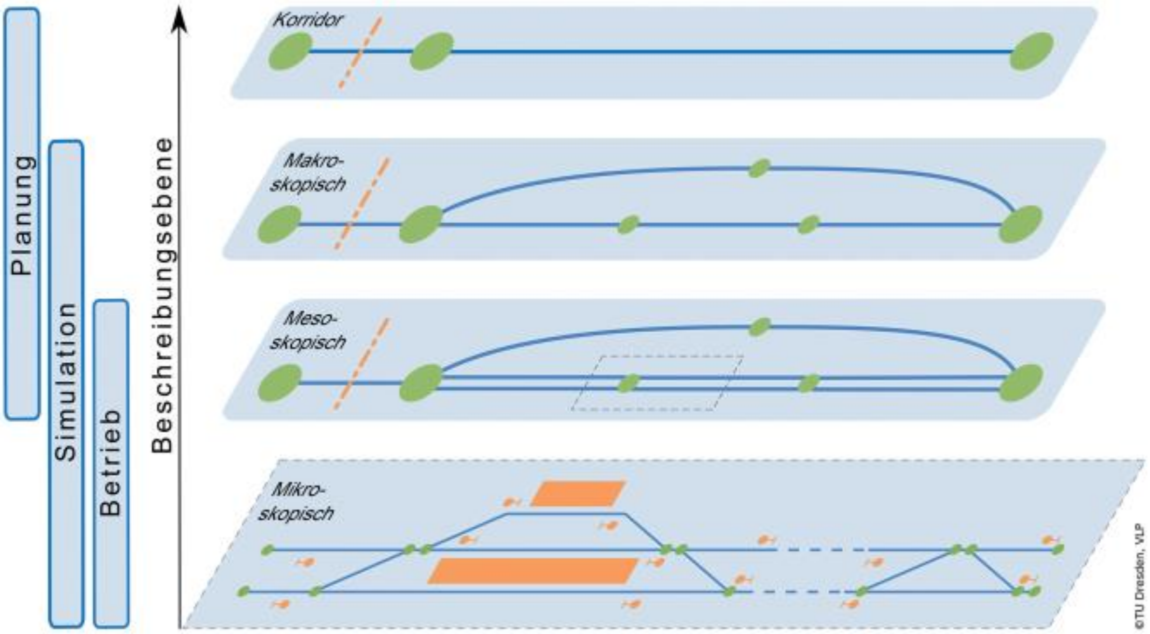


Figure 11: Topology of railway infrastructure, original from [71]

Control and safety aspects are described in the microscopic level of topological model shown in Figure 11. The link between operational data models and BIM oriented data exchange formats is developed with the domain specific data model “PlanPro”. The research project was initiated by DB Netz AG with the aim of digitizing the planning process in the area of control and safety technology [79]. PlanPro is an XML-based data model [79] that can be written or read by various specialized applications, such as ProSig and BEST [80]. In addition to a standardized data exchange format for control and safety technology, PlanPro also represents a system for rule-based checks. For this purpose, generally valid test rules are formulated by means of Schematron [81]. With the help of appropriate translation mechanisms (XML parsers), the data models stored in the XML schema are checked for compliance with the Schematron rules. As a result, result reports are generated automatically [82,83].

In addition to the software-neutral data formats presented, there are numerous proprietary formats that are not listed here.

2.5 Compliance checking of models

In order to be able to determine whether a building model meets the requirements, the model must be checked accordingly. In context of BIM projects this process is known as model checking. “The term Model checking or Code Checking refers to the automated examination of the model on the basis of rules” [61]. For every check and the associated process, regardless of the BIM method, the central question “Who checks when, what and how?” must be considered.

According to ISO 9000:2015 an audit is a “systematic, independent and documented process for obtaining objective evidence and evaluating it objectively to determine the extent to which the audit criteria are fulfilled. […] The fundamental elements of an audit include the determination of the conformity of an object according to a procedure carried out by personnel who are not responsible for the object audited” [47]. In the context of models, auditing is also referred to as checking. In the context of the dissertation, the term “checking” is used. This means that model checking must in principle be carried out by a person who did not create the model.

2.5.1 Necessity of checks

According to ISO 19650-2, a common data environment with integrated workflows and status information should be used for project implementation. The status of information can be defined as “work in progress”, “shared” or “published” [84]. Furthermore, ISO 19650-2 requires a quality check before each status change of information, e.g., from work-in-progress to shared.

The creation of a building model can have different objectives. This can be seen in the various use cases that are pursued in projects. An excerpt for possible BIM-based use cases is given below on the basis of [85]:

- Design Coordination
- Clash Detection
- 4D construction process animation
- Structural engineering
- Energy analysis
- Code Compliance Checking
- Quantity Take-Off
- 3D printing

From this it can be deduced that different tests are necessary depending on the objective pursued. “The geometry, semantics, and linked information of the model are assessed in order to check for compliance with certain planning principles, customer requirements, or building regulations (codes)” [61].

According to ISO 19650-2, the requirements for models should already be defined in the BIM execution plan. It represents the central document for BIM project and contains requirements for models.

Using the example of the City of Vienna, Fiedler [86] states that both the submission of plans and the subsequent building permit procedure are still paper-based and proposes modernizations for this process. The scenarios initially envisage the digital, paper-based handling of the procedure, followed by a computer-based partial or overall review of the digital models.

Eastman et al. [87] divide the design compliance checking process into four phases (see Figure 12), which have been adopted in the dissertation.

(1) Rule interpretation and its logical representation
(2) Building model preparation
(3) Rule execution
(4) Rule check reporting

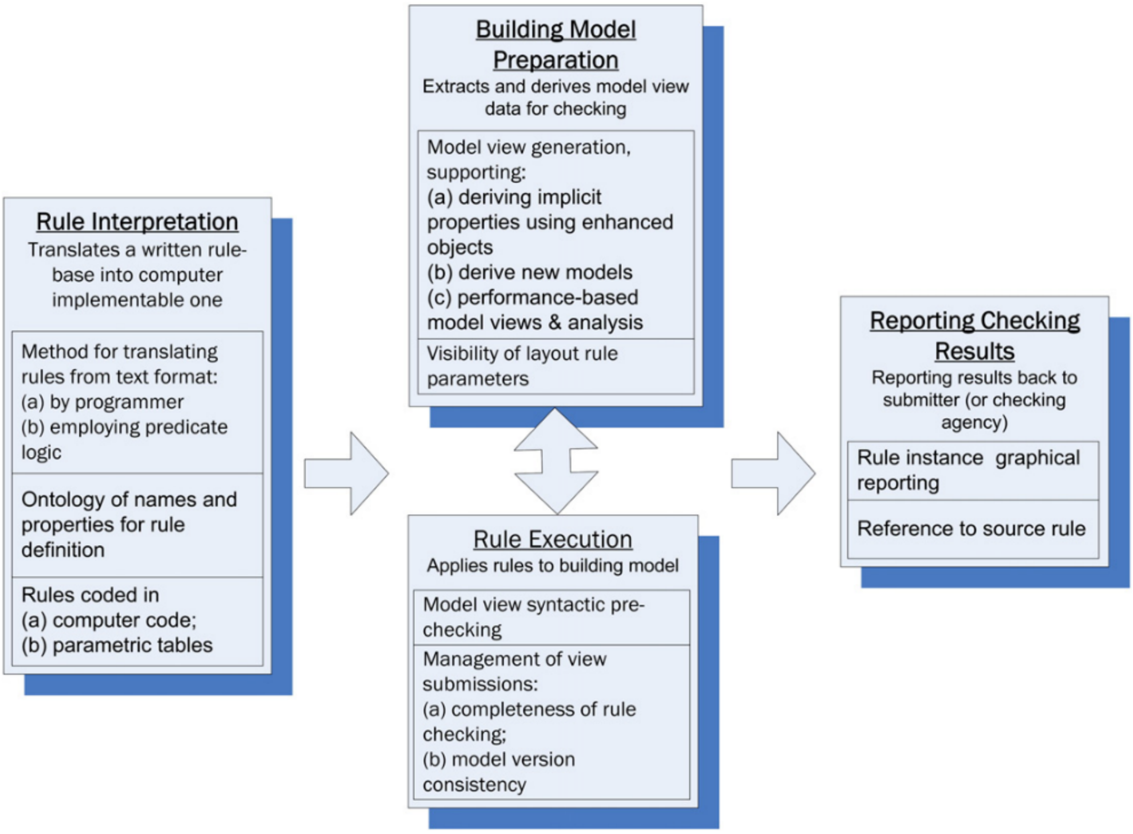


Figure 12: The four phases of a rule checking system by Eastman et al. [87]

2.5.2 Rule interpretation and representation

The interpretation and logical, machine-readable representation of rules is an essential preparatory task for model checking. “As a central element, the rules to be checked are to be phrased in such a manner that they can be interpreted by computers” [61]. Examples of logic-based approaches are propositional logic, predicate logic, and deontic logic, where propositional logic is the basis for other logical approaches (see [4,87,88]).

In addition to logic-based approaches to representing rules, there are also language-based approaches, such as the Extensible Markup Language (XML) [89]. These are characterized by the flexibility to be exchanged between different platforms, regardless of the programming language in which they were developed [87]. Another language-based approach is represented by RASE (Requirement, Applicability, Select, Exception). “RASE is a semantic based concept for transforming normative documents into a single well-defined rule which can be implemented into BIM / IFC based model checking software” [90]. The application of the method is limited to text-based rules and is a predominantly manual process, see [4].

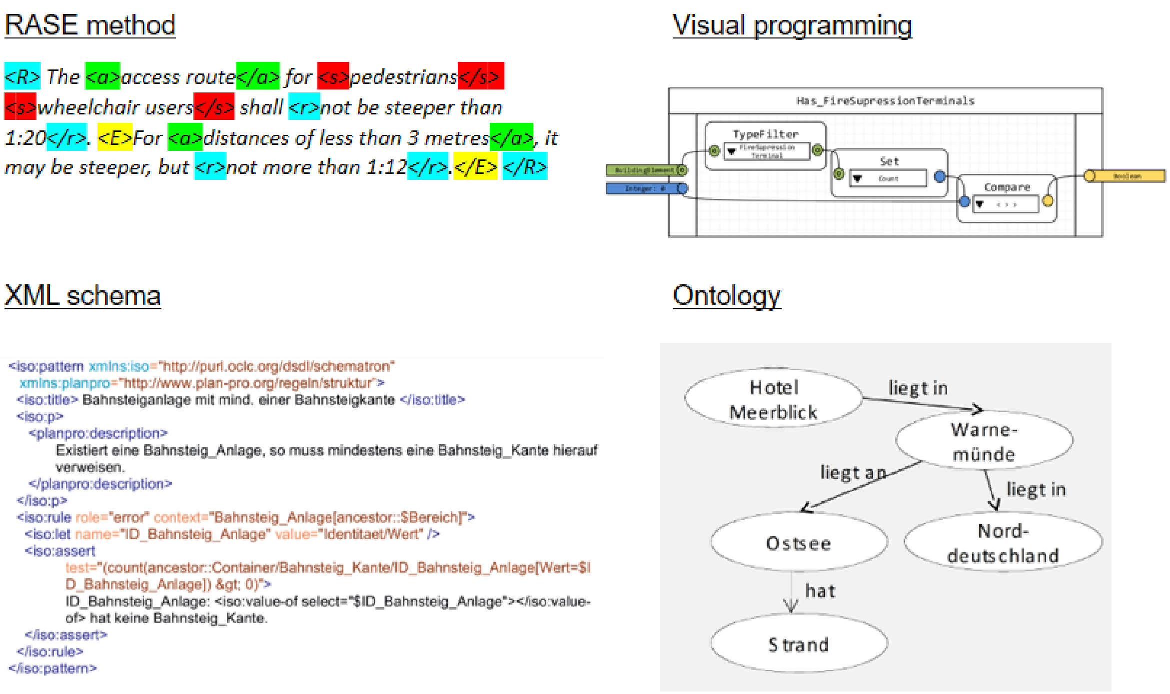
The automated translation of human-readable into machine-readable language is part of the research area "Natural Language Processing". “While these methods have many benefits in terms of ease of use, there is usually far too much leniency in the written language, which makes it impossible to process automatically and accurately; as a result, these methods are fundamentally limited in their capacity to capture” [91].

The translation and representation of human-readable language into machine-readable language can also be done with the help of ontologies. An ontology is the “formal definition of concepts and their relationships as the basis for a common understanding. […] Ontologies should improve or enable communication between computer applications, between computer applications and humans, but also between humans.” [92]. Studies related to BIM and ontologies can be found, among others, in [93–95]. According to *Sydora & Stroulia* ontologies are also increasingly used for model checking.

Especially for users without in-depth programming knowledge, it can be difficult to understand the program logic. This in turn can lead to a decrease in trust in test routines, see [4]. Applications whose program logic is not visible are also called black boxes [4,91,96,97]. The opposite are white-box solutions [91,98]. Visual programming languages pursue the approach of increasing the transparency of the program code through graphical elements and are therefore to be classified as white-box solutions. “A VPL has a high degree of user-friendliness and typically lowers the hurdle for normal end-users to enter the field of programming. Thanks to its visual representation and means for intuitive interaction, VPL programs are simpler to create and to understand than its textual counterpart” [99]. Analogous to text-based imperative programming languages, visual programming is used to explicitly define the sequence of instructions.

Declarative languages like SQL [91], SPARQL [100,101], QL4BIM [99,102] represent another approach to represent rules in a machine-readable way. These are also referred to as query languages.

Figure 13 shows selected representation methods as examples.



RASE method

<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>

Visual programming

XML schema

```
<iso:pattern xmlns:iso="http://purl.oclc.org/dsdl/schematron"
  xmlns:planpro="http://www.plan-pro.org/regeln/struktur">
  <iso:title> Bahnsteiganlage mit mind. einer Bahnsteigkante </iso:title>
  <iso:p>
    <planpro:description>
        Existiert eine Bahnsteig_Anlage, so muss mindestens eine Bahnsteig_Kante hierauf
        verweisen.
    </planpro:description>
  </iso:p>
  <iso:rule role="error" context="Bahnsteig_Anlage[ancestor::$Bereich]">
   <iso:let name="ID_Bahnsteig_Anlage" value="Identitaet/Wert" />
   <iso:assert
          test="(count(ancestor::Container/Bahnsteig_Kante/ID_Bahnsteig_Anlage[Wert=$I
          D_Bahnsteig_Anlage]) &gt; 0)">
          ID_Bahnsteig_Anlage: <iso:value-of select="$ID_Bahnsteig_Anlage"></iso:value-
          of> hat keine Bahnsteig_Kante.
   </iso:assert>
  </iso:rule>
</iso:pattern>
```

Ontology

Figure 13: Different representation of rules; RASE [90]*, visual programming* [4]*, XML-schema* [82] *and ontology* [92]

2.5.3 Technical solutions

Eastman et al. define “Automated Rule Checking” as software that does not change the building model, but evaluates the objects placed within it, their relations to each other, and the information they contain [87]. There are several technologies available to set up such rule checking tools:

- as an extension for authoring software, enabling the designer to check the model at any time,
- as stand-alone software,
- as a web-based solution.

The steps for preparing building models and rule checking depend on the software solution used. Various software solutions have been developed and tested in various projects worldwide which are predominantly focused on buildings. In Singapore, for example, the CORENET project was set up as early as 1995. According to *Eastman et al*., CORENET goes through a three-step process to check building models:

(1) Checking rules with current IFC information,
(2) Checking rules with property set extensions to IFC, and
(3) Checking rules with derived information from IFC.

Further approaches are described by *Eastman et al.* [87], which are only mentioned here as keywords:

- SMARTcodes (North America)
- Design Check (Australia)
- General services administration design rule checking (USA)

Amor & Dimyadi cite some more developments [103]:

- Express Data Manager (Norway)
- Residential Compliance – ResCheck und Commercial Compliance – ComCheck (USA)
- Design++
- Solibri Model Checker (Finland)

According to *Amor & Dimyadi*, all the used systems are based on checking mechanisms that are hard-coded in the source code. A detailed description of the systems and developments can also be found in [4].

The LicA system and the associated LiCAD user interface were developed specifically to meet the requirements of the Portuguese regulations for domestic water supply (see Figure 14).

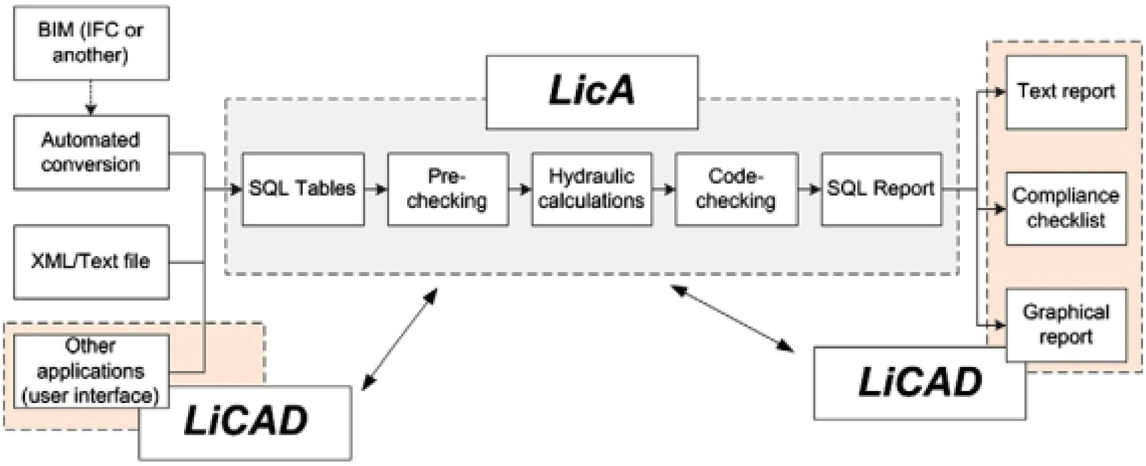


Figure 14: Workflow of LicA und LiCAD [104]

Here, both the geometric model and the rules are converted into an SQL database. Subsequently, preliminary checks, hydraulic calculations and the actual rule check are performed against the preceding calculations. The results are output as a report in various presentation forms.

Preidel develops an approach to check building information models using a visual programming language called VCCL [4]. *Dimyadi et al.* use a graphical notation called BPMN to develop processes for the automated checking of building models [105–107]. BPMN is standardized in ISO/IEC 19510.

While the approaches described so far are limited to the checking of a 3D model, *Daum & Borrmann* are developing the first approaches to also include 4D models in the checking process [108]. For this purpose, the IFC data structure is used, which makes it possible to define so-called IfcTask or IfcTaskTime information in addition to the geometric representation. With the help of the specially developed query language Live LINQ, building models can be queried and checked for logic over the construction period. The query language, which was continued under the name QL4BIM, was also used by *Daum & Braun* to automatically generate schedules from IFC data models [109].

2.5.4 Reporting

Documentation of check results is the final step of the four step process. Eastman et al. [87] show how reports can be presented using various examples. These range from simple text-based documents in Word format or as PDF, through HTML or XML, to graphical feedback on the model. The BIM Collaboration Format standardized by buildingSMART International is available to support model-based communication. It “allows different BIM applications to communicate model-based issues with each other by leveraging IFC models that have been previously shared among project collaborators” [110].

According to Preidel [4], a check can basically be concluded with the results ‘passed’, ‘failed’, ‘warning’ or ‘unknown’.

2.6 Knowledge Based Engineering

The aim of the considerations presented in the previous sections is to check existing models for their correctness or compliance. This approach represents a downstream quality check, which in turn means that detected problems, errors, clashes, etc. must be corrected in an iterative process. Subsequently, the newly generated model information is repeatedly checked. Although automatisms can be used to speed up the checking process, this does not affect the iterative process for post-processing the models. The downstream checking of models merely detects the symptoms of a faulty model creation process.

One goal of the work is the early, preferably automated minimization of errors. For this purpose, it is necessary to include the model creation process in the quality assurance strategy. Knowledge Based Engineering is a recognized synonym for automated model creation and is defined by *Stokes* as follows [111]: “The use of advanced software techniques to capture and re-use product and process knowledge in an integrated way.” At present, however, the construction industry lacks suitable methods for digitally mapping existing knowledge in such a way that it is machine-readable, so that it can be evaluated and reused. This is where knowledge-based engineering comes in. “Knowledge-based engineering (KBE) stands at the intersection of diverse fundamental disciplines, such as artificial intelligence (AI), computer aided design (CAD) and computer programming” [112]. KBE systems can be applied to achieve a variety of goals, the most important of which is to increase efficiency by using automated systems for routine tasks, as described, for example, in [111].

According to *Abulawi* [113], the following motives exist for the application of automated design processes: (1) process acceleration, (2) quality management, (3) knowledge management, (4) ergonomics, (5) increased efficiency, (6) increased safety or increased reliability of decisions.

According to *Singer & Borrmann* [114], the use of KBE applications is useful in the following situations:

- a high degree of similarity between product versions exists,
- a large number of customization options exist,
- a large number of design processes exist,
- many competing or conflicting requirements exist,
- knowledge from many different sources is available and has to be considered,
- the design is affected by many disciplines,
- many iterations are performed towards the final design,
- a high amount of resembling time-consuming but primitive design tasks exist,
- many decisions during the design process have to be made.

“Compared to traditional CAD usage with approximately 80% repetitive tasks, this percentage can be significantly decreased by applying KBE. This reduces the overall product development time and allows the engineer to focus on the true technical challenges” [114].

As far back as 1988 *Adeli & Balasubramanyam* developed a KBE application for the design of bridge trusses and noted: “The entire design process involves many decisions to be made by the designer based on rules of thumb, heuristics, and previous experience. […] Intuition, judgment, and previous experience have to be used to select the right values of the design parameters. Further, since design is an open-ended problem, that is, in general there are numerous design alternatives satisfying all the specified constraints, the selection of the ‘best’ design becomes an extremely challenging problem” [115]. *Chassiakos et al.* developed a maintenance planning KBE-system of highway concrete bridges [116]. Singer also applies KBE methods in the field of bridge design [117]. According to *Stjepandić et al.* [118], practical examples of KBE applications outside the building industry exist in the following fields Automotive, Aerospace, Space, Shipbuilding, Electrical Engineering, Dentistry and Manufacturing.

There are many studies related to the optimization of road alignment. The investigations are based on different optimization algorithms and methods, e.g. geographic information systems (GIS) [119], intersections [120] or bridges and tunnels [121] can be considered during path finding. *Zhao et al.* combine the use of BIM, GIS, and optimization methods for road design. [122]. More recent studies such as those by *Li et al.* [123], *Pu et al.* [124,125], *Song et al.* [126,127] adapt the considerations to railway alignments. Even if this topic is not explicitly mentioned in connection with KBE, there are some similarities. For example, design parameters are used as input variables to automatically obtain different solutions for a problem based on algorithms.

Based on a literature review, *Verhagen et al.* [128] formulate five shortcomings in the development of KBE applications:

(1) Case-based, ad hoc development of KBE applications: […] instead of using a structural framework or methodology to develop KBE applications, it seems that developers identify a problem and improvise a KBE solution based on a custom development process.
(2) A tendency toward development of ‘black-box’ applications: There is no explication of formulas and the actual meaning and context of the captured knowledge, let alone provisions for capturing design intent.
(3) A lack of knowledge re-use: Case-based black-box KBE applications do not particularly invite knowledge re-use.
(4) A failure to include a quantitative assessment of KBE costs and benefit: Case studies […] do not mention the resulting time or cost advantages associated with KBE adoption, let alone the more sensitive information about KBE development cost.
(5) A lack of a (quantitative) framework to identify and justify KBE development: […] no solid framework or method using both qualitative and quantitative aspects is available to determine whether a design task, product or process is suitable to develop a KBE application.

There are numerous approaches to commercial KBE applications, see Figure 15. According to *La Rocca* only GDL and AML are true KBE applications, “[…] all the others are basically KBE-augmented CAD systems, where a true KBE language (e.g., Siemens NX Knowledge Fusion) or some sort of KBE capabilities (e.g., Dassault Systemes CATIA V5 Knowledge-ware) have been integrated to augment the core CAD functionality" [112]. Other automation solutions not shown are being developed by Lino [129], Tacton [130], or Autodesk for their product family [131]. The solutions focus predominantly on areas of mechanical engineering or product design. Autodesk also offers a product-specific solution in the area of civil engineering by linking Design Automation to Revit.

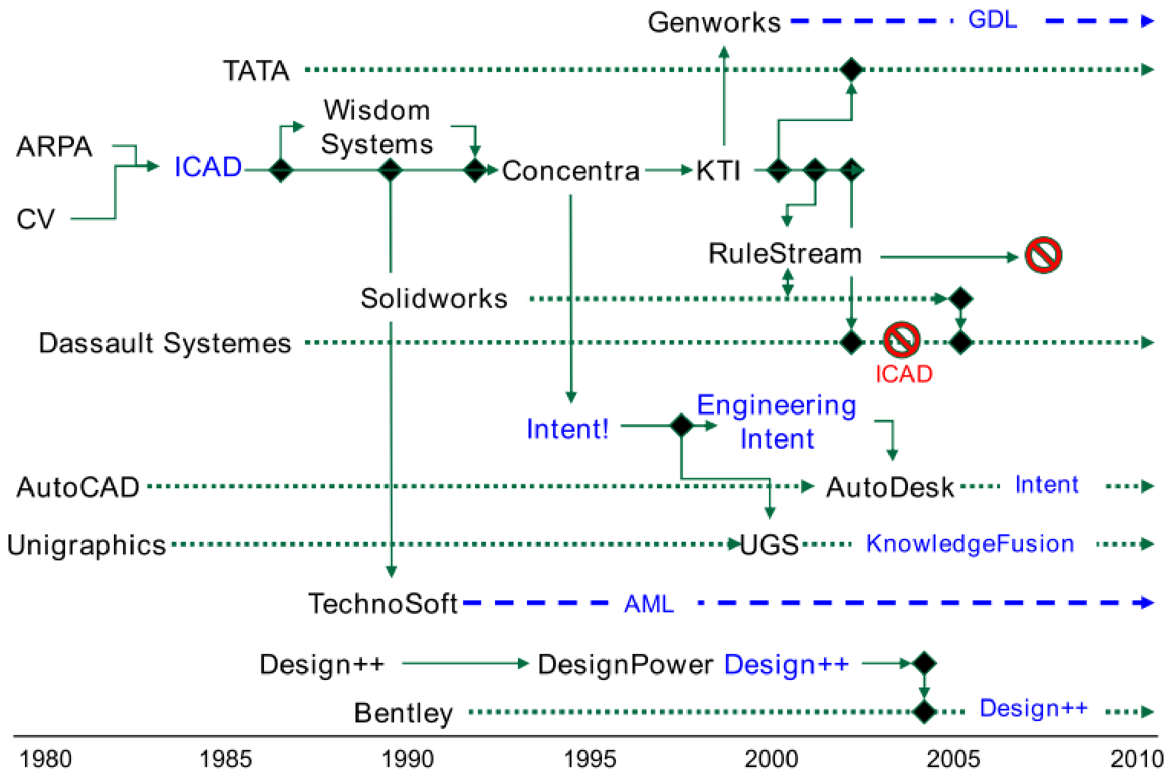


Figure 15: Timeline of KBE-systems, original from McGoey [132]

The definition of requirements for a KBE application is related to the definition of KBE itself. *Verhagen et al.* states that "older definitions are more narrow and technology-oriented; for instance, the notion of KBE as a combination of CAD and AI techniques. More recent definitions of KBE are wider and less restrictive; they for instance do not contain the geometry focus that often seems to constrain KBE applicability. Instead, newer definitions focus on the automation of repetitive engineering tasks while capturing, retaining and re-using associated knowledge" [128].

2.7 Summary

Quality assurance in construction planning and especially in the BIM context is a multi-layered topic. A distinction must be made between downstream quality assurance and quality assurance during model creation.

Eastman presents a four-step model checking process that has been adopted in numerous studies. This process is also adapted for the present dissertation. It becomes clear that models have to be tested depending on the use case to be implemented and that there is no single criterion for quality. In the literature there are mainly examples of model checking in the context of building design. The literature shows that the developed approaches are focused on single aspects of the possible range of model checking. Cross-domain approaches are not found in the literature. The criteria referred to as data quality by *Preidel* are not considered in this thesis. The other approaches are adapted as best as possible to the design of railway structures within the scope of the dissertation.

In particular, various methods exist for the representation of rules. Visual programming languages have a high degree of transparency due to the graphical representation of the code, which increases user-friendliness. The notation BPMN, standardized in ISO/IEC 19510, covers this advantage, but is developed for the purpose of business process modeling. BPMN is not a notation designed specifically for the design industry. By combining it with DMN, sophisticated process and decision logics can be developed and represented in a user-friendly way. There are only a few studies from building design for model checking using BPMN. DMN has not been used for model checking to date. The combination of BPMN and DMN for checking railway models therefore represents a new approach.

There are several reasons for the development and use of KBE applications. The reasons relevant to this dissertation are: Process acceleration, quality management and knowledge management. The development of a KBE application is understood as a quality assurance measure accompanying 3D model creation. However, there are also negative phenomena in the development of such applications, which are to be excluded in the present dissertation: ad hoc development, black-box solutions, lack of knowledge re-use, missing assessment of costs and benefit. Analogous to the downstream model checking, the graphical notation BPMN is used in combination with DMN. This represents a novel approach to the development of KBE applications.

Chapters 3 to 5 present a more detailed review of the state of the art.

3 Model-based quality assurance in railway infrastructure planning

Previously published as: Marco Häußler, André Borrmann. *Model-based quality assurance in railway infrastructure planning,* Automation in Construction, Volume 109, 2020, 102971, ISSN 0926-5805 https://doi.org/10.1016/j.autcon.2019.102971.

Abstract

A primary motive for adopting the methodology Building Information Modeling (BIM) in planning processes is to improve planning accuracy, cost security, and in turn quality. Up to now, however, a generally applicable, standardized means of validating design quality has been lacking. To address this shortcoming, this article presents 14 quality parameters in the domains of clash detection, semantics as well as quantities and costs that apply to the field of infrastructure planning. The sets of rules outlined in the article are adaptable and extendable in order to respond flexibly to different model structures.

The investigation focuses on how important and recurring tests can be carried out automatically and how to make the results analyzable in a transparent and standardized manner. The proposed concept thoroughly extends well-known methods such as attribute testing and clash detection analysis of the 3D model. Doing so, the paper presents a set of novel methods for quality assurance, including 4D clash detection and checks for semantic-geometric coherence. The paper discusses in detail: the influence of modeling errors on clash detection, the difference of 3D and 4D clashes, formal methods for checking the correct linkage between 3D BIM and the bill of quantities, formal approaches for checking the semantic-geometric coherence of BIM objects.

The quality assurance concept presented in the article concludes with a standardized evaluation for the individual quality criteria using a school grading system, traffic light, and percentage scale. Finally, the concept is applied in a comprehensive case study on a large-scale infrastructure project and the results of the formal quality assessment are presented.

3.1 Introduction

The construction industry promotes and develops modern and efficient technologies in an effort to respond to the increasing complexity of construction projects. Of these, the digitalization of the construction process using Building Information Modeling is one of the most well-known. The Reform Commission for Major Projects (Reformkommission Großprojekte) set up by the German Federal Ministry of Transport and Digital Infrastructure (Bundesministerium für Verkehr und digitale Infrastruktur) recommends that building owners should “make greater use of digital methods such as Building Information Modeling (BIM) in all phases of the project process” [3]. With the introduction of a staged implementation plan for digitally-mediated planning and construction (the “Stufenplan Digitales Planen und Bauen”) the German Federal Ministry of Transport and Digital Infrastructure has committed itself to the digitalization of projects within its area of responsibility in three successive steps:

- Set-up phase (2015 – 2017)
- Extended pilot phase (2017 – 2020)
- Broad implementation for all new projects (starting in 2020)

According to [1], “Design and coordination with 2D CAD systems is error-prone, labor intensive and relies on long cycle – times. BIM addresses these problems […]. The benefits of BIM for subcontractors and fabricators include: ….] reduced cycle-times for detailed design and production; elimination of almost all design coordination errors; lower engineering and detailing costs […].” Bryde et al. [133] compares 35 projects and identifies various criteria that aim to demonstrate the influence of BIM on the course of the project. The study revealed that the use of BIM was of particular benefit for the criteria of project costs and positively influenced the criteria of time, communication, coordination, and quality. The influence of the BIM methodology on project costs and project duration is the focus of various studies [134–136]. Each compare different projects with one another and find that in all the examined projects, the BIM methodology has a positive influence on project costs and project duration. As they employ different methods, the results are not directly comparable. Berg [137] considers five Norwegian projects and identifies the reasons why the actual construction deviated from the original plans, and how this impacted on construction costs. The results showed that the rate of contract deviations due to incorrect or insufficient basic information ranged from 2 to 26% depending on the project. In 23 to 48% of the cases evaluated, planning errors were identified as the reason for contract deviations during construction. The projects that employed model-based planning tended to cause fewer necessary deviations from the contract: the impact of subsequent amendments on the construction costs lay between 4 and 10% of the original cost estimate for model-based planning compared with approx. 19% for conventional planning. While these values are of limited general validity due to the small number of projects compared, they do provide a good indication of the benefits of BIM for construction projects – a finding that is also borne out by other publications in the field.

One of the most important clients driving adoption of the BIM methodology for infrastructure projects in the German construction sector is currently Deutsche Bahn AG. “No other client in Germany invests as much in infrastructure projects and their operation than Deutsche Bahn AG" [138]. German railways has defined a comprehensive BIM implementation strategy covering both, stations as well as infrastructure network. In this context, a set of 13 pilot projects have been conducted and scientifically analyzed. In 2022, all construction projects of German Railways are supposed to be executed as BIM projects. This involves the implementation of the procedures defined in ISO 19650 [139].

3.2 Current best practice in quality assurance

ISO 19650 demands the implementation of quality checks when project data is transitioning from one status to the other, i.e. from Work in progress to Shared or from Shared to Published. As part of the current best practice of BIM project execution, basic model quality checking is implemented already today, mainly in the context of model coordination and data handover to the client. However, the tests applied are limited to basic clash detection and simple checks for the provision of the attributes demanded by the client. Executing only these basic checks does not exploit the full potential of automated quality assurance that becomes available when using comprehensive geometric-semantic models of infrastructure assets. For example, there is currently no best practice for implementing high-level consistency checks of 4D (geometry + time) and 5D (geometry + time + costs) models. The lack of quality checking can result in severe errors with significant impact on project costs and project duration. This paper addresses this issue by providing an in-depth analysis of quality analysis in railway BIM projects. It presents comprehensive methods with which 3D, 4D, and 5D models can be systematically examined for possible errors.

The quality assurance mechanisms currently used for BIM infrastructure projects are generally limited to attribute checks and clash-detection. During clash detection, the model is often checked against itself, which can lead to the detection of numerous but insignificant clashes that are not the product of planning errors but can be attributed to inaccuracies in the respective software for infrastructure planning on the German market, which sometimes fail to automatically output entirely clash-free objects.

The fact that the BIM model created is not computer-tested completely represents a break in the digital chain of infrastructure planning. Errors in 3D models inevitably lead to errors in scheduling (construction sequence) and in the calculation of quantities and costs. The overall objective of "increasing planning accuracy and cost reliability", as outlined in the staged implementation plan, implies a need to improve the quality of planning processes. "High quality results and efficient workflow in the construction phase can only be achieved if the data basis is accurate" [102]. To achieve this, a quality assurance system tailored to model-based working processes is necessary.

Various software products are already available on the market to carry out corresponding quality checks. Examples are Navisworks Manage, Solibri Model Checker and Desite MD Pro [140–142].

3.3 Current state of research

“Project performance control can be defined as the identification of deviations between the desired and the actual performance of a project.” [143] or model. Navon states that “a comparison between the desired and the actual performances is the beginning of the control procedure”. The fact that an efficient, information-rich model-based working method requires standardized input data is, however, hard to reconcile with the often intuitive and rather unstructured planning process in early planning phases [144]. To exploit the potential of model-based working methods, it is therefore necessary to develop automated, standardized procedures for quality control.

Various approaches to digitally evaluating the quality of planning and models have been discussed in current research. Solihin and Eastman classify the possible quality criteria as follows [49]:

- Checks for well-formedness of a building model, i.e. the syntactic properties of the digital model
- Building regulatory code checking
- Specific client requirements
- Constructability and other contractor requirements
- Safety and other rules with possible programmed corrective actions
- Warrantee approvals
- BIM data completeness for handover to the facilities management

“The building industry uses numerous engineering standards, building codes, specifications, and regulations (henceforth, all are referred to as “regulations” for the purposes of brevity), and a diverse set of industry vocabularies to describe, assess, and deliver constructed facilities. These building regulations are available as hardcopy and searchable digital documents. Some building design software applications (e.g., building-energy analysis and fire-egress assessment) are available that include computer-interpretable representations of the logic and rules from relevant building regulations” [145]. A significant amount of research work has focused on verification of the 3D model with regard to compliance with standards and guidelines. Methods such as ‘Automated Code Checking’ or ‘Code Compliance Checking’ “allow speedier, dematerialized and more transparent review processes” [146]. An overview of work conducted in this area is outlined in [87,103]. Charles gives an overview of research dealing with code compliance checking and presents an approach on how to check for rule conformity using RDF [147]. The research presented by Getuli et al. uses code compliance checking in the context of health and safety on the construction site [148]. Preidel et al. [96] use a visual programming language to check for code compliance.

An important method for checking the constructability of objects is clash detection. “The goal of collision detection (also known as interference detection or contact determination) is to automatically report a geometric contact when it is about to occur or has actually occurred” [149]. “Collision detection between rigid, and/or soft bodies is important for many fields of computer science, e.g. for physically-based simulations, medical applications [...]” [150] and also for civil engineering. Schauer and Nüchter illustrate different clash detection procedures by means of two point clouds using the example of a railway wagon in a section of a tunnel [151]. Staub-French [152] describes the added value of 3D and 4D modeling and possible clash detection, but also points out that clashes cannot yet be detected automatically over the course of the construction project on site. Mawlana et al. [153] developed a method in conjunction with the planning of large motorway junctions with several flyovers for generating optimal construction sequences and avoiding clashes over the course of time.

The research conducted by Leite [154] and Zhang et al. [155] examines the model-based verification of safety risks on construction sites.

By integrating a so-called “POP quality model” and a 4D model, Chen and Huo [156] demonstrated that the processes and quality parameters of a construction site can be represented in an integrated model-based system. Alongside scheduling deadlines, the quality parameters also included the manufacturing tolerances of the respective structures.

The literature review shows that various approaches and systems for checking BIM models have been addressed in research, although in most cases they focus on individual aspects of building models. While they provide a basis from which to derive individual quality criteria, “Quality criteria are difficult to use […], if they become too abstract. A typical approach to rectify this issue is to disaggregate the complex criteria into a series of more understandable criteria of lower conceptual difficulty. A problem arises when a compact list of abstract or dense criteria is replaced by a long list of simpler ones, which in many cases can make them impractical and time-consuming“ [157]. Johansson et al. [158] recommend using model qualities with the help of the scoring system detailed in [157]. Here too, however, the research only pertains to the checking and evaluation of the CAD model structure.

In summary, a review of research shows that scant research work has been undertaken on model-based quality assurance in infrastructure planning. For the most part, current research activities have focused on individual rule classes and are not integrated into an overall quality assurance concept.

3.4 Quality assurance concept

While quality assurance mechanisms in the field of infrastructure planning are currently only applied on a random basis and several quality criteria are needed, which can be impractical and time-consuming compared to Company et al. [157], the model-based quality assurance concept proposed here is designed to be efficiently applicable and to use recurring rule types. Solihin and Eastman [49] categorize the rule types as follows:

(1) Rules that require a single or small number of explicit data
(2) Rules that require simple derived attribute values
(3) Rules that require extended data structure
(4) Rules that require a “proof of solution”

In this paper, rule types 1 and 2 are considered in more detail.

The individual work steps involved in model-based infrastructure planning are shown schematically in Figure 16.

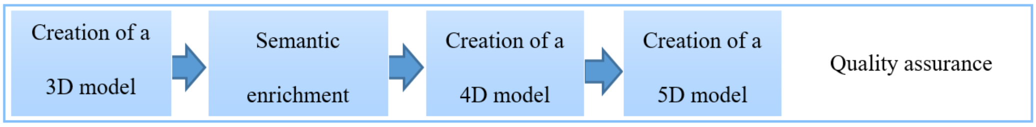


Figure 16: Procedure of model creation from 3D to 5D

The concept assumes that errors can occur in each of these steps. As such, it is necessary to have appropriate test routines in place in order to detect any errors that occur as part of quality assurance. Quality assurance checking can take place during the individual planning phases.

Eastman et al. [87] divide the inspection process into four phases, which have been adopted analogously in the present concept.

(1) Rule interpretation and its logical representation
(2) Building model preparation
(3) Rule execution
(4) Rule check reporting

Garrett et al. [145] define three steps for developing computable representation of regulations which are taken into account while creating the presented quality assurance concept. These principles are:

(1) Developing a simple understandable representation syntax for building-regulation writers and software developers
(2) Providing computerized support to enable regulation organizations to easily develop, test, and maintain these regulation representations
(3) Testing the sufficiency and implementability of the digital representations

“One of the key criteria […] is to be independent of any specific model-checker software used to check regulation compliance of building information models” [145]. Following these principles an independent model checker serves as the basis for model-based quality assurance. A database is linked to the program system, which contains both the test rules and the test results. The system design is shown in Figure 17.

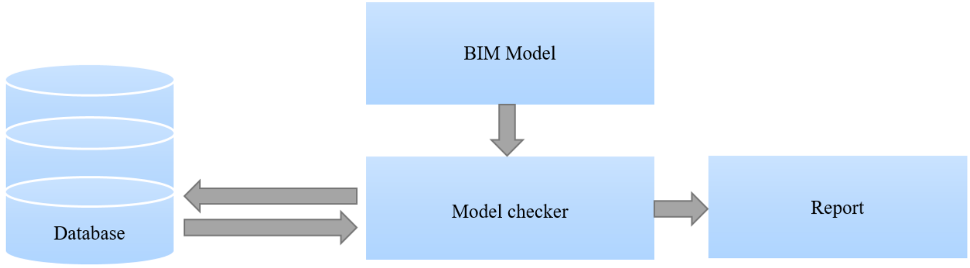


Figure 17: Software configuration, an independent model checker is linked to a database which includes both checking rules and checking results.

As part of literature research as shown in chapter 3.2 and 3.3, five domains have been identified for which various quality review mechanisms are assigned (see Figure 18) These are:

(1) Construction
(2) Clashes
(3) Semantics
(4) Construction sequence
(5) Quantities and costs

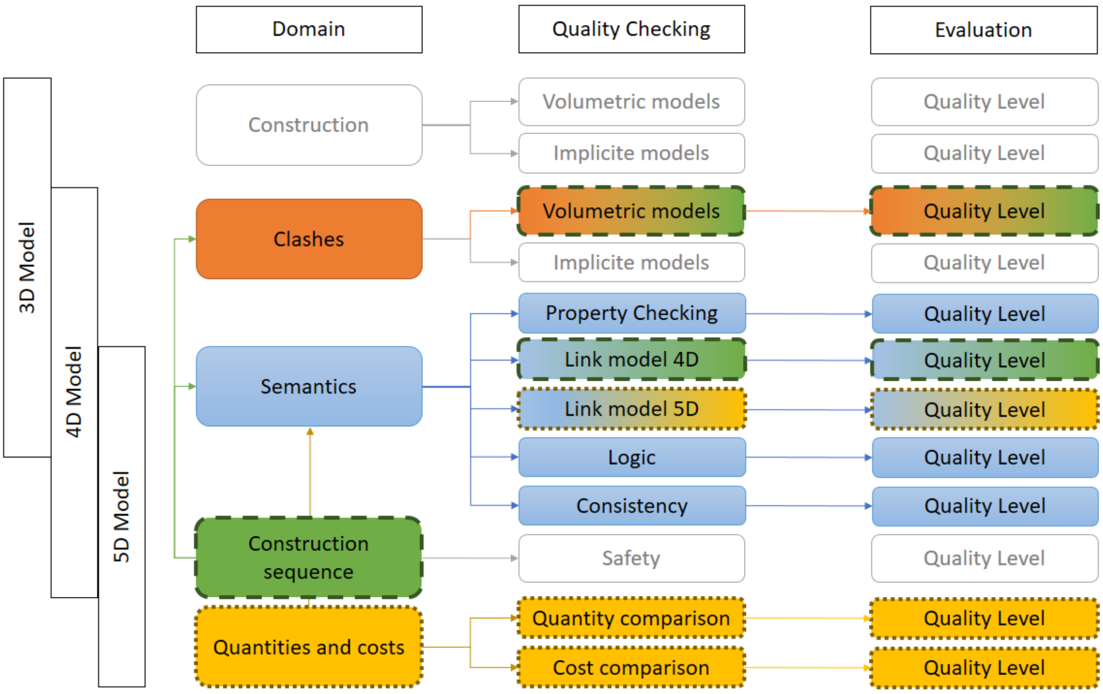


Figure 18: Quality assurance concept, the concept focuses on five domains. Each has their specifics aspects and needs for quality checks. To evaluate model quality, it is necessary to establish a generally applicable evaluation scheme.

Solihin et al. [159] is dealing with quality criteria of IFC exchanges and concludes that there is “an urgent need to define robust and rigorous test criteria, processes and tools.” This conclusion is not only valid for IFC exchanges but also for the quality for modeled infrastructure designs.

A 3D model must be considered in terms of its basic components of geometry and semantics. The geometry in turn comprises that of the construction itself and the resulting clashes when several objects are superimposed.

Most of the software products used for infrastructure planning offer a drawing-oriented view – split into site plan, cross-section and elevation – although these are stored internally as a three-dimensional model in the program. This type of model is referred to as implicit geometry description, since the parameters for creating the objects are saved, not the volume objects and their coordinates. The volumetric models are then generated from these parameters. As such, one should distinguish between volumetric 3D models that are the result of planning and the implicit models (2.5D models) used at the time of planning (see Figure 19). With implicit models, the governing design parameters become significantly more accessible than with explicit models. An example is the objects and parameters defining alignment. Accordingly, checking these parameters against codes and guidelines is more easily realizable with implicit models.

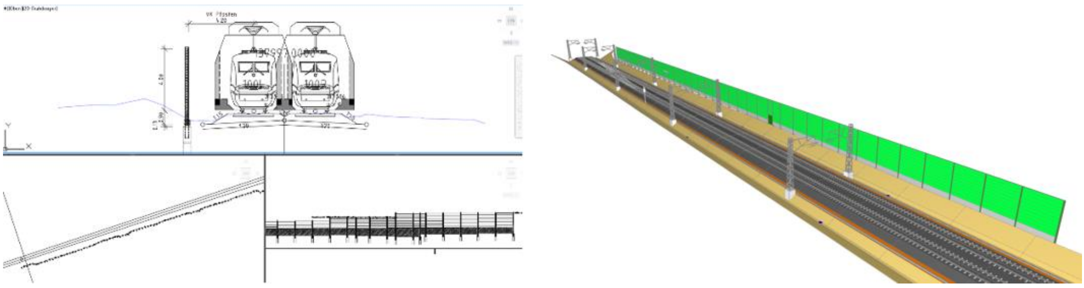


Figure 19: Comparison of implicit and volumetric 3D models, while implicit models (drawing-oriented view) are used at the time of planning, explicit models are used in the context of BIM-analysis

The quality of the 4D model, which comprises the 3D model, semantics and a schedule, is influenced by the “collisions” and “semantics” domains as well as by “construction process”. While the quality of the 5D model is largely informed by the domains “semantics” and “quantities and costs”.

The “construction” domain as well as the examination of a 4D model for “safety” will be dealt with in a subsequent step of the research project and will not be elaborated on in this paper. In the “semantics”, “construction sequence” and “quantities and costs” domains only volumetric models are considered.

To evaluate the model quality, it is necessary to establish a generally applicable evaluation scheme. This is referred to as Quality level.

The following sections explain the checking methods for the individual domains in more detail. In all the test methods, the aim is to minimize the work involved in preparing, carrying out and evaluating the results in order to achieve a time- and resource-efficient working method.

3.4.1 Model and database structure

Knowing the models’ structure is crucial for checking them. The concept presented here assumes that the model structure corresponds to a representation of built structures established in practice. In this example, it takes the form of the hierarchy shown for the “rail transport system” trade in Figure 20 and is described in detail by [8,160]. The trade is divided into different groups, each of which consist of different objects. Each object can be identified by specification features. For example, there are various rail shapes (e.g. S49, S54, UIC 60) which have an effect on the geometry of the object. This component logic is represented relationally in the database used.

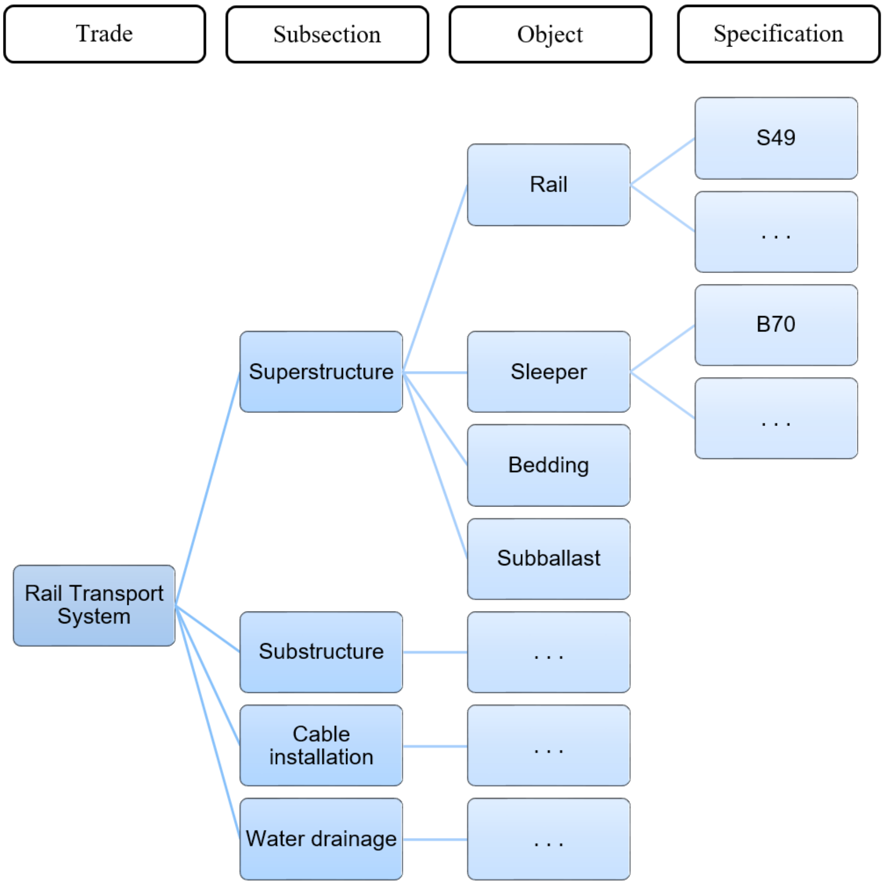


Figure 20: Building structure – the example of a rail transport system

3.4.2 Domain clashes

Clashes can occur in particular when merging different specialist models and thus plans from different specialist planners. There are several software products available, which offer good support for the automatic detection of such clashes. The BIM Center distinguishes between the following types of clashes [161], which are also considered in our concept.

- Hard Clash (HC) ≙ two or more objects overlap each other
- Soft Clash (SC) ≙ two or more objects come too close to each other, i.e. do not adhere to minimum distances between them
- 4D-Clash ≙ Clash during construction time

The software products for infrastructure planning currently available on the market have only limited ability to correlate different geometries with one another, producing clashes that do not exist in reality. These program-related clash detection errors are flagged up by the model checker but do not actually correspond to planning errors. The following irrelevant clashes (IC) are known examples of modeling errors:

- sleeper and ballast

- subgrade and manhole
- subgrade and mast of catenary or signal post

3.4.2.1 3D clash detection

Irrelevant clashes – produced by software and not by users – are often hard to avoid. But systematic clash detection errors can be avoided by explicitly specifying the objects that need to be checked for clashes. However, this entails both more preparatory work before undertaking the check and also introduces the risk of forgetting to include all the necessary checks. This approach is also not terribly efficient, since the process has to be repeated for each model.

The more common method of testing the model completely against itself for collisions is fairly quick, as no significant preparation is necessary, but requires a means of reducing the effort of evaluating the large number of clashes resulting from program-related modeling errors to a minimum. This can be partially automated with the help of a clash matrix that indicates the significance of collisions (see Table 3).

Table 3: Example of a clash matrix, HC – Hard Clash, IC – Irrelevant Clash

Object	Rail	Sleeper	Bedding
Rail	HC	HC	HC
Sleeper	HC	HC	IC
Bedding	HC	IC	

The clash matrix comes into play once clash detection has been undertaken using the model checker. Objects that clash are then checked against the matrix for permissible clashes and classified where appropriate as irrelevant. This therefore reduces the evaluation work necessary in the post-processing. An example is shown in Section 3.4.2.3.

The fact that the clash matrix can be updated and used across models also means this method is more efficient and sustainable. If necessary, it is also possible to define irrelevant clashes at group or specification level or across levels. The evaluation result is written to the results database, and the results can also be imported back into the model checker’s clash detection.

3.4.2.2 4D clash detection

With 3D clash detection, a model can only be checked at a fixed point in time. However, it is also relevant to consider the construction sequence in clash detection. This is particularly important for objects that do not exist at the beginning or at the end of the project phase, for example temporary constructions such as supporting scaffolds or shoring systems. In the literature, there is little evidence of approaches to this aspect.

For this, it is necessary to identify the status of individual objects of the 3D model over the course of the construction process. The following categories are relevant here (see Table 4):

Table 4: Categories during construction process

Category	Description
Existing	Objects that already exist at the beginning of the process
Deconstruction	Existing objects that will be dismantled and removed over time
New Construction	Objects that will be constructed over time
Temporary Construction	Objects that are erected at a time x and dismantled at a time y

To detect potential clashes, the various processes of the schedule are mapped as static clash detection instances. For each of the above categories, the relevant test sets of the project (the so-called “left” and “right” test sets) must be determined. An example illustrating the principle is shown in Section 3.4.2.3.

Logic dictates that at the start of construction (t=0), all existing objects are available. These are assigned to the left test set. Time t=1 marks the point after which the first work measures take place: for example some existing objects may have been deconstructed and removed or alternatively some new objects have been erected. If existing objects are deconstructed and removed, they should be removed from the left test set, since they cannot cause a clash. If new objects are created, they should be added to the right test set. Clash detection analysis can then be performed after both test sets have been created. For the next clash detection analysis at time t=2 after the next work measures, the objects from the right test set (newly created) are assigned to the left test set and the right test set is changed according to the work done, and so on.

As the objects that exist temporarily during the construction period can only be sensibly assigned to one category, they should be represented as “new during construction” and “removed during construction” and the point at which the change takes place recorded in the schedule. The assumption is that time objects in the schedule also contain details about the changes made to an object, so that these can be evaluated during clash detection. Where objects belong to “new during construction” at a defined point in the schedule, they are assigned to the right test set while those belonging to “removed during construction” removed from the left test set.

The clash detection analysis, its evaluation and the recording of the results in the database is carried out as described in Section 3.4.2.1.

Since with this method, the dynamic process of the construction site is reduced to individual points in time, the situation can arise where two processes take place simultaneously but are checked one after the other. This can lead to problems, in particular when an object that is deconstructed and removed in one of the two processes clashes with the objects from the second process. In certain circumstances, these clashes may not be recognized correctly. In such cases, the granularity of the 3D objects, processes and schedule times has a key impact on the result.

3.4.2.3 Case examples – Clash detection

The 3D models shown in the case studies were created with ProVI [162], and the 4D and 5D models as well as the test mechanisms were realized with Desite MD Pro [142].

The case of 3D clashes is demonstrated on the example of sleepers that have been bedded in ballast for better track stability (see Figure 21).

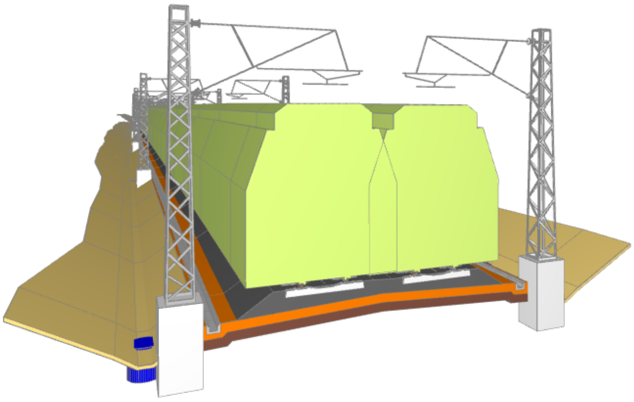


Figure 21: Irrelevant clash of sleeper and bedding caused by superimposition, clash due to missing recess in bedding

The planning software used generates the volume object for the bedding independently of the sleepers. As a result, clash detection analysis flags clashes at every point where the bedding and sleeper objects overlap. These are merely irrelevant clash detection errors and not actual project clashes. If one assumes a normed sleeper spacing of 0.6 m, this results in at least 1,667 clashes per kilometer of track that will require manual evaluation.

Figure 22 shows an example of how a hard clash and 4D clash can differ for a situation in which a road bridge is to be built over a railway line at a point where a mast for overhead lines stands. A time-independent clash detection analysis of the 3D model flags a clash. However, once the temporal development of the construction schedule is considered, a clash situation (4D clash) will only occur if the mast still stands at that position at the moment in time when the road bridge is built.

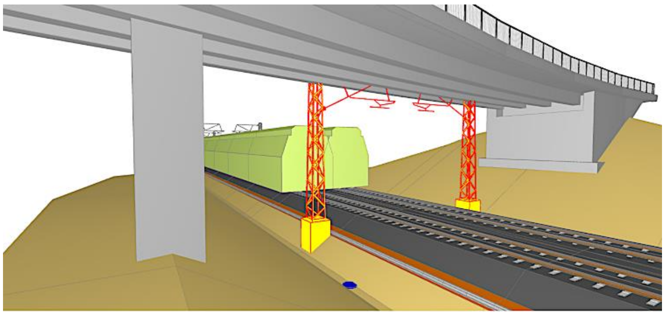


Figure 22: An existing mast structure and a new road bridge: a hard clash or a 4D clash?

3.4.3 Domain semantics

The following checks outline means of verifying the coherence and correctness of the semantic model. Various test routines are required which examine the semantics from different perspectives.

3.4.3.1 Attribute testing as per project specifications

The definition of attributes that a model may contain is a vital part of working successfully with models. They can range from relevant project information to the geometric properties of the objects, the materials used or operation-relevant data. Alongside the collision-free 3D and 4D models, these coordinated results represent a further quality parameter. In the field of building construction, standardized specifications already exist in the form of OmniClass [163], UniClass [164] or the buildingSMART Data Dictionary [165]. Corresponding software extensions for modeling software make it relatively easy to employ this data. Böger et al. [166] demonstrate how the buildingSMART Data Dictionary can be accessed right from the model creation phase with the help of a specially developed plug-in for the modeling software Autodesk Revit [42]. Although this method is not in itself a quality checking system, it contributes to limiting possible sources of error already at the model creation stage.

The existing classification instruments are, however, at present of little applicability for infrastructure planning. In Germany, several industry representatives have meanwhile joined forces to develop a similar standard within the framework of buildingSMART for the infrastructure construction sector in Germany. Up to now, each contractor in the Deutsche Bahn AG’s pilot projects has drawn up its own definition. And, as mentioned earlier, the verification of the models is rarely automated.

Although the various quality assurance tools offer corresponding test routines, they are not easily integrated into the concept described here. For this reason, an independent algorithm was developed that checks the attribute definition in the 3D model. The use of an open data schema such as mvdXML is also conceivable but has not been implemented here.

All objects are checked against the specifications according to the project specifications. During the check, the following status messages are returned with details of the object and attribute:

0 The object exists with the required attribute (value)
1 The 3D object found is not present in the project specifications
2 The required attribute is not defined in the 3D object
3 The attribute value of the 3D object does not correspond to the specified value range

In order to check value lists, a corresponding check rule must be formulated for each object. The functionality is explained in Section 3.4.3.2.

3.4.3.2 Checking link models

Technical project management encompasses construction design, scheduling, and cost planning. In conventional project management, these aspects are stored in independent documents (e.g. 2D plan and schedule) and require manual, cognitive input by the project manager to link them. The model-based working method presents a major advantage in that this information is linked digitally and logically through the 3D model via properties to the corresponding information in the time schedule (4D) and the quantity and cost calculation (5D). The linking of different information sources or specialist models is also referred to as a multi-model. “The basic idea of the multi-model is to combine selected specialist models from planning and project management in a single information resource and to map their dependencies through additional explicit link models” [167]. Studies at Stanford University “have shown that more project stakeholders can understand a construction schedule more quickly and completely with 4D visualizations than with the traditional construction management tools” [168]. The uses and benefits of 4D and 5D models are discussed by Fischer et al. [168,169].

The method of linking different sources of information is also called nD modeling and “is not limited to the domains represented in the building model. It concerns instead a concept for the integrated use of technical and building data. The aim is to create a multidimensional computer model to support the entire planning and construction process” [170]. These digital relationships are created with the help of link models. The 3D objects are filtered via their properties and linked to the processes of the 4D model or cost positions of the 5D model. “When filtering, a subset of the data is formed that corresponds to a predefined criterion, e.g. all processes in the month of May, all walls higher than 3 m or all specified services for concrete work. The criterion can be as complex as required, the only proviso being that they are calculable so that they can be executed automatically” [170]. Figure 23 shows the linking logic of the models and the necessary checks.

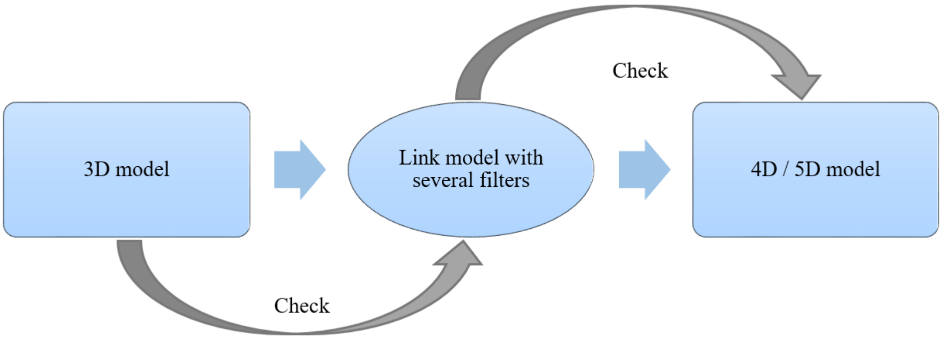


Figure 23: Linking logic of models. The correctness and completeness of these links should be considered as quality parameters. A model will only be of a high quality when the filters in a link model are correct and complete.

Since the interlinking of the various model objects is elementary for the BIM methodology, the correctness and completeness of these links should be considered as quality parameters. A model will only be of a high quality when the filters in a link model are correct and complete.

In the following, the procedure in principle is explained, although the process is identical for 4D and 5D models. In principle, the following situations must be distinguished in the evaluation and assessment (see also Figure 24):

(1) The filter of the link model is used several times with identical attribute values
(2) The filter of the link model is used several times with different attribute values
(3) There are several filters with identical attribute values
(4) There are several filters with different attribute values

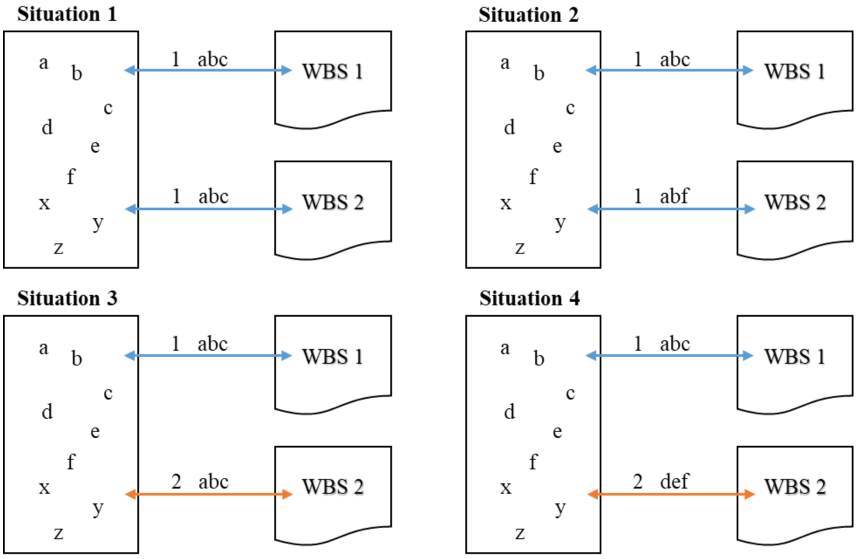


Figure 24: The different distinct filtering situations: the blue line describes link rule 1, orange line describes link rule 2 and letters a-z describe attribute values. A link rule can be used several times with the same or different values (situation 1 and 2).

In situation 1, the check rule is created only once, avoiding the need for multiple and identical check runs. In situations 2 to 4, several check rules must be created, since these are formally several filters that differ in at least one criterion. Corresponding case examples are presented in Section 3.4.3.3.

As with attribute checking, all objects are checked according to the project specifications (Section 3.4.3.1) against the same status messages. To filter the objects to be checked, all suitable checking rules that match a given criterion, e.g. object type, are determined and the check is carried out. In situation 2, this can mean that some objects within a filter may evaluate both positively and negatively. To correct this, a post-processing routine is necessary that checks whether objects in the negative check results also evaluated positively with the same filter number in the same attribute, and in such cases removes the object from the negative result set to ensure that the check results are evaluated correctly.

In addition, it can also be useful to create the same formal filter with the same properties but different filter numbers (situation 3), even if the same object types are queried. In this situation, the post-processing routine described above is not sufficient as it only searches for positive results with the same filter number. A second post-processing routine therefore queries the model structure described in Section 3.4.1 and checks whether the attribute value is a valid alternative value at the various levels. If it is found to be a valid alternative, the apparent error check is corrected.

After the above checks, what remains is the set of negative results. From these those objects that are not addressed by a filter in the link model can be identified. In addition to recognizing non-functioning links, it can also be an important indicator of missing processes in the scheduling plan or missing service items in the bill of quantities. An overview of the possible results situations is shown in Figure 25. On the left, the service items are shown with the link criteria (prop 1 to 4) and the corresponding property values (a to d). The right-hand side shows the objects of the 3D model with the respective attributes and their values.

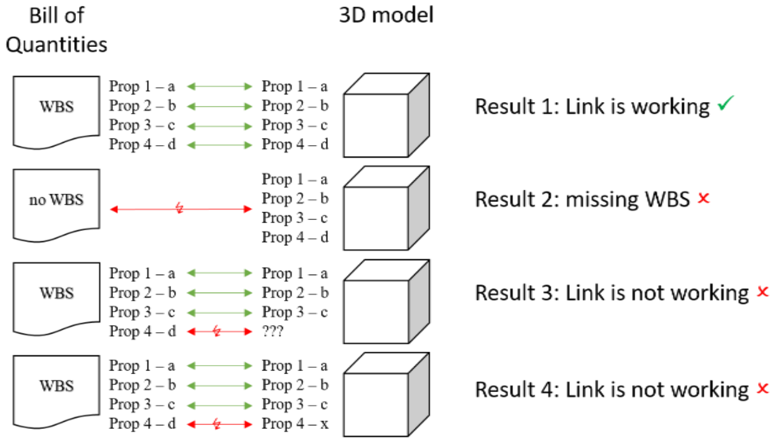


Figure 25: Overview of the results situations. On the left, the service items are shown with the link criteria (prop 1 to 4) and the corresponding property values (a to d). The right-hand side shows the objects of the 3D model with the respective attributes and their values. The green line (without flash) describes working links, red line (with flash) describes non-working links or links without a service item.

By the same token, in addition to checking the correctness of the link model against the 3D model, it also makes sense to check the filters against the process descriptions or service items. If, for example, object A (e.g. rail) is filtered from a 3D model, but object B (e.g. overhead line mast) is described in the respective linked position of the 4D or 5D model, a technical error occurs because the 3D object and the process or service position do not correspond. Although the filter finds the corresponding 3D objects, their content does not pertain to the linked position. The system checks per item or process whether the attribute values of the linking rules appear in the long texts or process descriptions. Because the system undertakes an exact string match of the attribute values against the descriptions, the terms contained in the descriptions must be written identically. If synonyms are used or the term itself is not used at all in the text, the check will flag an error. Synonyms can be identified with the help of a post-processing routine to reduce the degree of subsequent manual evaluation. For this, a corresponding synonym dictionary was created in the connected database that can be extended as required.

3.4.3.3 Case examples – Link models

The procedure explained below uses the example of a bill of quantity, but the process is identical for filter to the construction schedule.

To check the several filters in a link model, the linking criteria are first read and stored in a table. In certain cases, the same filter can be used for several service items (situation 2), usually when similar object types are specified in different service items. One example might be the posts of a noise barrier: depending on the structural calculations, different post profiles may be used for a noise barrier, e.g. HE-

B 180 and HE-B 200 wide flange profiles. While these must be listed separately in the bill of quantities, their semantic logic follows the same structure with the exception of the property value for the profile series.

An example for situation 3 might apply when the posts of the noise barrier have different foundation forms: for example, deep foundations could be driven or bored piles. To clearly delimit the service items, each is given its own independent filter. However, when searching for suitable objects only one criterion (here object type) is used for comparison in the course of property checking. Since both bored and driven-pipe piles match the “Foundation” object type, a driven-pipe pile will also be selected for checking when comparing against the rules for bored piles.

A case example for checking a filter to the 5D model can be illustrated by the following: The Deutsche Bahn AG provides model bill of quantities and prescribes their use when submitting tenders. For the practical tests carried out as part of the project, the model bill of quantities for noise barriers are examined. Based on the model structure shown in Figure 20 the various levels of connection logic were taken into account. Each filter begins with the object group (here “noise barrier”), followed by the object type and then a list of various specifications. While the Deutsche Bahn AG stipulates the use of their model service specifications as the standard for invitations to tender, checking the terminology used reveals non-standard inconsistencies in the wording of the service descriptions. Noise protection barriers are also referred to as sound insulation barriers and some long texts also explicitly state that these are descriptions of noise barriers, while others do not. When synonyms are used or even when terms are not used at all, the check will flag these up as errors, which increases the effort required for subsequent manual evaluation of the test results.

3.4.3.4 Logic checks for geometric properties

In addition to evaluating the coherence and consistency of semantics in the form of text values, it is also necessary to check the geometric properties of the 3D objects and derive corresponding quality parameters. To this end, various logic checks for geometric properties have been developed, which are stored in the database and can be extended or changed.

In some circumstances, it can happen that the export or import of 3D models via exchange format interfaces does not function perfectly. Typical problems are gaps between objects, surfaces that are not closed or vectors with incorrect orientations. This usually becomes evident when trying to evaluate the objects in the analysis software and often results in the geometric properties, especially the volume, being set to zero. With the help of the first logic check, all objects can be automatically checked for properties equal to zero. This quality check makes it possible to identify faulty objects and avoid incorrect evaluations.

The various authoring programs often also store the object’s geometric properties, such as volume, length, width, height, etc., as an attribute of the object. Analysis and evaluation software can independently determine these properties using their own calculation methods to verify the details and compare authoring and evaluation software. The second logic check therefore compares the individual geometric properties of both calculation sources with globally valid rules for each 3D object and saves the results in an object-specific manner. When comparing values such as height, length and width, the oriented bounding box is used because it describes the limits of the 3D object. The result is output as one of three possible statuses:

0	The comparison values agree
1	The comparison values differ from each other
–	There is no comparison value from the authoring software

The effect that the bounding box has on the comparison result is explained in a case example in Section 3.4.3.5.

In addition to the two logic checks mentioned above, one must also verify that the dimensions given for a 3D object are also correctly modeled. In research, this principle is described as semantic-geometric coherence and is sometimes used in 3D city models in combination with CityGML. “In the context of geodata, spatial-semantic consistency [...] describes the consistent relationship between spatial and semantic information” [171]. Daum and Borrmann extend the method to IFC models using the query language QL4BIM [102]. In the concept presented here, it is possible to link the rules database with the model structure and to store specific geometric properties at object or specification level. The value of these checks lies in detecting differences in the data in the geometric and semantic models. “High quality results and efficient workflow in the construction phase can only be achieved if the data basis is accurate” [102]. Correcting inconsistencies at this stage avoids errors further down the line, for example in the 5D modelling.

In the logic tests presented, a tolerance value in percent can also be specified in order to allow minor deviations between the comparison values. This tolerance value can also be used to map the different accuracy requirements within the individual work phases.

3.4.3.5 Case examples – Logic checks

To test for semantic-geometric coherence, it is possible to store specific geometric properties at an object or specification level linked to the model structure described in chapter 3.4.1. The oriented bounding box is used to compare geometric data against the respective checking rules. This can be determined by the model checker to differing degrees of accuracy and indicates the maximum dimensions of a 3D object (see Figure 26). The resulting degree of accuracy has implications for both the measurement results as well as the quality inspection. In Desite MD, for example, calculation accuracy is specified as a numerical precision with a standard value of 0.01. This means that the bounding box in all three coordinate directions is determined precisely to the second decimal place.

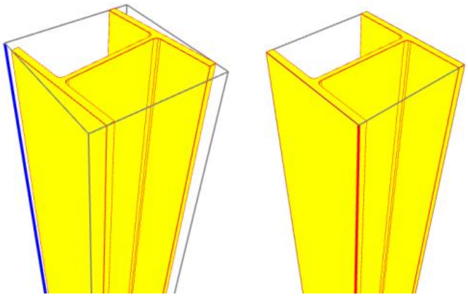


Figure 26: Bounding box (grey) with different degrees of accuracy: the left side is imprecise, the right side is precise.

To counteract any resulting inaccuracy, a tolerance value in percent can be stored in the database for the comparison test. For the tests performed on the wall and base elements of the aforementioned noise barrier, a permissible tolerance of 1.00% was set. The geometric property of the wall element and base thickness (d_req =0.12 m), which is defined by the bounding box as “cpOOBBWidth”, did not test positively to a numerical accuracy of 0.01 (bounding box calculation) and a permissible tolerance of 1.00% (n=1699). The deviations determined ranged from 1.0 to 19.5%. The measurement results of these objects are shown in Diagram 1.

In this case, the numerical accuracy for calculating the bounding box must be set to at least 0.001 in order to pass the control test with a tolerance of 1.00 %. Otherwise, the measurement results cannot be used as a quality criterion because system inaccuracies impact negatively on the apparent quality of the model.

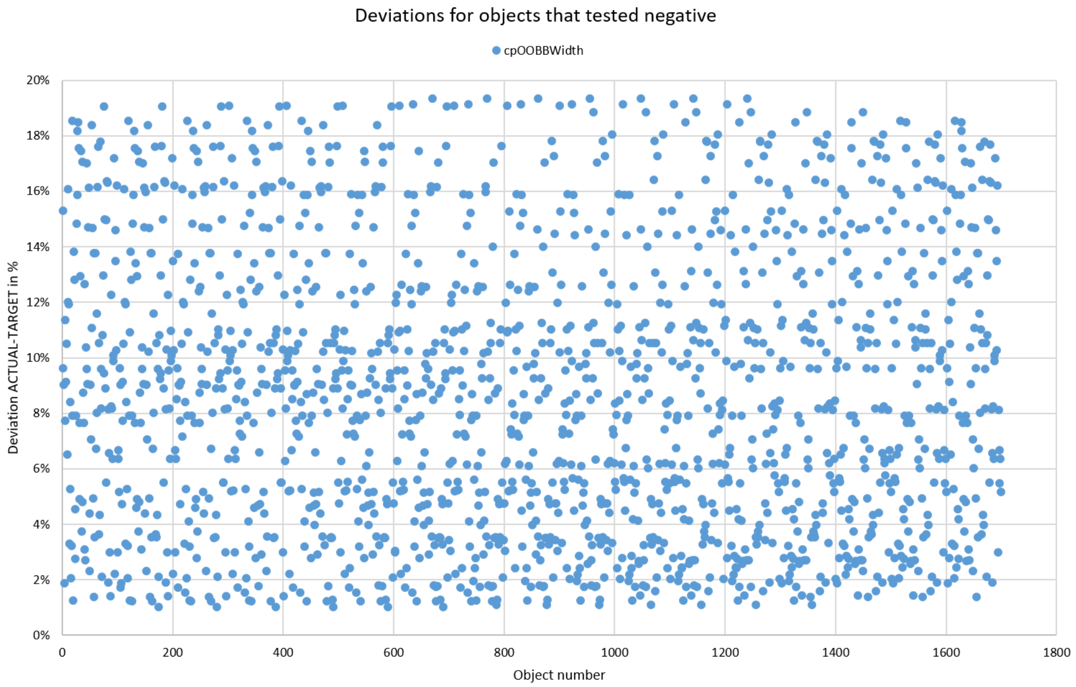


Diagram 1: Percentage deviation of the "thickness" of wall and base elements that tested negative. The geometric property of the wall element and base thickness (d_req =0.12 m), which is defined by the bounding box as "cpOOBBWidth", did not test positively to a numerical accuracy of 0.01 (bounding box calculation) and a permissible tolerance of 1.00% (n=1699). The deviations determined ranged from 1.0 to 19.5%.

It is also possible to carry out the corresponding tests at the specification level. Using the example once more of the noise barrier posts, the different profile series are characterized by different dimensions. For example, a post with a HE-A 160 profile has the dimensions h × w = 0.160 × 0.152 m, whereas a HE-A 180 profile has the dimensions h × w = 0.180 × 0.171 m. These rules can also be checked automatically.

Diagram 2 shows the results for the geometric property "post width" (w = 0.152 m), which were determined by the analysis software using a bounding box (numerical accuracy 0.01). Only standard posts were evaluated, since corner posts have irregular dimensions and cannot be tested by the logic. The deviation of the target from the actual dimensions (bounding box) lies between 0.0 and 0.5% for all the evaluated objects (n=188).

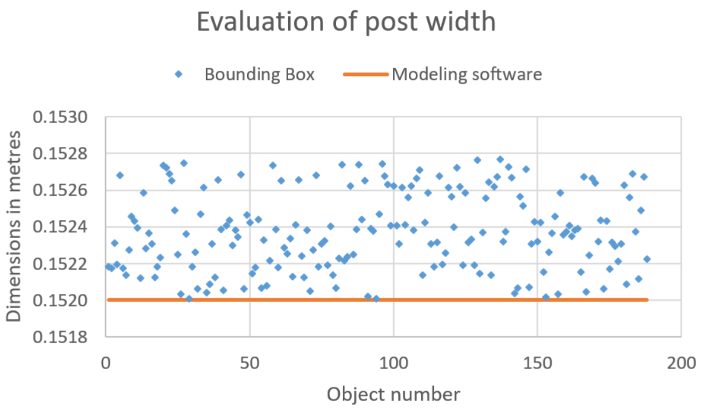


Diagram 2: Evaluation of the attribute "post width" for standard posts, shows the results for the geometric property "post width" (w = 0.152 m), which were determined by the analysis software using a bounding box (numerical accuracy 0.01).

The properties for post height and width therefore tested positive with a permissible tolerance of 1.00% and a numerical accuracy of the bounding box of 0.01.

A key problem of this approach is the difference between global and object-oriented coordinate systems. While in the model checker, the three coordinate planes x, y, and z describe the geometrical properties length, width and height in global terms, the coordinate system in the infrastructure software is object-oriented. When the geometric information in the 3D model is object-oriented (Figure 27, right) but the coordinate system used in the evaluation software is global (Figure 27, left), it is necessary to formulate several rules. For example, the condition height (global) ≙ length (object-specific) applies to the post, whereas length (global) ≙ length (object-specific) applies to the wall element. For the check, it is therefore advisable to formulate independent rules for the necessary combinations. This inevitably produces false checks, which can, however, be determined and filtered out with a post-processing routine.

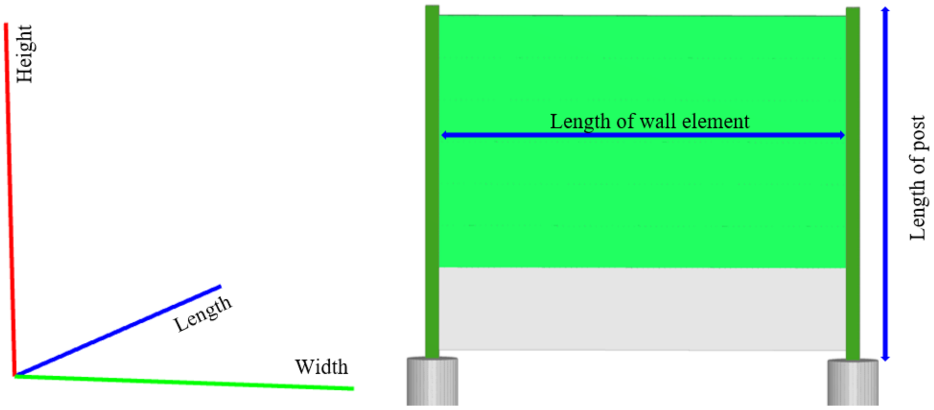


Figure 27: Difference between global and object-specific coordinate systems

Alongside orthogonal objects, as shown here using the example of a noise barrier, infrastructure constructions frequently have objects arranged along a three-dimensional curved path and that are rotated around their coordinate axes, as is the case with railway tracks. To test the influence of alignment parameters on three-dimensional object formation, test models were developed under standardized boundary conditions and tested using the same mechanisms. The curved path is essentially the product of the superimposition of axis and gradient. The axis is constructed as straight lines, circular arcs, and transitional segments. The axis also defines the cant of the rails, which in the transitional segment leads to a twisting of the objects around the longitudinal axis. The gradient in turn consist of straight pieces and rounding arcs. The latter results in a bending of the objects around their transverse axis, but this was not considered to simplify plausibility checking of the results. For the test models, a fictitious route was designed. For its design, discretionary limits were used as discussed in [172] that in real-life layouts are not permissible in such combinations but here make it possible to generate objects that are as twisted as possible in order to verify the measurement method. The parameters of the alignment for the test case are given in Table 5.

Table 5: Test case parameters for testing the influence of alignment parameters on three-dimensional object formation

Axis parameter	Length of straight section	100	m
	Cant of straight section	0	mm
	Length of Bloss transition arc	105	m
	Length of circular arc	100	m
	Radius of circular arc	300	m
	Cant of circular arc	160	mm

Gradient parameter	Longitudinal gradient 1	0	‰
	Longitudinal gradient 2	5	‰
	Longitudinal gradient 3	10	‰
	Longitudinal gradient 4	12,5	‰

Calculation interval	Calculation interval 1	1	m
	Calculation interval 2	5	m
	Calculation interval 3	10	m
	Calculation interval 4	20	m

Object formation in infrastructure planning frequently involves the extrusion of a cross section, with new cross sections calculated at regular intervals and connected linearly with each other. This step of model creation therefore also influences the accuracy of the objects created. The effect of different calculation intervals on the resulting 3D objects is shown in Figure 28. The figure shows the calculation results for tracks with the same underlying alignment (axis and gradient with longitudinal gradient 1) at different calculation intervals. While the rail tracks on the left side (calculation interval 1) follow the real situation well (continuous course), also in the curved section, the rail tracks on the right side (calculation interval 4) clearly deviate from the real situation.

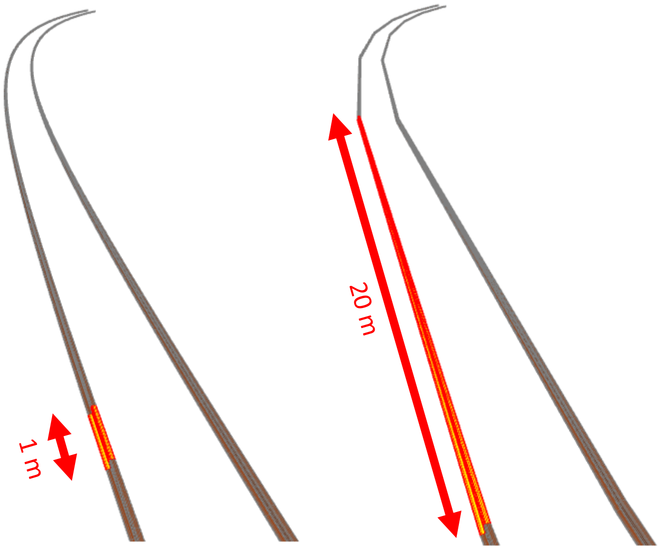


Figure 28: Calculation results of 3D objects at different intervals. The calculation results for tracks with the same underlying alignment (axis and gradient with longitudinal gradient 1) at different calculation intervals (1m and 20m) are shown. While the rail tracks on the left follow the real situation well (continuous course), also in the curved section, the rail tracks on the right clearly deviate from the real situation.

The interaction between the calculation interval and the cant influences the calculation of the bounding box. To illustrate this, Figure 29 shows an example of a superelevated rail. While the rail has no cant at the beginning of the section of track (left) and the bounding box perfectly matches the base of the rail, the rail at the end (right) tilts due to its cant. The bounding box, however, follows the maximum expansion of the 3D object and does not exactly fit the base of the rail profile. The situation is reversed at the top of the rail. The bounding box is therefore higher than the rail profile itself for the section where the rail is elevated.

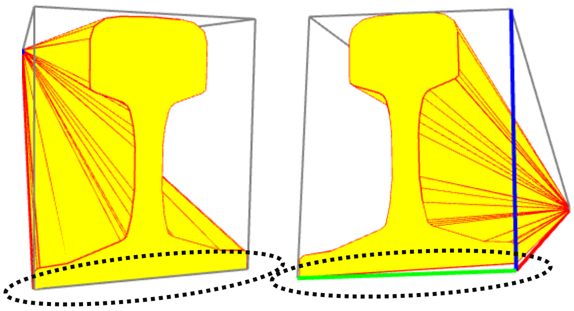


Figure 29: Effect of the elevation on the bounding box. The bounding box follows the maximum expansion of the 3D object and does not exactly fit the base of the rail profile in segments with an elevation greater than 0.

Taking into account the axis, the four gradients and four calculation intervals, 16 test cases result. The rail height of a S54 rail (h = 0.154 m) and the rail foot width (w = 0.125 m) are evaluated. As per description the accuracy of the bounding box has a decisive effect on the test, the numerical accuracy was set to 0.0001 so that the fourth decimal place of the bounding box adapts exactly to the 3D object. The two geometric properties were then evaluated and compared with the target values. The mean deviation of the objects that flagged as negative in testing is shown in Diagram 3 and ranges from 0.02% to 1.20%.

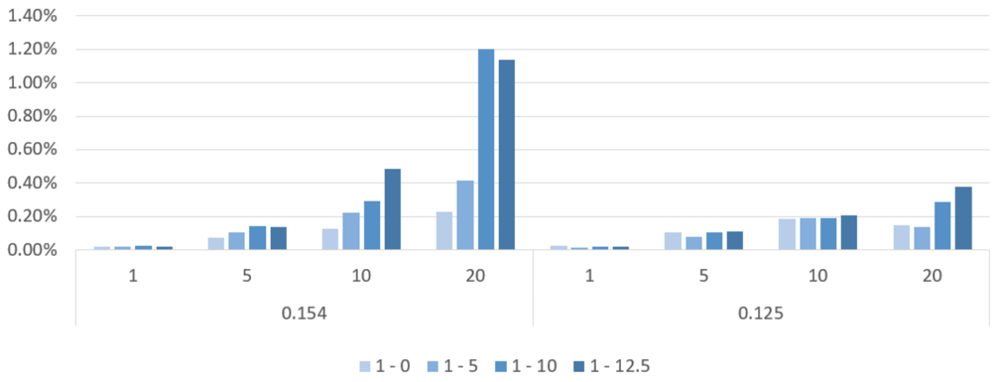


Diagram 3: Average deviation of objects inspected with errors. The rail height of a S54 rail (h = 0.154 m) and the rail foot width (w = 0.125 m) are evaluated. The mean deviation of the objects that flagged as negative in testing ranges from 0.02% to 1.20% depend on the four gradients (0 – 12.5 ‰) and four calculation intervals (1 – 20 m).

The minimum and maximum deviations of the negatively tested objects per test case are shown in Table 6. As expected, the greatest deviations occur at a calculation interval of 20 m: in combination with a high longitudinal inclination, the maximum deviation increases to 5.0%. This applies particularly to the objects in the transition curve, where the twist is greatest.

Table 6: Maximum and minimum deviation per test case. The greatest deviations occur at a calculation interval of 20 m: in combination with a high longitudinal inclination, the maximum deviation increases to 5.0%.

		Longitudinal gradient							
		0		5		10		12.5	
Property [m]	Interval [m]	Max	Min	Max	Min	Max	Min	Max	Min
h = 0.154	1	0.064%	0.000%	0.067%	0.003%	0.049%	-0.001%	0.049%	-0.002%
	5	0.258%	0.000%	0.282%	0.026%	0.315%	0.042%	0.261%	0.014%
	10	0.562%	0.000%	0.594%	0.056%	0.520%	0.100%	4.703%	0.016%
	20	1.333%	0.000%	1.300%	0.056%	4.847%	0.100%	5.032%	0.049%
w = 0.125	1	0.053%	0.000%	0.054%	0.000%	0.052%	0.000%	0.051%	0.000%
	5	0.241%	0.000%	0.248%	0.000%	0.255%	-0.002%	0.251%	0.000%
	10	0.508%	0.000%	0.512%	0.000%	0.516%	0.000%	0.572%	-0.010%
	20	0.931%	0.000%	1.028%	-0.046%	1.099%	-0.034%	1.043%	-0.040%

In addition, it is noticeable that the minimum values lie in the range of -0.04 to 0.1%. The minimum values shown at 0.00% only deviate in the third decimal place. A total of 3,304 objects were checked. With a tolerance value of 1.20%, 100% of the objects passed the test in terms of rail foot width and 99.3% of the objects with respect to rail height.

3.4.3.6 Consistency check: component semantics

Since constructions follow a structure established in practice (see Section 3.4.1), one must also check that the model corresponds to the expected structure. To this end, the 3D objects or their semantics are compared with the expected model structure (see Figure 20 for an example).

Deviations or erroneous results indicate either that the digital model structure in the database is incomplete or that the expected value has actually been violated. This makes it possible to detect inconsistencies in the semantic model and in turn to avoid evaluation errors.

3.4.4 The domain “quantities and costs”

In addition to checking the various aspects of the geometric and semantic models, the results of the 5D modelling likewise need checking. For the most part, this concerns quantities and cost evaluations, and corresponding test methods were devised and incorporated into the overall concept as described below.

3.4.4.1 5D model – Quantity checking

In the context of ensuring a continuous digital chain for model-based construction planning, it is necessary to develop a checking mechanism for validating the quantities determined. A 5D model is created from the building blocks “3D model”, “link model”, “quantity formula”, and “unit” and these must therefore also be considered in any verification procedure. In a first step, those service items in the specification that have a link to 3D objects are determined. The quantity recorded and the unit of measure is likewise retrieved for each item. The connected database contains rules for the individual units of measure, which reference the independently determined properties of the evaluation software. A tolerance value in percent can be specified in the rule definition, which is taken into account during the check.

Where evaluation programs allow the flexible input of mathematical functions using factors/quotients or similar when creating 5D models, these too must be validated and taken into account when checking the quantities.

In cases where the same unit is used – for example the square meter is universally used to denote areas – it is currently not possible to automatically recognize which area is concerned. It could be the footprint, the elevation surface or the entire surface of the tested object, each of which has a different absolute value. In the database, all surface attributes are defined as rules, which entails several test runs and therefore produces more erroneous results. A post-processing routine can, however, determine whether a service item unit that was flagged as incorrect was identified as being correct for another attribute or area value. The error code of the “wrong” area value is then automatically adjusted.

3.4.4.2 5D model – Cost checking

In addition to checking the model-based quantity determination, it is necessary to validate the unit prices for each service item. An object can, however, have multiple different unit prices depending on the unit. It must therefore be possible to define separate cost calculation rules specific to the object and to the unit. This is possible in the rule database.

It is also necessary to be able to define unit prices according to model´s level of detail. While the model and object structure may still be quite rough in early work phases and detailed only as far as the object level shown in Figure 20, the objects must nevertheless be costed at specification level as part of the preliminary planning and tendering.

This requirement is likewise supported by the relational database. For better user-friendliness, the permissible upper and lower limits of the unit prices can be defined as absolute values.

The method was developed conceptually and tested on a theoretical example. While the determination of the lower and upper cost limits at object level was not carried out on the basis of real projects, this data can be determined, for example, according to Sajadfar and Ma [173], where the historical data was evaluated using both regression analysis and data mining methods. These can, however, also be determined for a specific project using a unit price catalog.

3.5 Quality metrics and evaluation

The aim of the evaluations is to compare the elaborated quality criteria according to a standardized scale and in turn to identify those criteria which meet the project requirements and those which still need improvement. There are various ways of conducting this assessment. The following systems have been implemented in the concept presented here:

- Percentage scale
- Grades
- Traffic light scale

To define the correctness of a BIM model in a qualitative manner, the concept of “Quality Level” is used. A quality level describes the ratio of false results to checks conducted and so the quality of the BIM model for each quality parameter can be determined on a percentage scale. The concept considers six “Quality Levels”, as outlined in Table 7. The ratios used for each level are freely definable but should at least be defined consistently across a project. For cross-project comparisons, it is advisable to determine the ratio ranges once and apply them consistently to the different projects. A possible approach for a cross-project comparison method is outlined by Choi and Leite [174] although in that case the comparison parameters do not describe the digital correctness of the model components – as implemented here – but compare the results of different planning projects (costs, time, equipment, etc.).

The verbal description of the individual levels is based on the school grading system. Each “Quality level” is also given a corresponding color that follows the pattern of a traffic light scale. Green corresponds to levels A and B, with a slight color shift from green to yellow-green. Level C is assigned the color yellow, followed by orange for D, red-orange for E and red for F. This same concept for data quality visualization was used by Lee et al. [175] to verify the data integrity of IFC models.

Table 7: Evaluation metrics based on the ratio of false results to checks conducted and determined on a percentage scale. The ratios used for each level are freely definable. The verbal description of the individual levels is based on the school grading system.

Quality level	**Ratio of false results to checks conducted [%]**	**Description**	**Corresponding color**
A	0	very good model quality	A
B	$0 < x < 5$	good model quality	B
C	$5 \leq x < 10$	satisfactory model quality	C
D	$10 \leq x < 25$	sufficient model quality	D
E	$25 \leq x < 50$	poor model quality	E
F	$50 \leq x$	insufficient model quality	F

3.6 Validation

The concept presented in this paper was applied to a large-scale German project and tested for feasibility. Due to the degree of progress of the project, it was only possible to conduct tests that apply to the 3D and 4D model. Likewise link models could not be tests as the project does not employ them. The linear transport infrastructure project has an overall length of approximately 16.0 km and comprises 108,976 3D objects in the model. The results of the validation are presented below.

At the beginning of the project, a catalog of attribute definitions was agreed with the client and the project was then initially tested for compliance. 42 different attributes were defined, to which the model objects must adhere in different combinations, and a total of 1,511,279 checks were carried out. The check revealed that the agreed definitions had not been implemented consistently throughout the model as errors were identified in the semantic model in 32% (absolute 483,445) of the check runs. In this test, the model only reached quality level E.

In a second step, various logic tests were performed to check the semantic correctness of the model. To begin with, the geometric properties supplied by the authoring software in the attributes were compared against the evaluations of the model checker. Here, 762,832 checks were performed, and errors were detected in 15.36 % of the checks (absolute 117,190). In this test, the model achieved quality level D.

When checking the geometric properties determined by the model checker, 2,870 erroneous results were found (435,904 tests in total), indicating that some objects are not evaluable. This corresponds to an error ratio of 0.66% and a quality level of B.

In the object database as described in chapter 3.4.1, the standardized geometric dimensions of the elements of a noise barrier as well as of rails and sleepers were stored and checked for consistency in the model. According to the building logic, a distinction must be made between testing at object level and testing at object specification level. The test yielded a total of 33 false results for 7,634 inspections at object level (0.43% and quality level B). The test at object specification level yielded 2,105 incorrect results in 19,916 tests, which corresponds to an error rate of 10.57% and a quality level of D.

As described in chapter 3.4.3.6, the consistency of the model was also checked against the expected building logic described in the comparison database. 108,976 3D objects and their semantic logic were tested. In 15.81% of the tests, the expected value was not met, which is sufficient for a quality level of D.

The 3D model was also subjected to a collision check. First, the static model was examined: 103,528 collisions were detected. In order to filter out irrelevant collisions, the collision matrix described in Chapter 3.4.2.1 was extended and the results of the collision check re-verified. 64,997 of the collisions could be classified as irrelevant, so that the remaining error rate was 37.22%. The model therefore achieved quality level E in the “3D collision” check. In addition, the time schedule was also included in the collision check, taking the time dependencies into account. A total of 31,501 collisions were detected in the 4D collision check. After evaluating and discounting the irrelevant collisions, 92 % (absolute 28,935) remained, corresponding to a quality level of F. In total 28,633 clashes were identified between rails and sleepers, which follows from sleepers modeled in wrong height.

The validation process made it possible to evaluate and identify inconsistencies within the BIM model. The results helped to improve the subsequent processing of the model and have had a lasting positive influence on the quality of the model.

3.7 Conclusion

The application of the BIM method aims to make the entire process of a construction project more efficient. Initial studies have confirmed that BIM has a positive influence on the course of a project in terms of costs, time, communication, coordination, and quality. With the help of the BIM method, errors and their sources can be detected better and earlier. However, new error sources can arise in the process, which can have an impact on the model quality and in turn on all subsequent processes. This paper has presented a method with which 3D, 4D, and 5D models can be systematically examined for possible errors. This method was explored in the context of model-based rail infrastructure planning.

An overall quality assurance concept has been developed drawing on current best practice. A total of 14 quality parameters were developed for the three domains of “clash detection”, “semantics”, and “quantities and costs”, each of which were examined in more detail. Infrastructure planning, and model creation in infrastructure planning in particular, exhibit some special characteristics – as shown by the example of rail objects that are dynamically defined by route alignment – which were considered in the investigation. Finally, an evaluation metric was presented which allows model quality to be measured based on previously defined threshold values for the individual criteria. The quality assurance concept was applied on a large-scale infrastructure project. Depending on the selected quality parameter the quality of a model can vary considerably.

While the investigations described here were limited to explicit volumetric models, implicit models (2.5D models) will also be considered in future research, since these are an important pillar of digital project design for infrastructure planning. Further research is also needed in the domain of “construction”. Considerable research work has already been conducted in the field of building construction and it will be necessary to examine their applicability and feasibility for use in the field of infrastructure planning. A further benefit of model-based design is the possibility to carry out simulations. In practice, the focus here is usually on simulating the construction sequence. The aspects of safety checks and simulations will therefore also be a focus of future research activities.

4 Code compliance checking of railway designs by integrating BIM, BPMN and DMN

Previously published as: Marco Häußler, Sebastian Esser, André Borrmann. *Code compliance checking of railway designs by integrating BIM, BPMN and DMN,* Automation in Construction, Volume 121, 2021, 103427, ISSN 0926-5805, https://doi.org/10.1016/j.autcon.2020.103427.

Abstract

Code compliance checking has been the subject of scientific research for more than four decades and has been put into practice in numerous projects. To date, however, no universally valid, sustainable approach to the rule-based compliance checking of models has been established. Visual programming languages are easier to understand and thus more transparent than textual formats. The study presented here analyzes the requirements specified in the guidelines of the Deutsche Bahn AG regarding the technical design of structures in railway construction and examines the feasibility of implementing these rules using BPMN and DMN. The rules analyzed are categorized into 12 different classes. Depending on the guideline subset, the BPMN/DMN approach was found to be useable in 37%-75 % of the 943 rules examined. Considering only those rules that are relevant for the digital railway model, 68 % of the rules can be represented and automated using BPMN and DMN.

4.1 Introduction

Digital methods like building information modeling (BIM) offer considerable advantages in the construction industry over the conventional methods that still largely prevail in current practice. “BIM makes a new level of optimization possible. Most notably, it can afford greater planning, scheduling and cost certainty through transparency over the entire life cycle of a built asset. It simplifies risk management and provides better control of planning quality and industrial manufacturing processes. For the client, the main advantage of BIM is the creation of comprehensive, openly accessible building information that can be used by many different parties” [58]. The key motives for introducing BIM at a national level are therefore planning quality, cost minimization, risk management and increased efficiency (see also [3]).

The quality assurance concept for infrastructure planning presented in [176] identifies five domains to be considered over the course of model validation. While the domains ‘Clashes’, ‘Semantics’, ‘Construction Sequence’ and ‘Quantities and Costs’ have been examined in detail, the ‘Construction Design’ domain has not yet been addressed. This aim of this study is to fill this gap.

On the basis of [176], the ‘Construction Design’ domain is defined as the phase of creating a 3D digital building model taking into account the regulations set by the authorities. The domain is relevant for designers in the preliminary, conceptual and detailed design stages.

DIN EN ISO 9000-2015-11 defines ‘quality’ as the “degree to which a set of inherent characteristics of an object fulfils requirements” and “can be used with adjectives such as poor, good or excellent.” The term ‘requirement’ is defined as a “need or expectation that is stated, generally implied or obligatory” while “a specified requirement is one that is stated, for example in documented information” [47]. If the specified requirements are not met, EN ISO 9000-2015-11 speaks of non-compliance or errors. In relation to the design of building structures, compliance with the documented requirements for a building type at the design stage means that the design can be described as error-free and of excellent quality. “In the Architecture, Engineering, and Construction (AEC) industry, building projects must be checked against numerous building codes for compliance. They are allowed to be executed only when compliance with all applicable rules of the building code has been guaranteed. Failure to correctly assess projects for compliance can also have negative effects on building performance and allow errors that are expensive to correct.” [5]. The aim must therefore be to employ quality assurance measures that minimize both errors in design and, in turn, additional costs that may arise through deviations from the defined requirements. According to [4], ‘efficiency’ is defined as the relationship “between the result achieved and the resources used”. In the context of digitalization, this means that increasing automation of building model checks will result in a more efficient planning checking process.

The field of rail design, in particular, is subject to a large number of rules that must be complied with to ensure the functional and safe operation of railway infrastructure. Although, in contrast to many other fields, these rules are generally already elaborated in a highly formalized manner, they mostly exist as human-readable text and not in a machine-processable form. As such, the introduction of code compliance checking procedures is highly desirable, especially in railway engineering, so as to improve planning efficiency and ensure the necessary high level of planning quality. An essential prerequisite for code compliance checking is that the planning data is available in a digital form, preferably as a digital building model, also known as a ‘building information model’ (BIM). As this is now increasingly the case, this paper focuses on the digital description of rule sets and checking processes.

Code compliance checking has been the subject of scientific research for more than four decades [177] and has been put into practice in numerous projects. However, most studies have focused on building design and not on infrastructure projects. To date, no universally valid, sustainable approach to the rule-based compliance checking of models has been established [6]. Black-box solutions with hard-coded hidden, implementations of rules in vendor-specific solutions hamper the broader adoption of checking mechanisms [7,8]. To overcome this limitation, recent research has focused on the development of open and transparent methods of rule encoding, based on general-purpose programming languages or domain-specific solutions [179–182]. However, while textual programming languages tend to be challenging for AEC practitioners, visual programming languages enjoy increasing acceptance and widespread use by domain experts, as “information systems which are described by a visual language can be interpreted much faster and easier by humans” [96,183,184].

In contrast to previous studies, which are either software-specific or describe the development of a proprietary visual programming language, this study investigates an approach based on known and standardized elements of business-process modeling and makes use of Business Process Model and Notation (ISO/IEC 19510:2013). Alongside the visual, process-based representation of guidelines, a workflow engine was used to execute the processes developed and check existing models. In addition, the study investigates the extent to which existing definitions of linear reference systems already exist in the Industry Foundation Classes (IFC) data exchange standard that can be employed for checking purposes. The paper focuses on the automated checking of building information models representing the design of a railway project, in accordance with the quality assurance concept described in [176].

Thus, the research questions that this paper aims to answer are: (1) Are the standardized Business Process Model and Notation (BPMN) and Decision Model and Notation (DMN) sufficiently expressive for encoding the regulations of railway engineering in a manner that enables automated code compliance checking? (2) Do the railway BIM models presented in the IFC format provide all the required information?

The paper is organized as follows: Section 2 provides an overview of the state of the art on the subject of code compliance checking, before going on to describe and discuss the BPMN and DMN standards in more detail. While Section 3 introduces the relevant guidelines issued by German Railways (Deutsche Bahn AG), Section 4 discusses the representation of these rules by means of BPMN and DMN to achieve the desired automation of the code checking process. Section 5 investigates in detail to what extent the DB guidelines can be implemented using the developed approach. In Section 6, three representative case studies are examined, before Section 7 briefly presents the front end of the developed system. Section 8 concludes the paper and discusses its main findings.

4.2 Related work

4.2.1 Code compliance checking

Overview

Automated checking of standards and guidelines has been a focus of scientific research for many years, which is no surprise given the central role of regulations in the building industry: “As part of the design process, building designers ensure that every aspect of their design adheres to various regulatory requirements. The design is then subject to formal audit by the consent processing authority as part of the approval process” [103]. “The building industry uses numerous engineering standards, building codes, specifications, and regulations […] and a diverse set of industry vocabularies to describe, assess, and deliver constructed facilities. These building regulations are available as hardcopy and searchable digital documents. Some building design software applications (e.g., building-energy analysis and fire-egress assessment) are available that include computer-interpretable representations of the logic and rules from relevant building regulations” [145]. “Legal knowledge, in particular, is conveyed in voluminous paper-based documents in natural language text written for human interpretation” [106].

Until now, “the conventional practice of code compliance checking in the industry has largely been a manual process, which is laborious, costly, and error prone” [103]. In most countries throughout the world, conventional practice is based on drawings rather than semantically rich building models: “Nowadays the checking process is performed to a large extent manually based on two-dimensional technical drawings and textual documents by the responsible planning consultant as well as the building permission authorities” [98].

According to the McGraw-Hill Construction SmartMarket Report, compliance checking took 25 hours or more in 39% of the cases investigated, and in 11% of the cases more than 100 hours. On average, between 49 and 60 hours were spent on compliance checking [185].

The automated checking of models against guidelines, standards etc. is called ‘automated code checking’ or ‘code compliance checking’ and can result in “speedier, dematerialized and more transparent review processes” [146].

A large number of studies have been conducted on the subject, the majority of which focus on applications in building designs. Building models can be examined according to a variety of aspects, and therefore different quality criteria can be defined. Solihin and Eastman classify the possible quality criteria as follows [49]:

- Checks for the well-formedness of a building model, i.e. the syntactic properties of the digital model
- Building regulatory code checking
- Specific client requirements
- Constructability and other contractor requirements

- Safety and other rules with possible programmed corrective actions
- Warrantee approvals
- BIM data completeness for handover to facilities management

Phases of automated compliance checking

Eastman et al [87] identify four phases of the review process, which have also been adopted for this study:

(1) Interpretation and logical representation of rules
(2) Building model preparation
(3) Rule execution
(4) Rule check reporting

Garrett et al. [145] define the translation and implementation process of review procedures as follows:

(1) Development of a simple and easily understandable representation syntax for building-regulation writers and software developers
(2) Provision of computerized support to enable regulatory organizations to easily develop, check, and maintain these regulation representations
(3) Checking the sufficiency and implementability of the digital representations

The three steps described by Garrett et al. focus on the design of digital representations. It is a more detailed view of the first phase of Eastman et al., who describes the necessary steps for the whole checking process.

“One of the key criteria […] is to be independent of any specific model-checker software used to check the regulation compliance of building information models” [145].

Digital representation of regulatory rules

An important requirement of automated code checking is that it must represent the rules that a regulation or guideline contains in a form that lends itself well to computer processing.

Various software products exist for digitally checking building models, but as yet, none of them offer comprehensive functionalities for user-driven rule definition: “With the exception of SMC [Solibri Model Checker], none of the existing tools deals with the geometry and spatial operations that frequently adorn the BIM-based rules, especially in building codes” [49,186]. “While some model-checking software systems exist, they either require that their users possess good software-programming knowledge to configure them with rules of interest, or they are black boxes, and not configurable at all” [91]. “This ‘hard-coding’ of design standards into design programs is a major barrier to the general acceptance and evolution of computer-aided engineering, as it does not provide designers having to make professional judgments with the ability to view and understand the representations of the design standard on which the computations are based” [178].

When developing automated checking methods for building models, it is important to consider the user and thus the practicability of the method: “Countries such as Singapore, USA, and Australia have begun a new era that utilizes BIM for an automated, flawless administrative building permit process. However, most studies are focused on a script language-based result. This type of result has a high threshold for the user, who requires a building rule-checking process but has little understanding of the computer-based process and the programming language. This causes the rule-checking process to require professional human resources. Therefore, the above users always require a rule-checking process with computer-related experts” [187]. “Most of the existing approaches [are lacking,] because of the insufficient transparency and visibility of the processing steps for the user. Many methods focus too much on the automation of the checking process and do not consider the incorporation of the user and therefore the practical applicability” [98].

The unavailability of digital regulations is also an obstacle to the sustainable use of automated checking methods. “The encoding of norms into rules is currently a manual process. It is expected that the development of […] representations of legal documents is to be undertaken by the same government agencies responsible for authoring the legal documents in the first place” [106]. A common approach for translating plain text into digital representations is to use RASE technology, which appends regulation texts with tag markups (requirement, applies, select, exception) [90,188]. “Other attempts at automated model checking have taken the Natural Language Processing (NLP) approach and aim to automatically transform rules from human-readable specifications into programmatic executable code. While these methods have many benefits in terms of ease of use, there is usually far too much leniency in the written language, which makes it impossible to process automatically and accurately; as a result, these methods are fundamentally limited in their capacity to capture the requirements around compliance checking” [91]. To tackle these limitations, *Zhang & El-Gohary* used machine learning methods to examine different NLP approaches [189]. Despite the progress achieved, significant research still remains, in order to make the approach suitable for use with a large set of different standards.

Regardless of the method used to translate natural language into machine-interpretable language, various types of rules must be considered when checking building models. Solihin and Eastman [49] categorize the rule types as follows:

(1) Rules that require a single piece or small number of explicit data
(2) Rules that require simple derived attribute values
(3) Rules that require extended data structures
(4) Rules that require a “proof of solution”

This study focuses on rules of type 3.

Rule encoding approaches

The numerous approaches in the field of code compliance checking are summarized in [87,103]. *Charles* presents an approach to performing compliance checks using RDF [147] while *Xu & Cai* investigate how RDF, ontologies and SPARQL GIS-based data on utilities networks can be checked for compliance [190].

Bus et al describe the applications of Semantic Web methods for BIM checking [191,192] and *Zhang & El-Gohary* use Semantic Web as the basis for implementing their NLP approach [193,194]

Various investigations have focused on checking models with regard to safety aspects, for example safety on construction sites [148], in the context of fire protection regulations [98,195] or in connection with the structural design of buildings [196].

Both the interaction of the different actors in a construction project and the building model check itself is a gradual and thus process-based procedure. “The conventional compliance audit process is procedural in nature, which lend itself to automation. However, there are still roles in the process that are best played by human experts such as specifying what information to retrieve from which sources and how to process them. Machines excel in executing instructions efficiently and accurately and so should be given such a role to play in the process.” [106]

Visual programming languages

Myers investigated visual programming languages (VPL) as early as 1990 and declared them to be an interesting subject area, concluding that they “show promise for improving the programming process, especially for non-programmers” [197]. “The construction of programs is probably easier in VPLs than in textual languages” [198]. “A VPL […] describes a system of signs and rules on the syntactic and semantic level with the help of visual elements. Through the visual presentation of the elements, the language may be interpreted more quickly and easily” [99]. Green et al. “believe that in many respects VPLs offer substantial gains over conventional textual languages […]. Improvements in secondary notation, in editing and in searching will greatly raise their overall usability” [198]. *Catarci and Santucci* also conclude from their investigations that visual programming languages (in this case QBD) have advantages over classical query languages (in this case SQL), since users find the visual language more accessible and are therefore less bogged down by programming [199].

Preidel et al. are developing a query language called QL4BIM, which is available in both text-based and visual-language versions [99,102]. In [98,200,201], Preidel et al. are also developing a ‘visual code checking language’, which “is intended to perform compliance checks automatically or semi-automatically [and] increases the efficiency and quality of the overall process significantly.”

Ghannad et al. investigate the possibility of mapping the contents of regulations and standards on an xml basis as LegalRuleML notation using NLP methods and checking them using VPL. Currently, translation from LegalRuleML to VPL is done manually [202].

Ritter et al. examine the state of the art of visual programming languages in civil engineering and sum up the fields of application of visual programming languages as follows [203]:

- Inquiry languages
- Geometric modeling
- Knowledge-based design
- Design decision support
- Code checking
- Modeling of systems

The areas ‘Design Decision Support’ and ‘Code Checking’ are especially relevant for this study. “Information systems which are described by a visual language can be interpreted much faster and more easily by humans” [96]. “However, the state-of-the-art ACC systems cannot achieve full automation because they rely on the use of hard-coded, proprietary rules for representing regulatory requirements, which requires major manual effort in extracting regulatory information from textual regulatory documents and coding this information into a rule format” [193].

Numerous studies have been undertaken in the field of code compliance checking, and a recurring challenge is how to deal with the many regulations, norms and standards available only in human-readable form. ‘Natural language processing’ provides methods for translating these into a machine-interpretable form. The studies undertaken up to now have been limited to text-based regulations, which represent only a part of the content. However, the effort required to prepare texts for translation is currently still high. Instead of first writing regulations in their present form and then translating them, they should be designed in a machine-interpretable form from the outset.

The majority of the published approaches for code compliance checking focus on building designs and not on infrastructure projects. Many of the tools available on the market are ‘black box’ solutions, which makes it difficult for users to ascertain how checks are conducted and therefore how correct they are, in turn hampering their adoption. ‘Visual programming languages’ (VPLs), on the other hand, have the advantage that the checking process can be displayed graphically, which significantly increases readability. They also enable users to design inspection routines without any previous programming knowledge.

Both text-based and visual programming have advantages and disadvantages. This study aims to combine the advantages of both variants: comprehensibility, traceability, simplicity, automation, and adaptability. To this end, it examines the potential of a pre-existing notation taken from business process modeling to ascertain its applicability in the area of code compliance checking.

Building information models and the IFC standard

A prerequisite of automated code compliance checking is the availability of design information in a semantically rich representation, i.e. a building information model. Most of the existing checking systems make use of information provided in the Industry Foundation Classes (IFC) data format, which is a standardized vendor-neutral format for representing and exchanging geometric-semantic building information models.

The SEEBIM project, for example, investigated whether and how IFC models can be automatically enriched with additional information so that more extensive compliance checks can be performed. Predefined operators are available for rule creation, which in turn reduce programming complexity to the benefit of the user [204]. The suitability of the IFC data model for code compliance checking in the railway domain has not yet been investigated. Only recently has the IFC-Rail project published a first extension of the IFC data model to cover the railway domain [205].

Solihin et al [159] examine the quality criteria of IFC exchanges and conclude that there is “an urgent need to define robust and rigorous test criteria, processes and tools.” In this study, this is extended to include the technical checking of infrastructure models. *Zhang et al.* present a method for checking the conformance of IFC models using mvdXML [206].

4.2.2 BPMN: Business Process Model and Notation

VPLs are of particular interest when it comes to the interdisciplinary development (software development, civil engineering, and quality inspection) of automated inspection methods, due to their better comprehensibility. “In many enterprises, there is a need to facilitate smooth communication between business experts, software engineers and other people with technical knowledge. There are several methods of business knowledge representation, such as business rules or business process models. Both these representations can describe how the company works. Some issues, like constraints or detailed regulations, are better represented as rules, while others like procedures or workflows are better represented as process models” [207]. “The sequence of steps in a typical compliant design procedure can be represented as a series of activities, events, and sequence flows in a process model such as the open standard Business Process Model and Notation (BPMN)” [106,208].

BPMN is an international standard (ISO/IEC 19510) that is maintained by the Object Management Group [208]. *Dimyadi et al.* use BPMN to develop processes for the automated checking of building models [105–107]. In contrast to the work of Dimyadi et al. the study presented here focuses on railway projects and dedicated regulations, which are significantly more formalized than in the building domain. The BPMN-approach is additionally supplemented by the Decision Model and Notation (DMN), which will be in introduced in the next section.

The BPM notation provides a means of graphically representing processes in a formal manner. Various standardized node and edge elements are available, as illustrated in Figure 1.

Recker et al. describe BPMN as a “[..] structured, coherent and consistent way of understanding, documenting, modeling, analyzing, simulating, executing, and continuously changing end-to-end business processes and all involved resources in light of their contribution to business performance” [209]. And Janssens et al. state that “Business process management (BPM) and decision management (DM) are being used to improve the efficiency and effectiveness of organizations. Companies are interested in running effective and competitive processes, and use BPM to describe and improve these processes” [210].

In the AEC context, BPMN is often used to visualize the processes of model creation, collaboration and data exchange during planning (Information Delivery Manual or IDM) and larger development tasks, as shown in [211,212]. ISO 29481-1:2016 recommends the use of BPMN for the creation of IDMs [213], as implemented for example in [214,215]. The elements of BPMN used most frequently for creating IDMs are described in [216], which goes on to elaborate a proposal, based on the investigations, for which BPMN elements should be used for the creation of IDMs (see Figure 51). “The most frequently used BPMN elements for representing business processes in the AEC industry were sequence flow, pool, lane, task/activity (exclusive), gateway, and message flow, which are all basic BPMN modeling elements” [216]. “One important type of activity in a BPMN-compliant process model is the script task, which allows the embedding of computer scripts that convey user-specified instructions such as where to retrieve which information and how to use the collected information to perform specific calculations” [106]. Many “modeling software systems also assist the user in developing processes using graphical specifications. In particular, the option to use BPMN in the context of workflow management systems as an implementation language [...] is an important criterion for use in construction projects” [217].

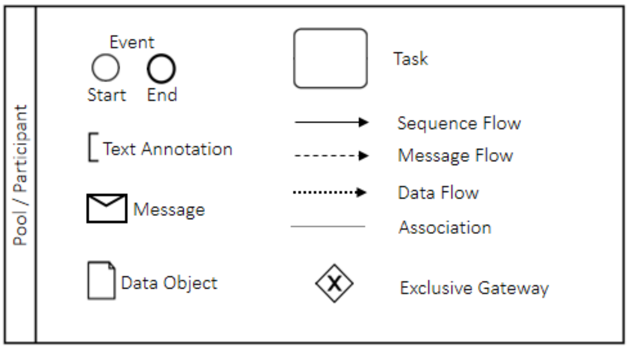


Figure 30: Proposal for the ‘Essential Subset’ of BPMN for IDM Development [216].

How the process itself can be subjected to a compliance check is explained in more detail in [218–220]. *Awad* presents a method of querying process diagrams using query language (BPMN-Q) to identify similar patterns in different process diagrams [221].

Recker states that “‘Classical’ process management applications such as documentation, redesign, continuous improvement and knowledge management dominate the application areas of BPMN, while more technical application areas such as software development, workflow management or process simulation are not (yet) widespread" [222]. This a significant gap, which this study aims to help fill.

In addition to visualizing a process, it is possible to execute it automatically once generated, using a workflow engine that performs the process step by step and in accordance with the modeled logic and can react to events during runtime as well as trigger events itself. Such an event might be a data input by the user, the calculation of mathematical formulae, a decision resulting from an if-then condition, or the execution of text-based source code. By virtue of its ability to automate the developed processes by means of a workflow engine, BPMN can be classified as belonging to the category of visual programming languages. BPMN has already found its way into the construction industry, in particular for IDM development, but there are now also initial approaches to using the method in the area of code compliance checking. This is promising because, besides graphical notation, it also enables the development of individual extensions using script tasks to address aspects not covered by the notation. A script task makes it possible to incorporate individual instructions, such as formulae, in the process, using script languages like Javascript or Groovy.

The utilization of BPMN for automating code compliance checking is something that is rarely addressed. Among the few published approaches, *Kog et al.* use Petri nets to check BPMN-based construction processes [223] and *Zolfagharian and Irizarry* use BPMN to visualize the process for the automated checking of construction site equipment models [224].

4.2.3 DMN: Decision Model and Notation

As discussed in the previous section, it is possible to represent processes visually by means of BPMN and execute them with the help of workflow engines. BPM notation's gateway elements can be used to map simple decisions (in if-then-else relationships), but if several criteria within these relationships need to be evaluated, the process quickly becomes confusing. To simplify this situation and to present decision options more clearly, the 'Decision Model and Notation' (DMN) was developed, version 1.3 of which was published in December 2019. "The primary goal of DMN is to provide a common notation that is readily understandable by all business users, from the business analysts needing to create initial decision requirements and then more detailed decision models, to the technical developers responsible for automating the decisions in processes, and finally, to the business people who will manage and monitor those decisions. DMN creates a standardized bridge for the gap between the business decision design and decision implementation. DMN notation is designed to be usable alongside the standard BPMN business process notation" [225]. "The separation of Decision Modeling from Business Process Modeling is a good principle, whatever the models, notations and languages are. It will enhance agility when changes are required, in reducing their impact, among them the risk of failure, and in increasing the resilience of the Information System" [226].

As early as 1969, *Fenves et al.* described a method of automating decision paths with decision tables [227]. *Huysmans et al.* compared several methods of presenting decision paths in a machine-readable form and validated them against their comprehensibility for users. “The results showed that, on the aspect of comprehensibility, decision tables provide significant advantages. For each part of the experiment, the respondents were able to answer the questions faster, more accurately and more confidently using decision tables than using any of the other representation formats” [228]. “The use of DMN for modeling the requirements for automated decision-making is similar to its use in modeling human decision-making, except that it is entirely prescriptive, rather than descriptive, and there is more emphasis on the detailed decision logic. For full automation of decisions, the decision logic must be complete, i.e., capable of providing a decision result for any possible set of values of the input data” [225].

The DMN is therefore a useful supplement to the BPMN that makes it possible to keep process representations clear, understandable and easily comprehensible. With the help of DMN, process representations can be reduced to their essential parts. Like workflow engines, so-called decision engines enable automated decisions to be made. By integrating DMN into BPMN, the decision engine can be triggered by the workflow engine. This study shows how DMN and BPMN are integrated to achieve a high level of automation of code compliance checking in railway design.

4.2.4 BIM in railway design and construction

BIM is being increasingly adopted in the railway domain worldwide. A number of national railway organizations have declared BIM as the obligatory way forward in improving quality in design and construction while reducing project costs and delays. To reach these goals, such organizations publish master-plans and/or define mandatory standards. These include China Railway [12,13], Korean railways [16], French railways [18], Swiss railways [19], Italian railways [20], Swedish, Danish, Norwegian and Finnish railways [22,23] and German railways, among others.

BIM has been mandatory for DB Station & Service AG projects since 2017 [9]. It is the first company within the Deutsche Bahn AG Group to issue standardized “Guidelines for the Application of the BIM Methodology” for the provision of model-based planning and construction services. While the guidelines also mention checking methods for the quality assurance of models, including collision detection, visual inspection and compliance with regulations, they do not describe any concrete implementation strategies. Most of the quality assurance criteria are organizational criteria, for instance, concerning adherence to deadlines. The quality assurance concept developed by the authors, on the other hand, defines strategies for implementing and deriving model-based quality assurance criteria [176]. To check the digital building model, it is necessary to scrutinize both the model creation process and the transfer of model data.

4.2.4.1 Geometric modeling

This paper examines methods for checking the compliance of 3D infrastructure models and the underlying route mapping with the respective guidelines. To begin with, it is necessary to differentiate between “two fundamentally different approaches to modeling the geometry of three-dimensional bodies:

Explicit modeling, which describes a volume in terms of its surface […]. Implicit modeling by contrast employs a sequence of construction steps to describe a volumetric body, and is therefore commonly termed a procedural approach” [62].

Borrmann et al. [62] outline the following procedures in detail:

- Explicit procedures
    - Boundary representation method
    - Triangulated surface description
- Implicit procedures
    - Constructive solid geometry
    - Extrusion and rotation processes
    - Parametric modeling
    - Freeform curves and surfaces

“Most of the software products used for infrastructure planning offer a drawing-oriented view – split into site plan, cross-section and elevation […]. This type of model is referred to as an implicit geometry description. […] With implicit models, the governing design parameters become significantly more accessible than with explicit models. An example is the objects and parameters that define alignment. Consequently, implicit models offer a better basis for checking these parameters against codes and guidelines” [176].

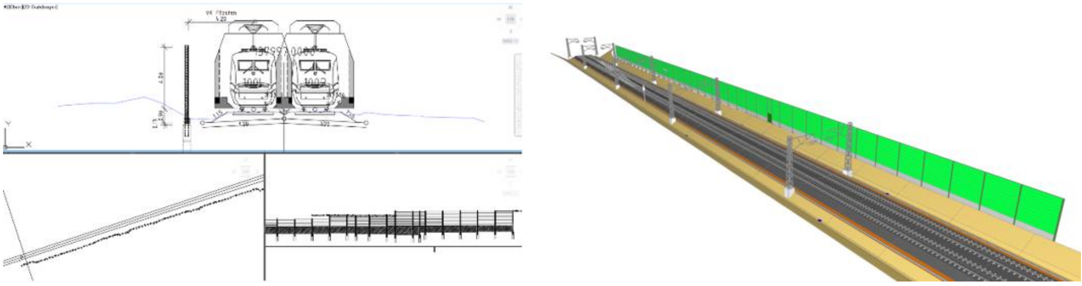


Figure 31: Comparison of implicit and volumetric 3D models: while implicit models (drawing-oriented view) are used during the design process, explicit models are used in the context of BIM-based analysis [176].

In addition to the pure parametric description of digital building models, some software systems also keep a history log of the individual modeling steps and enable the modification of individual steps, which is an improvement over the drawing-based method. “The […] important concept provided by parametric CAD systems is the explicitly available construction history. The system records each single construction operation […]. All operations are parameterized, for example, the height of an extrusion is an explicitly available parameter. The maintenance of the construction history stands in strong contrast to conventional systems which typically only store the result of the construction operations as an explicit boundary representation. The procedural approach provides the user of the system with the possibility to easily modify an existing model by going back in the construction history and adapting the corresponding parameter […]” [32]. “Explicit parameters of building elements are one of the merits of a building model and resolve difficulties in interpreting code checking” [87]. “[…] It is equally important to have efficient access to the building data as it is having good, transparent, and maintainable computable rules. Efficient access to the building data is important since rule checking potentially has to iteratively go through almost the entire data set with a large number of rules. […] Another critical issue that has been absent in the discussion so far is the support for geometry and spatial operations on the BIM data. Studies of building codes show that the majority of rules involve geometry and spatial operations” [186]. “A parametric modeling system will require careful engineering judgment and responsibilities in setting up the input and reviewing the output and a method to specify the requirements in an unambiguous way” [229].

While planning and modeling tools used for railway design employ implicit data models, they support the generation and export of explicit geometries using the boundary representation method or triangulated surface descriptions. This information can then be transferred using data exchange formats such as Industry Foundation Classes.

4.2.4.2 Industry Foundation Classes

The vendor-neutral data model Industry Foundation Classes (IFC) allows the high-quality exchange of geometric-semantic models. Over the past few years, it has been continually extended for use in the model-based data exchange of infrastructure facilities. The ‘Overall Architecture’ project and the IfcAlignment extension (also known as IFC4x1) first introduced a set of classes for the exchange of alignment axes and digital terrain models [230]. These initiatives aimed to provide basic concepts that are now supplemented by domain and subject-specific requirements. In the context of railway design, *Reifenhäuser et al.* have expressed the criticism that although horizontal and vertical parts of an alignment geometry can be mapped with IFC-Alignment classes, comprehensively representing a railway line also requires that cant is taken into account [231]. This shortcoming has since been recognized and has recently been corrected in the extension project *IfcRail.* Similarly to the completed IfcBridge extension project [232,233] in the context of bridge design, the IfcRail project deals with the domain-specific requirements of vendor-neutral data exchange for railway design.

Even though IfcRail development is still in progress, the IFC4x1 version of IFC instance models (the most recent final schema version) can already be used for numerous automated rule checks in railway design. For example, an alignment is described in the IFC data model by its horizontal segments in the xy-plane and the corresponding vertical segments in the projected coordinate system (s,z) [234]. In addition, it is possible to place *IfcReferent* objects at specified locations along an alignment axis and to attach user-defined property sets to them. The parameters in these property sets are not defined in the data schema and provide the modeler with a very flexible method for transferring project- or organization-specific attributes. The downside is that any flexibility in a data schema also requires specific processing of user-defined, non-standard properties must also be specifically addressed for import processes, which makes automatic interpretation considerably more difficult. Nevertheless, this ability to dynamically extend the schema with additional semantics is a useful feature of IFC, making additional rules testable.

In many parts of the IFC data model, it is possible to choose between different representations and dynamic semantic extensions. This flexibility, however, inevitably means that not every instance model can be used for every check routine, since the data cannot be extracted in a consistent form. To counteract this, the model to be checked must comply with additional specifications [49] that can be defined through the mechanism of a model view definition (MVD). An MVD defines a subset of the entire IFC data model and thus restricts, for example, the geometric presentations that may be used [235]. At the time of writing, there are no internationally defined and published MVDs that reflect the correct use of *IfcAlignment* instances or related concepts such as *IfcLinearPlacement* for the description of positions along an alignment element. Nevertheless, the schema definition contains so-called *concept templates*, which represent specific rules of a complete MVD and can be assembled in a modular fashion. An MVD for specifying the required input data can be elaborated using the *IfcAlignment Concept Template* and defining any additional necessary parameters.

The latest candidate standard extension is called IFC4x3 and includes the extension proposals of IfcBride, IfcRoad, IfcPortsAndWaterways and IfcRail projects [236].

4.3 The DB Netz AG guidelines

Due to the very stringent safety requirements that apply to rail facilities, numerous guidelines are in place that strictly regulate design as it relates to rail infrastructure facilities. While aesthetic aspects of the structure are important for the individual expression to the respective building designs, infrastructure facilities are on the whole characterized by a high degree of standardization both with components and geometric dimensions.

The study henceforth takes the guidelines of Deutsche Bahn AG as an example, but the principle can be extended to other countries with similar guidelines of their own. Comparable regulations also exist for road infrastructures.

The guidelines are presented in a clearly defined structure. In the context of code compliance checking with building information models, it is the construction engineering guidelines, in which most of the rules for design and construction are defined, that are the most interesting. The main groups are divided by trade into sub-groups. The guidelines examined in this study are as follows:

Table 8: Assignment of trades or sub-groups to guideline numbers.

Trade/sub-group	Guideline number
Network infrastructure technology design	800
Basics of superstructure	820
Earthworks and other geotechnical structures	836

Guideline Group 800 defines the parameters for track routing and sets out the basis for the entire track system. It includes definitions for switches and crossings as well as for track cross-sections and regulates the dependencies between individual objects. It also sets boundary conditions such as the design speed. Guideline Group 820 defines the design of the superstructure, such as the rail form and sleeper type, while Guideline Group 836 specifies the conditions relating to the required civil engineering measures, for example, earthworks, drainage, etc. These three groups set out the most relevant guidelines for the planning, construction and operation of a railway infrastructure and are therefore examined in detail in this study. Further guidelines exist that deal with technical equipment (e.g. control and safety equipment and overhead catenary lines) and structural engineering (e.g. bridges and tunnels) but are not considered here.

Particular attention must be paid to the modal verb formulations used in the guidelines, as they are used to indicate specifically whether the item in question is a requirement, permission, recommendation or possibility and capability. Table 9 shows the terms used and their implications based on the guidelines and DIN 820-2 [160,237,238]. As these are relevant for both planning and manual quality control, it is essential that they are used accordingly in model-based quality assurance.

Table 9: Meaning of auxiliary modal verbs in DB Netz AG guidelines according to [160,237,238].

Auxiliary modal verb	**Meaning**
Shall	Requirement
Shall not	
May	Permission
Need not	
Should	Recommendation
Should not	
Can	Possibility and capability
Cannot	

4.4 Rule representation using BPMN and DMN

Building information modeling is used increasingly in practice for designing buildings. The quality assurance concept presented in [176] has therefore been taken as the basis for the concept presented in this study. Our concept additionally considers the domain of ‘Construction Design’, which is concerned with checking digital building models during the design stages. An essential aspect here is the ability to perform a digital and machine-evaluable representation of the existing guidelines for automated model checking. “Case examples and requirements in guidelines [...] can be presented in many different ways, ranging from simple and clearly structured tables with limiting values to graphical representations or written descriptions” [98]. This also applies to the guidelines of Deutsche Bahn AG. For the purpose of the concept, the process of checking models against guidelines involves performing calculations and comparisons and making decisions. The process-based description of the guideline content is represented by BPMN and DMN. The following section begins by formulating the general requirements for automation before going on to describe the logic of translation into a machine-interpretable form. The translation is done manually, but supported by the RASE methodology [90], which is a semi-formal process for analyzing regulatory text in natural language and marking it up for further processing.

4.4.1 General requirements for automation

BPMN, as a node-edge model comparable to visual programming languages, provides a set of process elements to enable the representation of policy content or rules. Node elements include events (e.g. start, end), tasks (e.g. user tasks, script tasks, or DMN tasks), and gateways (e.g. exclusive, parallel) as shown in Figure 51. Edge elements (e.g. sequence flow) connect the respective nodes into a continuous process. In addition, it is possible to aggregate sub-processes into a higher-level process to enable the hierarchical organization of processes.

To facilitate automation, the following conditions apply when creating processes:

(1) Every process has a start and an end event.
(2) Events, tasks, and gateways are connected by flow elements.
(3) All tasks are addressed and initiated by a start event.
(4) All tasks are integrated into the process in such a way that they are connected to an end event.
(5) Gateways have at least two output edges, but there may be a default path.

4.4.2 Principles of process modeling

As discussed by *Preidel & Borrmann in* [98], regulations can be presented in various ways, for example as continuous text, tables, graphics, or formulae. This section describes the principles of process modeling and explains how guidelines are represented in this concept using BPMN and DMN.

Textual descriptions place a rule in an overall context and describe decision paths, individual parameters and entire parameter sets. They also contain embedded formulae as well as references to graphics, tables, attachments, etc. Typically, the superordinate process description is textual.

Graphics are used to clarify the content of a textual description and illustrate component interdependencies as well as parameters and boundary conditions that need to be considered. Because they can be part of an aggregated representation, graphics can describe both individual as well as multiple rules.

4.4.2.1 Translation of regulations

RASE-syntax [90] is used to support the translation of regulatory texts into workflow diagrams. The regulations are structured with the tags *requirement* <*r*> *or* <*R*>, *applicability* <*a*>, *select* <*s*> and *exception* <*e*>.

The following section illustrates the process of translating a rule using the example of the routing element ‘arc’ and a chosen radius. The resulting process and its individual steps are shown in Figure 33, along with the textual descriptions.

Guideline 800.0110 § 6 No. 3 defines the following requirement:

The arc radius (r) of a track curve is calculated by the formula [Equation 1] taking into account the speed (v), superelevation (u) and superelevation deficit (u_f) and shall not be greater than 25.000 m.

$$r = 11{,}8 \times \frac{v^2}{u + u_f}$$

Equation 1: Determination of the arc radius of a track curve according to Guideline 800.0110.

The marked rule following the RASE-syntax looks like this:

<R>The <a>arc radius (r) of a track curve is calculated by the formula <r>[Equation 1]</r> taking into account the speed (v), superelevation (u) and superelevation deficit (u_f) and shall <e>not be greater than 25.000 m</e>.</R>

The rule does not contain any selection.

4.4.2.2 Representation with BPMN and DMN

In principle, all notational elements described by ISO/IEC 19510 may be used, but in this study, the elements shown in Figure 32 (script task, gateway, DMN task) were found to be particularly suitable and therefore play a focal role.

Guideline specifications in the form of calculation formulae (requirements) are incorporated in the workflow by means of a script task (see Figure 32, number 1). The calculation formula is written in Javascript.

Both text and tables can describe decisions (select-tag in RASE) that can usually be resolved into simple if-then-else relationships. Such decisions can be represented in the process by a gateway or as a DMN task. If the result of the decision is described with formulae, representation by means of a gateway (see Figure 32, number 2) and the following script task is chosen, since DMN tasks do not accommodate formula-based evaluations. The if-condition is formulated using the flow elements assigned to the gateway. A gateway can also be used to compare individual parameters with the model independently of other boundary conditions.

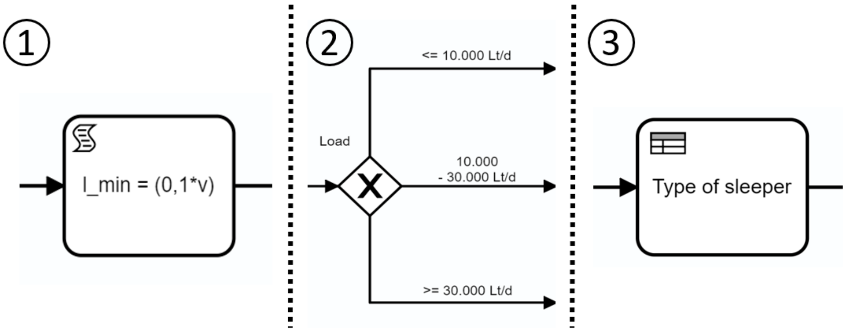


Figure 32: BPMN elements: 1 – Script task used to represent formulae; 2 – Exclusive gateway used to represent simple decisions/branches; 3 – DMN task used to represent complex decisions.

Remark to the published paper: The RASE-marking of the shown rule (Section 4.4.2.1) must be corrected to:

<R>The <a>arc radius (r) of a <s>track curve</s> is calculated by the formula <a>[Equation 1] taking into account the <a>speed (v), <a>superelevation (u) and <a>superelevation deficit (u_f) and shall <r>not be greater than 25.000 m</r>.</R>

The rule does not contain any exception.

A DMN task (see Figure 32, number 3) is used if the decision result depends on several input variables (if-conditions) and is expressed as a static value (no formula). When creating a decision table, the necessary input variables and results must be defined. Table 10 shows a schematic decision table. Compared with a succession of numerous gateways, the tabular representation of several decision paths is both clear and easily understandable. There is no limit to the number of input or output variables or rules to be considered.

Table 10: Schema of a DMN decision table, in which any number of rules can be created. To generate a decision, the necessary input and output variables, including associated values, need to be defined.

Rule number	**Input**			**Output**		
	Input variable $\boldsymbol{i_1}$	*Input variable* $\boldsymbol{i_2}$	*Input variable* $\boldsymbol{i_m}$	*Output variable* $\boldsymbol{o_1}$	*Output variable* $\boldsymbol{o_2}$	*Output variable* $\boldsymbol{o_z}$
1	*value* $i_{1,1}$	*value* $i_{1,2}$	*value* $i_{1,m}$	*value* $o_{1,1}$	*value* $o_{1,2}$	*value* $o_{1,z}$
2	*value* $i_{2,1}$	*value* $i_{2,2}$	*value* $i_{2,m}$	*value* $o_{2,1}$	*value* $o_{2,2}$	*value* $o_{2,z}$
3	*value* $i_{3,1}$	*value* $i_{3,2}$	*value* $i_{3,m}$	*value* $o_{3,1}$	*value* $o_{3,2}$	*value* $o_{3,z}$
n	*value* $i_{n,1}$	*value* $i_{n,2}$	*value* $i_{n,m}$	*value* $o_{n,1}$	*value* $o_{n,2}$	*value* $o_{n,z}$

In this way, rules are broken down into the smallest constituent units so that the process representation is comprehensible and can be easily followed. Consequently, complex calculations or observations are divided into multiple tasks. To ensure that the method is accessible and comprehensible to users without programming knowledge, only simple formula and variable definitions are used in the process. Textual descriptions of individual elements and additional text annotations can be used to provide explanations of the process modules.

Mapping into BPMN is described in this section on the basis of a marked paragraph, following the RASE-syntax described in Section 4.4.2.1 (compare also Figure 4). The paragraph of the guideline begins with a description of the input variables (velocity, superelevation, superelevation deficit) necessary for calculating the arc radius and then describes their interdependency in terms of a formula. Therefore, to enable automated checking of the digital building model, it is essential that the necessary input variables are made available for processing (Step 1). Alongside the quantities declared in the guideline, the modeled radius must also be passed as a parameter. The process can then be started (Step 2). It is necessary to confirm that the model data will be checked by the appropriate rules, e.g. that an arc will be checked by rules dealing with arc information. This is taken care of by a gateway in Step 3 which represents the RASE-tag applicability. The radius calculation formula according to the guideline is represented by a script task (Step 4), and the result of the calculation is stored as an independent variable. The model's compliance with the guideline is checked by comparing the modeled radius with the radius calculated according to the guidelines. This takes the form of a simple if-then condition in the form of a gateway (Step 5), whereby the two sequence flows contain the corresponding decision logic for the subsequent process path. The flows also contain the exception to the rule. Depending on which path is automatically selected, the check result is set (Step 6) to either ‘passed’ or ‘failed’. At the end of the process, the result is returned to the user (Step 7).

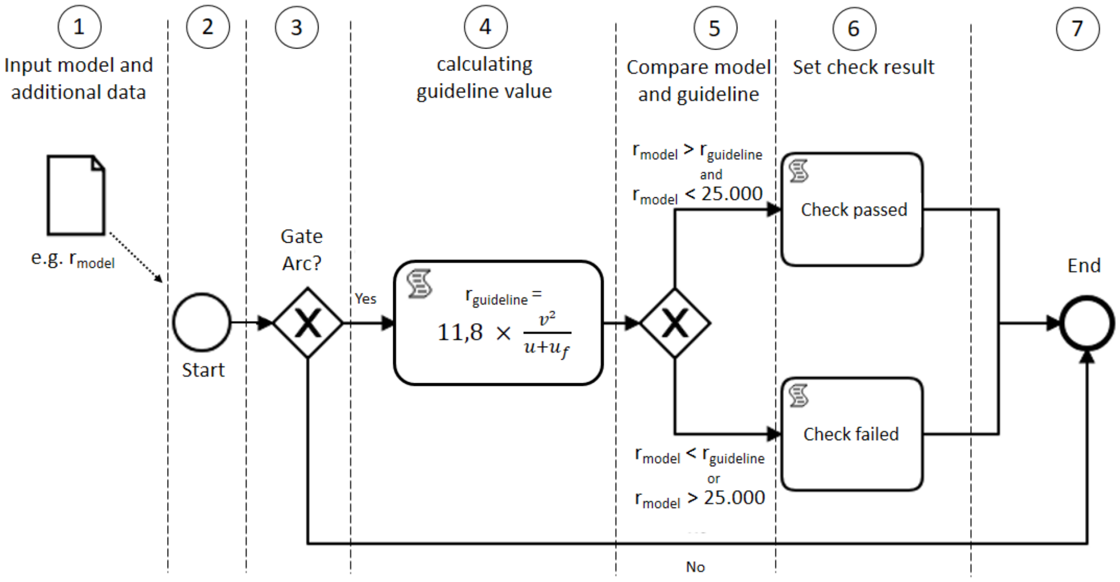


Figure 33: Process definition of a simple rule: Guideline 800.0110 – calculation of the arc radius and validation of model data.

From the example described, it is possible to formulate generally valid steps for the creation of a process:

(1) The model data to be checked is transferred to the workflow engine. Information that is not defined in the model but is essential for correct execution of the process is defined by the user and passed to the workflow engine together with the model data to be checked for compliance.

(2) In a pre-processing stage, the data is prepared for the workflow (transfer to variables, adaptation of dot-comma notation, etc.) and passed to the start of the checking process.

(3) The workflow engine selects objects and parameters from the submitted data set to be checked for compliance. If the workflow is not usable for checking submitted objects or parameters, the process is skipped.

(4) The workflow engine determines the target parameters based on the guidelines for the defined boundary conditions of the model and other information.

(5) The workflow engine compares the target parameters with the transferred model data (actual parameters).

(6) The workflow engine defines whether the check has passed or failed.

(7) The checking process is terminated, and the results are made available to the user in the form of a report.

4.5 Evaluating guidelines for implementation with BPMN

Studies of norms and standards in England and Wales have shown that 20% of the rule sets examined are declarative and thus directly computer-interpretable, while 47% need additional human intervention to make them computer-readable. In 33% of the cases examined, automation is not possible [239].

The extent to which models can be checked for compliance with the guidelines of Deutsche Bahn AG using BPMN and DMN was checked using Guideline Groups 800, 820 and 836 and analyzed for a total of 943 rule sets. Of all the rules analyzed, 486 (52 %) can be generally classified as automatable. A rule is classified as automatable if all of the following criteria apply (necessary conditions):

(1) All necessary input variables have been described.
(2) Boundary conditions/decision paths to be considered have been described.
(3) All required output variables have been described.

To be able to automate the content of guidelines, the rules must have a deterministic description. Policy contents are classified as non-automatable when at least one of the three criteria is not met or the contents aim to describe the readability and structure of the policy.

As described in Section 5.2.6, BIM models in infrastructure planning are created with the aid of parameters (e.g. length, height, width of objects) using an implicit data model. Analysis of the guidelines includes examining the extent to which guidelines describe the parameters that are required for modeling. A total of 460 rules (49%) were classified as ‘parameter-oriented’.

A total of 437 rules (46%) were classified as ‘BPMN implementable’, while 3% of the rules (32 in absolute terms) were evaluated as only implementable by means of higher-level quality tests, such as simulations.

A comparison of the evaluation results at policy level (see Table 11) reveals that Policy Group 800 in particular is predominantly parameter-oriented (82%), so here, models can be checked to a great extent using BPMN (75%). The rules contained in Guideline Groups 820 (38%) and 836 (39%) are far less parameter-oriented, and consequently have significantly lower potential for implementation with BPMN (37% and 39%, respectively).

Overall, the potential for implementing automated rule checking by means of BPMN was deemed sufficiently high (46% of the analyzed rules), which is a valuable basis on which to continue with the method, despite the unequal distribution across the individual guidelines.

Table 11: Evaluation of feasibility of Guideline Groups 800, 820 and 836 for checking models for guideline compliance.

Guideline group	Number of paragraphs	Automatable?	Parameter-oriented?	Realizable with BPMN?	Simulation necessary?
800	223	187 (84%)	182 (82%)	168 (75%)	19 (9%)
820	203	85 (42%)	80 (39%)	80 (39%)	1 (0%)
836	517	214 (41%)	198 (38%)	189 (37%)	12 (2%)

Over the course of the detailed analysis, the examined rules were divided into 12 different classes. These rule classes are defined in Table 12. Figure 34 shows a graphical representation of the classes 2, 4 and 7. Rule classes are mainly defined by their typology. In the case of distance definitions, the rule classes are subdivided, depending on the influence they have on direction in a 3D-coordinate system.

Table 12: Definition of rule classes and their respective descriptions.

No.	Rule class	Description
1	Not relevant for modeling	Specifications that have no influence on the modeling of buildings, e.g. procedural processes in communications with the client or specifications for document structures such as explanatory reports.
2	Distance definitions: transverse direction	Definitions (horizontal distances) that can be derived from the cross-section of the building model (see Figure 34).
3	Distance definitions: position	Site plan view in the sense of a 3-panel projection: distances between elements.
4	Distance definitions: height	Definitions (vertical distances) that can be derived from the cross-section of the building model (see Figure 34).
5	Component definitions	Specifications determining the component or component type to be used.
6	Definitions for alignment elements	Definitions that define parameters for vertical and horizontal alignment elements (straight line, arc, transition arc).
7	Slope definitions	Definitions (slope distances) that can be derived from the cross-section of the building model (see Figure 34).
8	Directional definitions: Longitudinal direction	Orientations of components in the layout plan.
9	Clearance tests	Rules for ensuring clearance along railway lines.
10	Other definitions	Rules that do not correspond to the other rule classes but still have an impact on the building model.
11	Construction process	Rules that define temporal components of a construction site.
12	Dimensioning specification	Rules for dimensioning components, such as hydraulic calculations.

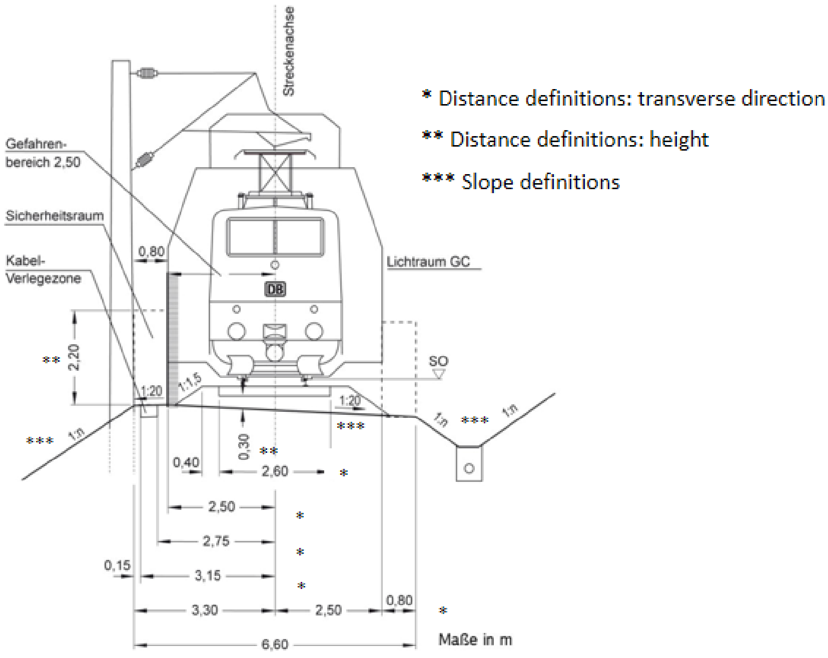


Figure 34: Standard cross section of a single-line railway alignment incl. corresponding distances (drawing adapted from [240]*).*

Table 13 shows the detailed evaluation of the analysis. No distinction is made with respect to the underlying guideline; rather, the results shown are cumulated for the examined guidelines. Column 2 of Table 13 shows the number of evaluated rules. The large proportion of rules classified as 'not relevant for modeling' (33%) is striking, and consequently column 3 shows the distribution with rule class 1 factored out. Column 3 shows the number of rules per rule class that can be implemented using BPMN/DMN. When the rules not relevant to model creation are factored out, the proportion of rules that can be implemented using BPMN increases from 46% of all those reviewed to 68% of those in rule classes 2 to 12. The three most frequently occurring classes that can be implemented with BPMN are 2, 4 and 5.

Table 13: Detailed evaluation of the guideline analysis according to the rule classes defined in Table 5. The top three rule classes are highlighted.

1		2		3		4	
Rule class		**Number of rule sets examined**		**Ratio convertible with BPMN/ DMN, without rule class 1**		**Ratio not convertible with BPMN/ DMN, without rule class 1**	
	Total	943	(100%)	432	(100%)	202	(100%)
1	Not relevant for modeling	309	(33%)	-	-	-	-
2	Distance definitions: transverse direction	77	(8%)	74	(17%)	3	(1.5%)
3	Distance definitions: position	46	(5%)	43	(10%)	3	(1.5%)
4	Distance definitions: height	69	(7%)	69	(16%)	0	(0%)
5	Component definitions	166	(18%)	141	(33%)	25	(12.5%)
6	Definitions for alignment elements	10	(1%)	10	(2%)	0	(0%)
7	Slope definitions	33	(3%)	32	(7%)	1	(0.5%)
8	Directional definitions: Longitudinal direction	9	(1%)	8	(2%)	1	(0.5%)
9	Clearance tests	24	(3%)	3	(1%)	21	(10.5%)
10	Other definitions	72	(8%)	25	(6%)	47	(23%)
11	Construction process	1	(0%)	1	(0%)	0	(0%)
12	Dimensioning specification	127	(13%)	26	(6%)	101	(50%)

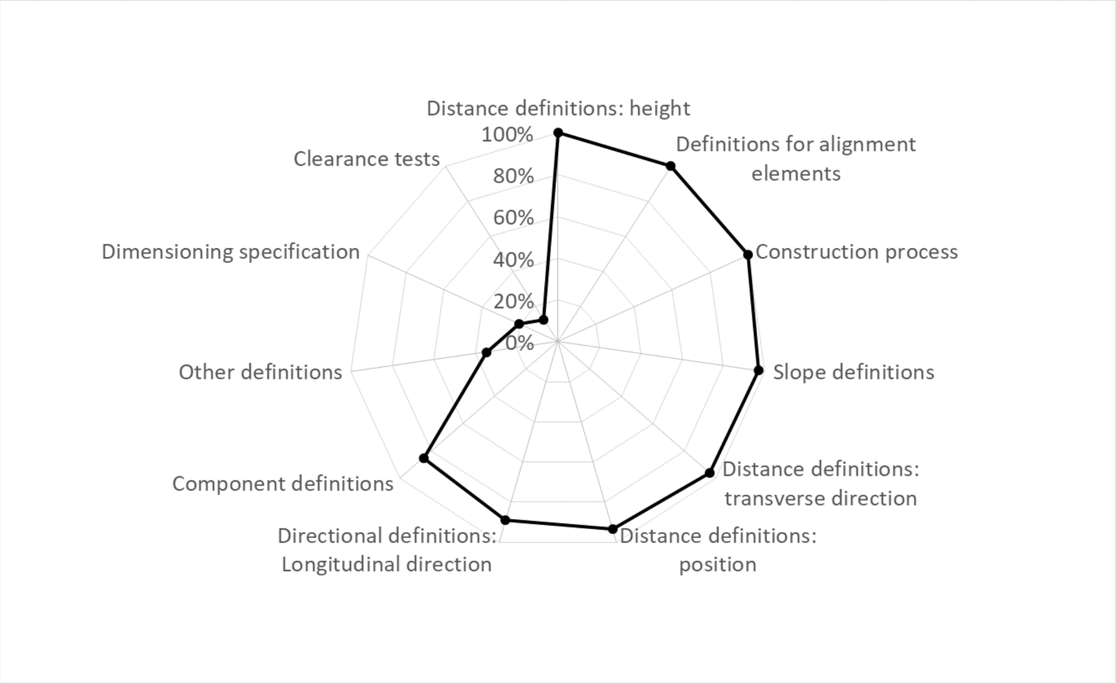
Column 4 shows the ratio of rules that *cannot* be implemented with BPMN and DMN. The rule classes ‘Dimensioning specification’, ‘Other definitions’, ‘Component definitions’ and ‘Clearance tests’ are the four most common ones whose rules cannot be implemented with BPMN and DMN. These rules or rule classes account for 30.6% of all rules relevant to the model. Reasons for being classified as ‘not implementable with BPMN and DMN’ are:

- Rules and their descriptions do not fulfill the necessary conditions for automation
- Sophisticated software for structural analysis or simulation is necessary
- Clearance tests are better performed as clash detection

The remaining rule classes thus account for 1.3 %. The rule class ‘Component definitions’ is an interesting case, as it is also the class whose rules can be automated most often with BPMN and DMN. This is due to the comparatively large number of rules that this class contains.

Diagram 4 shows the relationship between the total number of rules checked and the rules that can be implemented with BPMN and DMN. With a rate of between 12% and 35%, the rule classes 'Dimensioning specification', 'Other definitions', and 'Clearance tests' are much less frequently automatically verifiable with BPMN and DMN than the other classes, whose rates range between 85% and 100%.

Diagram 4: Ratio of 'total' rules to 'rules implementable with BPMN/DMN'.



The following section considers the three rule classes 'Component definitions', 'Distance definitions: transverse direction', and 'Distance definitions: height' in more detail, as they contain the most frequently occurring rules. These are discussed in case studies. A further example highlights the importance of vertical and horizontal alignment in infrastructure planning.

4.6 Case studies

4.6.1 Software configuration

Various software products were used to develop and validate the case studies detailed in according to BPM notation (see also Figure 35).

To automate the process of checking the compliance of model data with guidelines, both models and guidelines need to be available in digital form. ProVI [162] software was used to generate the models for checking. For the purpose of alignment, the model data was transferred using IFC. Model data that cannot be represented in IFC schema version 4x1 is transferred in Comma-Separated Value (CSV) format, implementing a table structure listing stations (rows) and the corresponding parameters of the model (columns). For example, with a railway superstructure, the exported CSV will include a list of rail types, sleeper types and distances between sleepers per station. The advantage of the CSV format is that it is the most straightforward way of exporting the parameters of the implicit data model from ProVI. However, as this representation is not vendor-neutral, it is strongly recommended that the IFC schema is used as soon as it is extended to cover railway designs.

As discussed in the previous Sections, BPMN and DMN were used to implement automation. Camunda Modeler [241] was used to model the guideline contents in these notations. The translation and representation of the guidelines as BPMN and DMN were performed using the rules described in Section 4.4.2. In addition to Camunda Modeler, Camunda Community Platform was also used as a workflow engine. Together with the supplied database, they represent the back end.

The IFC and CSV data were evaluated using programs written in Javascript. A front end was developed to transfer the model data to the workflow engine using web technology (web page: HTML, CSS, Javascript) and display the results of the check accordingly. A Rest API was used to facilitate data exchange with the back end.

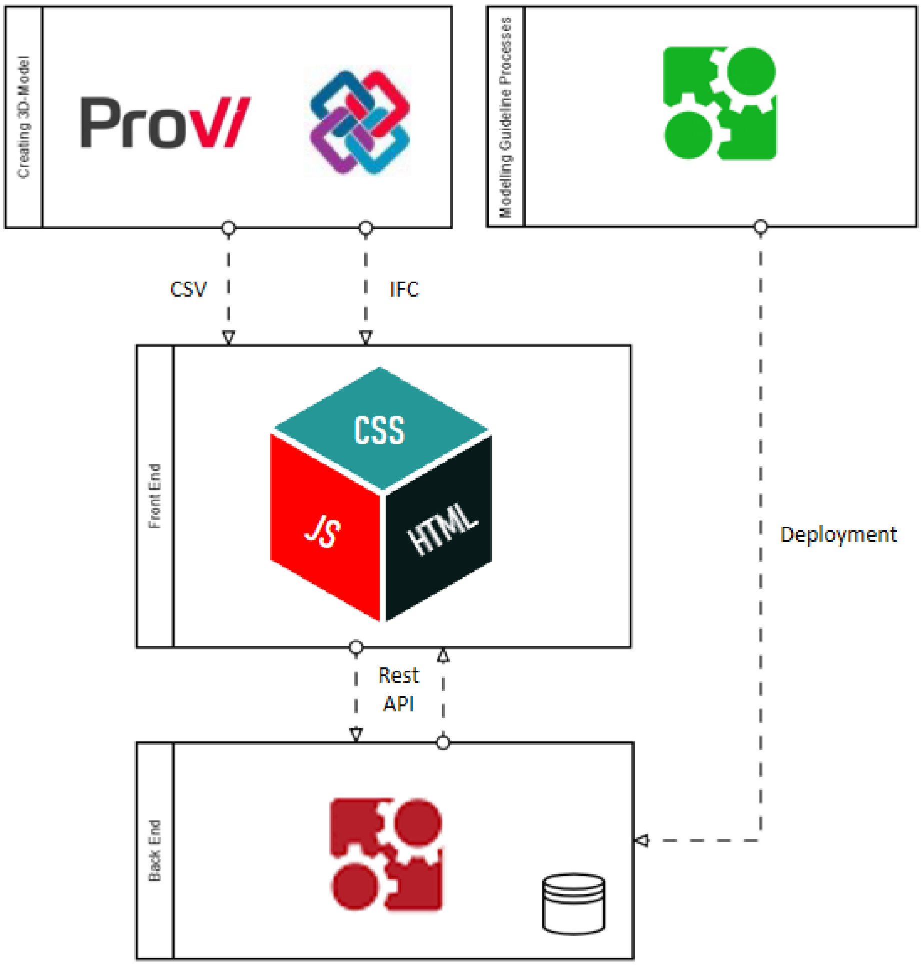


Figure 35: Software configuration (representation according to BPMN).

For the purpose of model preparation, in addition to the technical mapping of the rules and regulations in BPMNs, an information extractor was used to extract the required input data from an IFC model and make it available to the downstream checking process. Only IFC 4x1 models were used in the prototype extractor, since this version – unlike its successors – had already attained ‘final standard’ status. The IfcInfra proposals collected in IFC4x3, by comparison, are tested in a validation phase before granted as the next “Final Standard” version of IFC. As a result, the example models used do not contain all the required parameters available from the proprietary database of the BIM route modeling tool ProVI.

4.6.2 Case study: Alignment

Given the importance of alignment in the planning of infrastructure facilities, the following section describes the process of checking alignment elements for their compliance with the guidelines. The checks relate to Rule Class 6 ‘Definitions for alignment elements’. Figure 36 shows the elements that occur along an alignment: straight lines (black), transition arcs (red), and circular arcs (green).

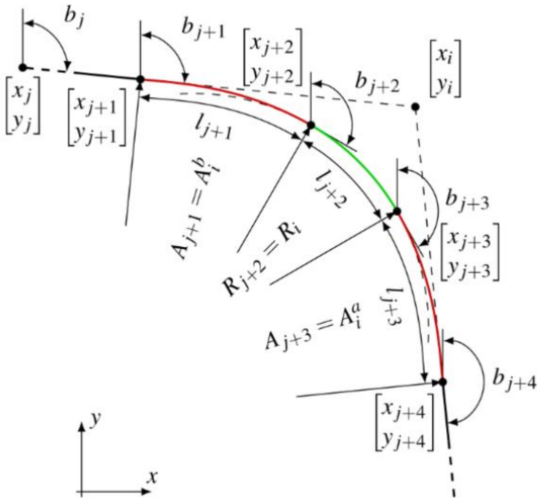


Figure 36: Different types of segments along a horizontal alignment: straight elements in black, transition curves in red, circular arc in green [234].

The description of a track or route alignment (three-dimensional space curve) with IFC 4x1 and used for verifying the here presented approach. Table 14 provides an overview of the parameters that are either available or can be derived in the authoring software or in the IFC model. It shows only those parameters that are relevant for the subsequent analysis. Up to version 4x1, no specifications for superelevation and velocity exist in the IFC schema. The importance of this information is apparent in the checking routines described below.

Table 14: Comparison of available information in the ProVI authoring software and the neutral data format IFC 4x1. While all information in ProVI is explicitly available, some information in IFC can only be derived through additional calculation. No information on superelevation or velocity is defined in the data schema.

Parameter / characteristic value	**ProVI**	**IFC 4x1 – explicitly defined**	**IFC 4x1 – implicitly derivable**
Entry station of an element	✓		✓
Element type, with exact subtype for transition curves	✓	✓	
Element length	✓	✓	
Radius for circular arcs	✓	✓	
Superelevation	✓	×	×
Superelevation deficit	✓	×	×
Velocity	✓	×	only if velocity is the same for all segments

Guideline 800.0110 specifies that the radius of a circular arc should be in the range 150 m to 25,000 m [172]. Figure 37 presents the logic employed for checking arc radii according to Guideline 800.0110. Since an alignment can consist of different element types (see Figure 36), the first step (1) is to verify that the element to be checked is an arc. A BPMN exclusive gateway is used for this purpose. If the element to be checked is a straight line or a transition arc, the process is terminated immediately. The directional description of the radii is specified in the alignment modeling software as positive (right-curved) or negative (left-curved). For this reason, the second step (2) calculates the absolute value of the radius to be checked, using a script task containing the following code:

Code 1: Calculating the absolute value of an arc radius in Groovy notation [242]*. The code is integrated into the process as a script task.*

$$Radius = Math.abs(Radius)$$

Now it is possible to check whether the radius is within the range 150 m to 25,000 m, as defined in the guideline. Here, too, an exclusive gateway is used, returning ‘check passed’ when true, or ‘check failed’ when false.

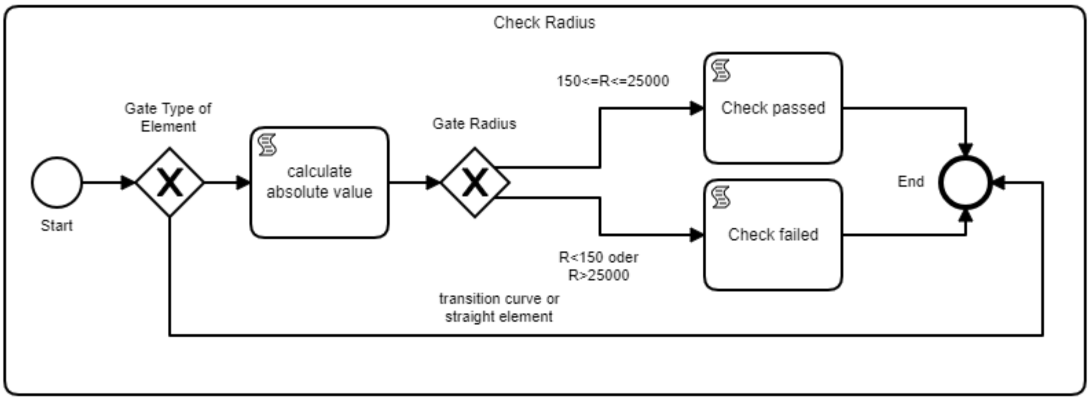


Figure 37: Process flow: Check the radius of an arc element of an alignment, represented as a BPMN process.

According to Guideline 800.0110 § 6 No. 2, the regular and minimum length of track curves and straight lines must be determined according to the details given in Table 15. Different formulae for determining the minimum length of straight lines and circular arcs apply, depending on the velocity. For the regular length, a single formula not subdivided into different velocity groups applies.

Table 15: Planning values for the lengths of track curves and straight lines according to [172]*, grouped by velocity classes, showing formulae for calculating minimum length and regular length (with no differentiation by velocity).*

Minimum length	
$v \leq 70$ km/h	$l_{min} \geq 0.10 \times v$ [m]
$70 \leq v \leq 100$ km/h	$l_{min} \geq 0.15 \times v$ [m]
$v > 100$ km/h	$l_{min} \geq 0.20 \times v$ [m]

Regular length
$l_{reg} \geq 0.40 \times v$ [m]

The checking process is shown in Figure 38. As in the process described above, the first step is to determine whether the element is a straight line or an arc (Gateway 1). If the element to be checked is a transition arc, the checking process is terminated without any further steps. If the element is a straight line or an arc, the next process module – a parallel gateway – is executed. This enables the process to be split into two or more paths, so that both the regular length and the minimum length can be determined within a process run according to the formulae specified in Table 15. An exclusive gateway is used to apply the respective formula to the corresponding velocity. The calculation of the minimum or control length is carried out using different script tasks. The calculated target value is then compared against the model value using an exclusive gateway in which the status of the 'regular length' is set to 'check passed' if the condition

$$l_{Model} \geq l_{regular}$$

is fulfilled. The status of the 'minimum length' check is set to 'check passed' if the following condition applies:

$$l_{Model} \geq l_{min}$$

Otherwise, the status of the checks is set to 'check failed'.

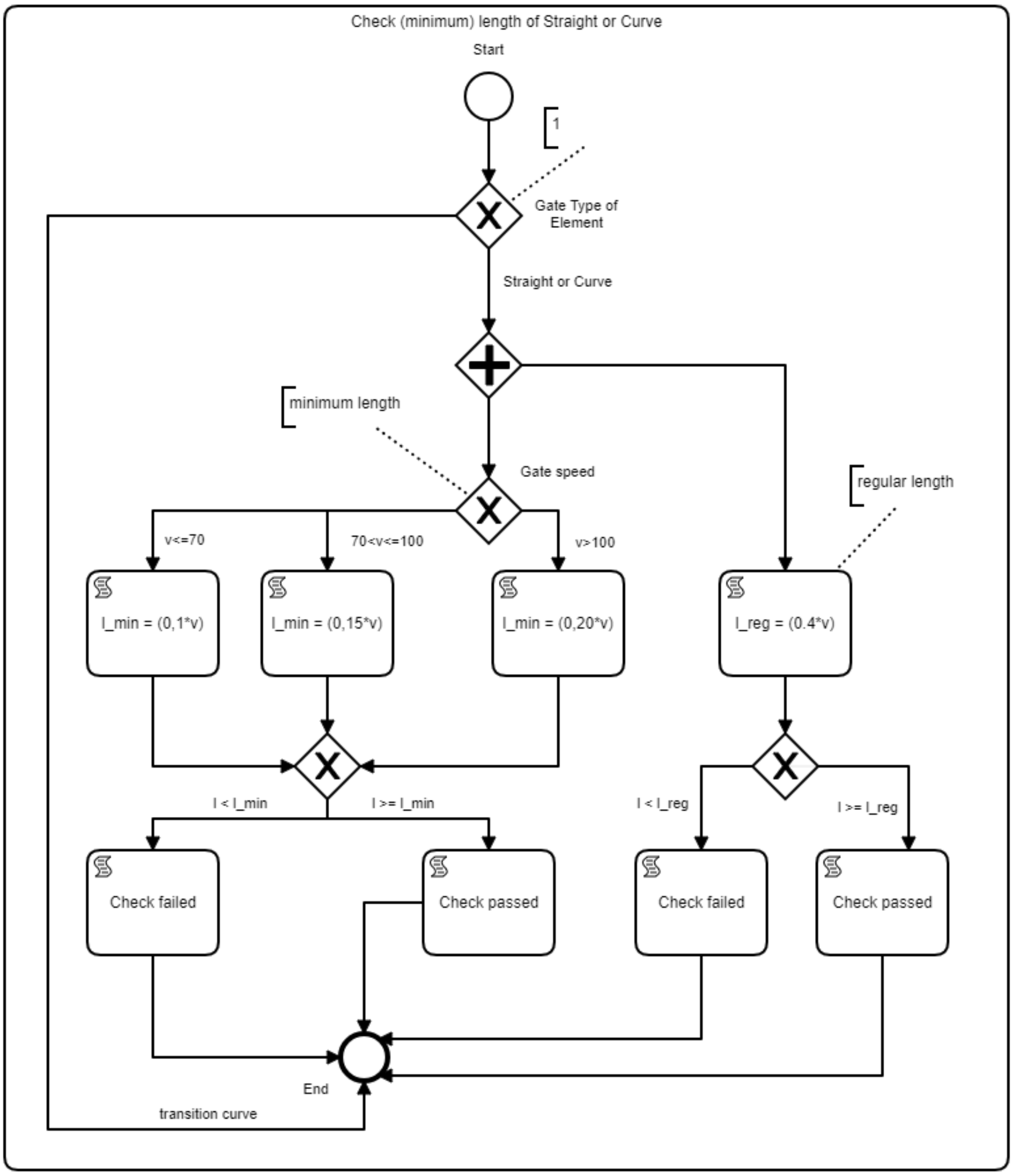


Figure 38: Process: Check the regular and minimum lengths of straight and curved arc elements of an alignment represented as a BPMN process.

In addition to the minimum lengths of straight lines and curved arcs, the guideline also describes the minimum length of transition arcs. The guideline describes three different types of transition curves: clothoid, Bloss-type, and s-shaped. The minimum length again depends on the velocity (v) and the difference of the superelevation deficits of two successive elements (Δu_f). In our case here, it is assumed that

$$\Delta u_f = u_f$$

Table 16 shows the formulae needed to determine the minimum length of the transition curves according to [172].

Table 16: Minimum length of transition curves according to [172].

	Transition curve	
Clothoid	**Bloss-type**	**s-shaped**
$\min l_U = \frac{4 \times v \times \Delta u_f}{1000}$	$\min l_{UB} = \frac{4{,}5 \times v \times \Delta u_f}{1000}$	$\min l_{US} = \frac{6 \times v \times \Delta u_f}{1000}$

The respective BPMN process is shown in Figure 39. Here, too, the first step is to determine whether the element type to be checked is a transition curve. If the check is negative, the process is terminated without any further steps. The second gateway splits the process based on the transition curve type (clothoid, Bloss, s-shaped) and passes the process to the respective script task, which then calculates the required minimum length using the relevant formula according to [172]. The third gateway then checks the result against the model and returns either 'check passed' or 'check failed'.

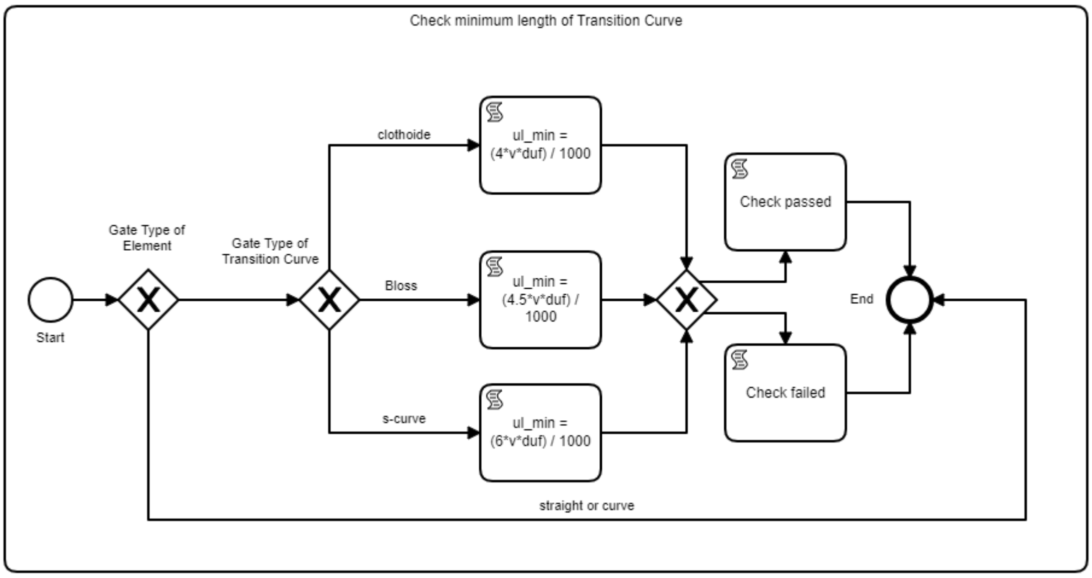


Figure 39: Process: Check the minimum length of transition curves represented as a BPMN process.

In line with the four model checking phases of Eastman et al [87], the time needed for checking data manually is compared to that of automated checking. The first phase describes the steps needed to interpret and represent the rules given by the regulations. For the manual working process, the auditor needs due training to be able to check designs correctly. Since it is a subjective process, it is not possible to measure the training time needed. The comparison therefore does not take this phase into consideration. The time required for the manual process in phases two to four is approximated. The time needed to create a process definition for the automated checking process depends on the level of difficulty. A simple rule, as shown in Figure 37, can be created in two days, including the time needed for researching the guideline, creating the BPMN/ DMN, and writing the model analyzer for extracting the parameters. An axis of a total length of 9.5 km is used as test data. In total, the axis contains 45 elements (arc, straight, transition curve) and 259 parameters that describe them. Table 17 presents the results of the measurements. The automated checking process takes 20 seconds (measured) to perform model preparation and rule execution. It is assumed that doing the same manually will take 20 min. Because the tool creates the report automatically, the results have to be validated and the report has to be sent to the designer. It is assumed that this phase will take 5 to 10 min. Following the manual method, the auditor also has to write a report, which takes approximately 15 min. In total, the ratio between manual and automated checking is approximately 24 %.

Table 17: Comparison of manual and automated checking and the time needed to check the axis test data

Phase	Description	Time needed for manual checking (approximation)	Time needed for automated checking	Ratio
1	Interpretation and logical representation of rules	-	-	
2	Building model preparation	5 min	10 sec	
3	Rule execution	10 min	10 sec	
4	Rule checking report	15 min	7 min	
	Total	30 min	7 min 20 sec	24%

In addition to the checking routines described above, further processes can be derived from the specifications of Guideline 800.0110. These include:

- Determining and checking the exact radius according to §6 No. 3
- Determining and checking necessary track extensions according to §6 No. 5
- Checking track curve radii in platform areas according to §6 No. 7
- Determining and checking design values for superelevation and superelevation deficit according to §7
- Determining and checking design values for superelevation ramps according to §8
- Determining and checking longitudinal inclination and changes in inclination in accordance with §10, including the design values for fillet radii and intermediate straight lines

The list shows that there is much greater potential for automating the checking of alignment elements than just the processes shown above. There are many more possibilities for modeling checking processes in the context of alignment elements.

4.6.3 Case study: Superstructure

Guideline 820 (“Basics of Superstructure”) summarizes the stipulations of general and overarching importance that govern the design, construction and maintenance of the railway superstructure [8]. This guideline is for use “within the area of DB Netz AG” for all railway superstructures up to a permissible velocity of v = 300 km/h” [8]. The following examples refer specifically to Guideline 820.2010 “Standard specifications for ballasted track for tracks and switches”. These serve as a basis for describing and developing checking processes for the rule classes ‘Component definitions’ and ‘Distance definitions: Height’.

Guideline 820.2010 “governs the application areas of superstructure components in tracks and switches as well as the dimensions of the ballast bed cross-section according to technical/economic aspects. In the following, the term ‘superstructure components’ is understood to comprise rails, sleepers, rail fastenings, sub-ballast mats, insulated joints, ballast and switch components” [8]. Sub-Clause 1 Sentence 3 stipulates that the guideline applies “in the area of DB Netz AG in tracks and switches travelled on with wheelset loads of up to 22.5 t” [8].

As such, the area of application of the guideline is clearly defined and the applicability of the guideline can be summarized as follows:

(1) Only valid for ballasted track
(2) Only valid for wheelset loads up to 22.5 ton
(3) Only valid for the superstructure components defined above

The corresponding process logic checks conditions 1 and 2 at the beginning of each run (see Figure 40) using two successive exclusive gateways: the first queries whether the model is a “ballasted track” and the second whether the wheelset load is less than 22.5 t. In the current process, the user enters this data, but if the data is stored in the model, it can also be used for the decision. If the conditions are met, the system continues with the sub-process of the respective superstructure components. In Figure 40, the sub-process is shown in a 'collapsed' state but will be explained in more detail shortly.

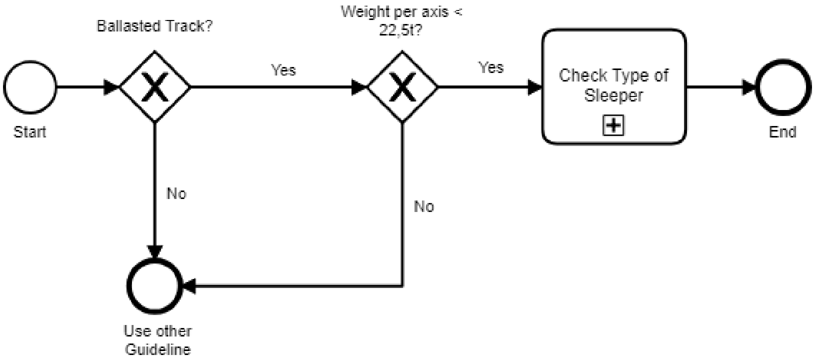


Figure 40: Basic data check to ascertain applicability of the guideline: Does the model data meet the requirements of ballasted track and axle load < 22.5 t (currently inputted manually by the user)?

The following parameters and characteristic values are available in the modeling software for station-wise evaluation:

- Station
- Rail form
- Sleeper type
- Sleeper spacing
- Bedding thickness
- Ballast shoulder

The definition of the superstructure can be refined in the modeling software using additional parameters and definitions, opening up further possibilities for automated checking (that are not described here in detail).

The guideline defines permissible superstructure components according to different parameters. "The superstructure components are determined according to the track load per day (Lt/d) and the maximum speed according to the List of Permissible Speeds (Hg VzG) or the local permissible speeds […]" [8].

The guideline is supplemented by various appendices that define the superstructure components and the respective dependencies. The process checks the sleeper type according to the logic of the guideline (see Figure 41) using an exclusive gateway that selects the respective appendix of the guideline depending on the track load. There are three categories for track loading:

(1) ≤ 10.000 Lt/d
(2) > 10,000 and < 30,000 Lt/d
(3) ≥ 30.000 Lt/d

The respective guideline appendix is integrated in the review process as a decision table in the "Decision Model and Notation" (DMN) standard. Table 18 shows the content of this decision table in the original format.

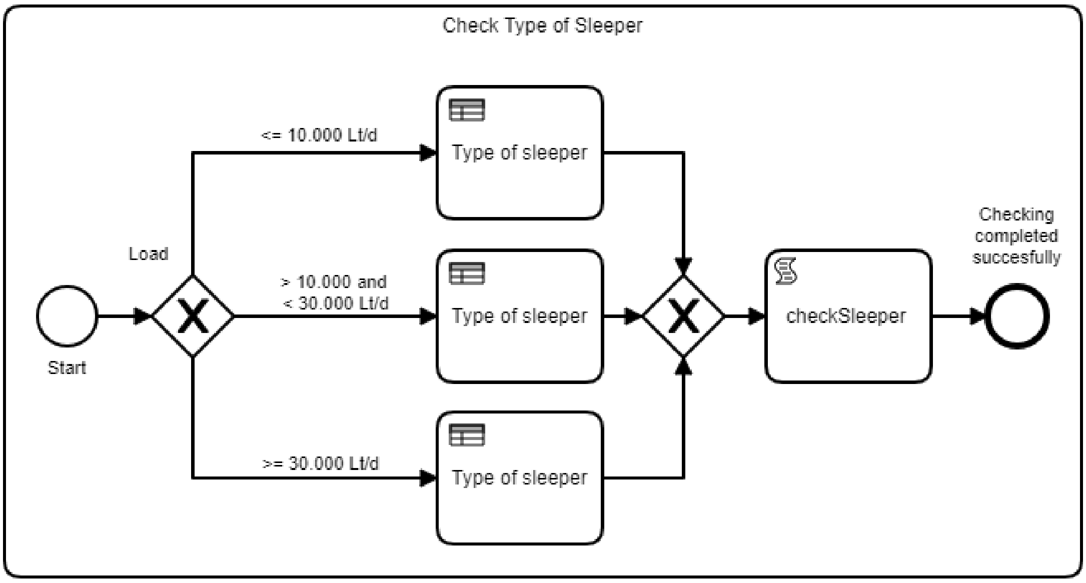


Figure 41: Process of checking the sleeper type represented as a BPMN process.

The example shown in Table 18 is an extract of the contents of the decision table for the selection of sleepers at a daily track load of >= 30,000 Lt/d. A relevant input variable for the choice of permissible sleeper type is speed. According to the table, three sleeper types, B70W, B70W-2,4 and B90W, are available for speeds below 160 km/h. For speeds in excess of 230 km/h, only sleepers of type B07W are permissible. Each sleeper type has a corresponding rail fastening and a permissible condition. In the example, only new material is permissible. Finally, the 'annotation' column contains notes on the use of the materials and any restrictions in use.

As the table shows, there are situations where several outcomes are possible, i.e. several sleeper types are permissible for the same speed. If there are no further restrictions, or none can be applied, all the applicable results are equally valid. In the context of the model check described here, a model already exists in which a corresponding configuration of superstructure components has been defined. Before the process is completed, a script task compares the sleeper type of the model with the possible options in the decision table, and the sleeper type in the model is checked to see if it matches one of the permissible sleeper types in the decision table. If the check is positive, the check status is set to ‘check passed’, and if it is negative, to ‘check failed’ (see Code 2).

Table 18: Decision table for sleeper types at a track load of ≥ 30,000 Lt/ d according to [8] *(extract).*

Input	**Output**			
Pace	**Type of sleeper**	**Type of rail fastening**	**Condition**	**Annotation**
< 160	B 70W	W14K 686a/687a	new	
< 160	B 70W-2,4	W14K 686a/687a	new	Only in crowded areas (e.g. slim subgrade)
< 160	B 90W	W14K 686a/687a	new	Only at railroad crossings and connection areas
[160…230]	B 70W	W14K 900	new	If radius > 800 m
> 230	B 07W	W21K 1000	new	
≥ 160	B 07W	W21K 1000	new	Only at connection areas

Code 2: Pseudo-code for comparing the superstructure components with the results of the decision table.

```
For (var index=0; index < decisionTypeOfSleeper.length; index++)
{
        if (decisionTypeOfSleeper == modelTypeOfSleeper) {
                setVariable("statusCheckSleeper", "check passed");
                exit for-loop;
        } else {
                setVariable("statusCheckSleeper ","check failed");
        }
}
```

The process configuration shown in Figure 41 can be used for all the track components mentioned above. Alongside text values, a decision table can also hold numerical values for automatic evaluation. The decision table shown in Table 19 describes the guideline specification for the ballast thickness of a track with a load ≤ 10,000 Lt/d. Depending on speed and sleeper type, the specified ballast bed thickness is 0.20 m or 0.30 m for steel sleepers. At track loads ≤ 10,000 Lt/d, speeds > 120 km/h are not regulated. In addition, several input variables can be supplied to the decision table. The table also serves as an example of how rules in the class ‘Distance Definitions: height’ can be defined with the help of BPMN and DMN.

Table 19: Decision table for ballast bed thickness at a track load of ≤ 10,000 Lt/d according to [8]. *The main decision criterion is sleeper type. Speeds greater than 120 km/h are not regulated.*

Input		Output	
Pace [km/h]	**Type of sleeper**	**Thickness of ballast [m]**	**Annotation**
≤ 120	not("steel sleeper")	0.20	
≤ 120	"steel sleeper"	0.30	

It takes 1-2 days to create a process as shown in Figure 41. A configuration of a superstructure with a total length of 15.1 km is used as test data. In total, the dataset contains 49 stations with 250 parameters that describe them. Table 20 presents the results of the measurements. The automated checking process takes 40 seconds (measured) for model preparation and rule execution. It is assumed that doing the same manually will take 25 min. The manual process takes approximately 40 min in total, compared with 7 min 40 sec for the automated checking process, which equals a ratio of 19 %.

Table 20: Comparison of manual and automated checking and the time needed to check superstructure test data

Phase	Description	Time needed for manual checking (approximation)	Time needed for automated checking	Ratio
1	Interpretation and logical representation of rules	-	-	
2	Building model preparation	10 min	20 sec	
3	Rule execution	15 min	20 sec	
4	Rule checking report	15 min	7 min	
	Total	40 min	7 min 40 sec	19%

In addition to the checking routines described here, other processes based on Guideline 820.2010 can be developed to check the compliance of track components, for example:

- Check rail type (rule class ‘Component definitions’)
- Distance check for ballast shoulder (rule class ‘Distance definitions: transverse direction’)
- Distance check for sleeper spacing (rule class ‘Distance definitions: position’)

4.6.4 Case study: Distance between tracks

The second most common rule class is ‘Distance definitions: transverse direction’. The case study presented here concerns a check of the distance between the track centers of double-track railway lines according to Guideline 800.0130, Appendix 02.

The following parameters and characteristic values are available in the modeling software for station-wise evaluation:

- Station per track
- Speed per track
- Radius per track
- Vertical distance between track centers

The process can be modeled using the same basic BPMN nodes as in the case study discussed above. Decision tables are likewise integrated into the process.

The guideline distinguishes between different areas and different structural and operational conditions when defining track spacing. Consequently, several decisions have to be taken to ascertain the respective situation and allow the model to be checked correctly. These are as follows:

(1) Is the radius of the alignment element greater than 250 m?
(2) Is the model of an open section of track or a station area?
(3) Is the open section equipped for bi-directional line operation?
(4) Are catenary masts needed between the tracks?
(5) Is an inspection walkway needed between the tracks?
(6) Is the track a new line or an upgrade of an existing line, and are the routes regional or long-distance?
(7) Are these main, secondary or passing tracks?

Irrespective of conditions 2 to 7, the question of whether the radius of the underlying elements at the station is greater than 250 m needs to be clarified. According to the guideline, this is the first exclusive gateway in the process (see Figure 42).

If the radius of the line element is greater than 250 m, the next step (Gateway 2) is to check whether catenary masts needs to be provided between the two tracks. If no catenary masts are required, then an inspection walkway is needed for the safe passage of railway personnel in the track area (Gateway 3). If the answer to this question is ‘No’, the required track spacing (without catenary masts or inspection walkway) shall be determined by means of a decision table. At Gateway 5, the track spacing in the model is checked against the result of the checking process and, if met, the status is set to ‘check passed’, otherwise to ‘check failed’.

If the radius of the alignment element is less than 250 m, the checking process is diverted to the ‘No’ branch at Gateway 1. In such cases, Guideline 800.0130 stipulates that the track spacing must be increased as specified in Table 21. The required increase in track spacing for different radii is given in the table, and the guidelines specify that intermediate values should be interpolated linearly. In the checking process, this table is modeled as a sequence of exclusive gateways, in combination with script tasks. As the degree of spacing enlargement is not described by a uniform formula for all radius ranges in Table 21, the interpolation formulae for each radius range are anchored in the process. The calculation only concerns the necessary delta of enlargement. The decision path for the different radius ranges cannot be defined as a decision table, because the DMN engine used does not permit the use of formulae as values. Radii of less than 100 m are not covered by the guideline, and in such cases, the checking process is aborted.

Alongside Gateway 3, Gateway 4 also represents a decision node that directs the process either to the decision tables “Track spacing with intermediate inspection walkway” or “Track spacing without intermediate inspection walkway”. The determined necessary enlargement is added to the result of the respective decision table by means of a script task. Gateway 5 then compares the model value with the criteria of the guideline and evaluates the result.

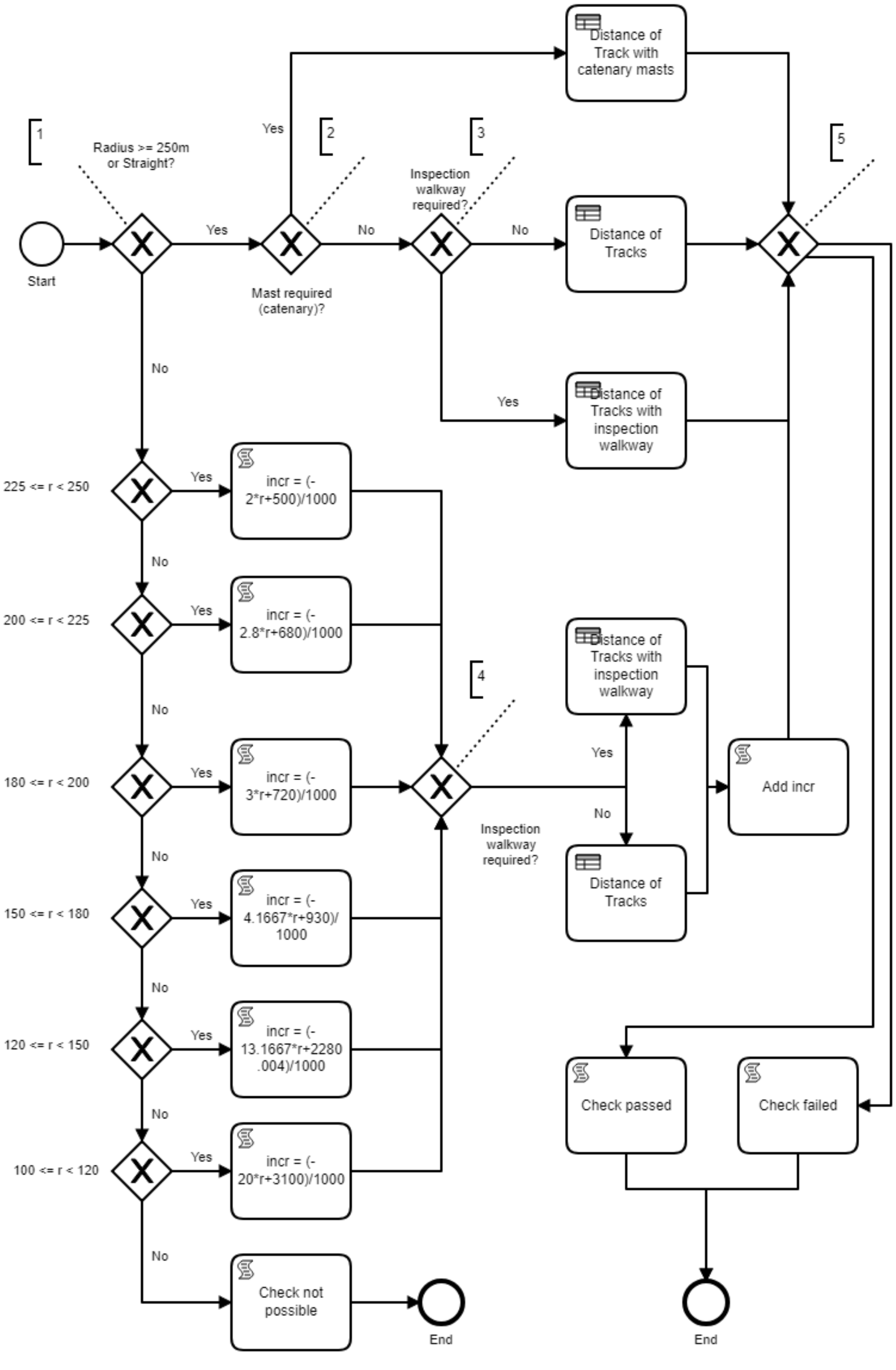


Figure 42: Process: Checking track spacing, represented as a BPMN process.

Table 21: Required increase in track distance according to the radius pursuant to [240]*. Intermediate values should be interpolated according to the guideline.*

Radius r [m]	**Required increase [mm]**
250	0
225	50
200	120
180	180
150	300
120	700
100	1000

Table 22 presents the decision table for the distance between tracks without an intermediate inspection walkway or catenary masts. Several input variables are needed, as determined by the questions above. Both the line itself ('open section' or 'station area') and the two tracks to be checked must be defined. Furthermore, the guideline defines speed specifications for the respective categories. For example, a track spacing of 4.5 m must always be maintained within station areas. Along upgraded tracks, at least 4.0 m is required in the speed range 0 to 200 km/h. "When upgrading existing tracks, a track spacing of 4.00 m can be maintained up to a design speed of ve = 230 km/h, if this faster speed is restricted to trains with favorable aerodynamic characteristics (such as the ICE)" [240]. Track spacings are also defined for new lines and local lines with suburban trains.

Table 22: Decision table for distances of tracks without intermediate walkway or catenary masts.

Input				**Output**	
Type of track	**Category Track 1**	**Category Track 2**	**Pace [km/h]**	**Distance of Tracks [m]**	**Annotation**
Outside stations	Upgraded track	Upgraded track	[0…200]	4.0	
Outside stations	New track	New track	[0…300]	4.5	
Outside stations	Short-distance track	Short-distance track	[0…120]	3.8	
Outside stations	Upgraded track	Upgraded track	[200…230]	4.0	Only for trains with aerodynamic characteristics
Inside stations				4.5	

In the case described, the track specifications are categorized from a construction point of view. Where definitions for track spacings for walkways are made, the categorization follows operational concerns. The terms main track, secondary track and passing track are used here. For this study, it has been assumed that this information is not contained in the model and must be supplied manually by the user. For the user, however, the problem arises that several descriptions are available, and it is not clear which categorization is used for which decision table. A user might assign the track data 'correctly' from his or her point of view – for example from a construction point of view – but the decision table requires operational categorization. In such a case, the check would not be carried out correctly. This is a limitation in the context of check and decision automation.

It takes 1-2 weeks to create a process such as the one shown in Figure 42. Two axes, each of a total length of 9.4 km, are used as test data. The dataset contains 946 distance values in total. Table 23 shows the results of the measurements. The automated checking process takes 40 seconds (measured) for model preparation and rule execution. It is estimated that performing the same checking process manually will take approximately 35 min. As a result, the manual process takes approximately 50 min in total, compared with 9 min 30 sec for the automated checking process, which equals a ratio of 19 %.

Table 23: Comparison of manual and automated checking and the time needed to check superstructure test data

Phase	**Description**	**Time needed for manual checking (approximation)**	**Time needed for automated checking**	**Ratio**
1	Interpretation and logical representation of rules	-	-	
2	Building model preparation	10 min	30 sec	
3	Rule execution	25 min	2 min	
4	Rule checking report	15 min	7 min	
	Total	50 min	9 min 30 sec	19%

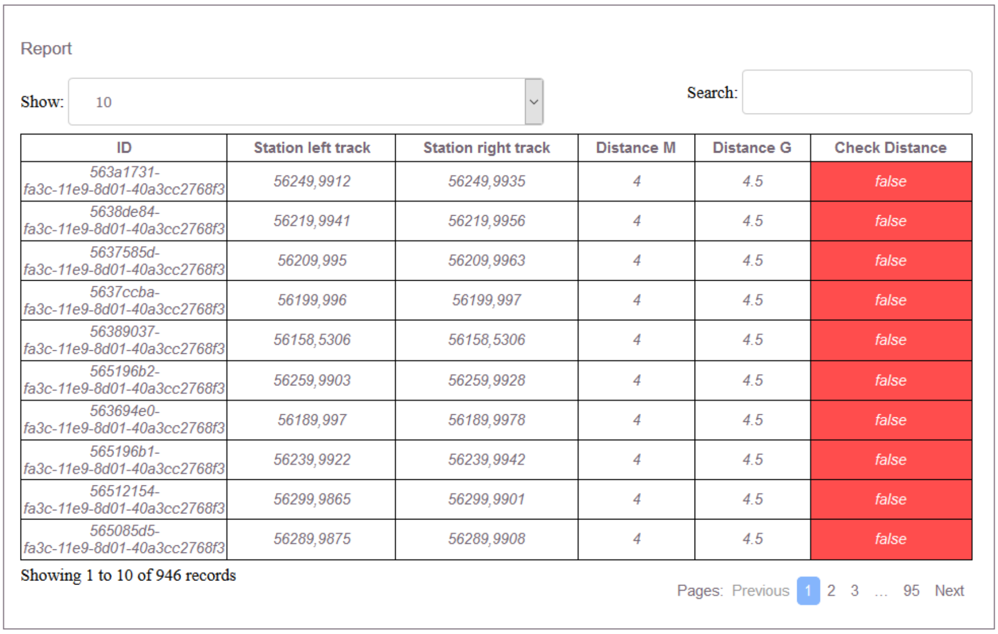
4.7 The front end of the checking tool

To make the checking system more user-friendly, a front end using current web technologies has been developed that echoes the four phases defined by Eastman in [87]:

(1) Interpretation and logical representation of rules
(2) Building model preparation
(3) Rule execution
(4) Rule check reporting

As explained in the preceding sections, the interpretation of the guidelines and their specifications and their translation into process diagrams were undertaken manually in DM and BPM Notation. The user can access the checking processes in the user interface using a BPMN viewer. The 3D model is transferred to the checking routine by the user. Additional information which is required but not available in the model can be supplied by the user via web input forms. The check is then performed automatically by the workflow engine and the results passed from the back end (workflow engine) to the front end via a Rest API.

The results are displayed in the form of a table showing each checked station and each checked parameter. Figure 43 shows an example report of the checking procedure for the distance between two tracks. The model contains a track distance of 4.0 m but the boundary conditions supplied by the user (as described in Section 4.6.4) meant that the guidelines specified a track distance of 4.5 m. The model check therefore returns a negative result throughout.



Report

Show: 10

Search:

ID	Station left track	Station right track	Distance M	Distance G	Check Distance
563a1731-fa3c-11e9-8d01-40a3cc2768f3	56249,9912	56249,9935	4	4.5	false
5638de84-fa3c-11e9-8d01-40a3cc2768f3	56219,9941	56219,9956	4	4.5	false
5637585d-fa3c-11e9-8d01-40a3cc2768f3	56209,995	56209,9963	4	4.5	false
5637ccba-fa3c-11e9-8d01-40a3cc2768f3	56199,996	56199,997	4	4.5	false
56389037-fa3c-11e9-8d01-40a3cc2768f3	56158,5306	56158,5306	4	4.5	false
565196b2-fa3c-11e9-8d01-40a3cc2768f3	56259,9903	56259,9928	4	4.5	false
563694e0-fa3c-11e9-8d01-40a3cc2768f3	56189,997	56189,9978	4	4.5	false
565196b1-fa3c-11e9-8d01-40a3cc2768f3	56239,9922	56239,9942	4	4.5	false
56512154-fa3c-11e9-8d01-40a3cc2768f3	56299,9865	56299,9901	4	4.5	false
565085d5-fa3c-11e9-8d01-40a3cc2768f3	56289,9875	56289,9908	4	4.5	false

Showing 1 to 10 of 946 records

Figure 43: Example of a report showing the result of a compliance check for the required track distance. The distance in the model is 4.0m but the guidelines stipulate a distance of 4.5m. The report also shows the ID of the checking process and the station where the two tracks are located.

The meaning of the modal auxiliary verbs (see Section 4.3) for the rules are simple to integrate into the workflow using a script task for the entire workflow that defines a corresponding variable. Where the weighting of a rule's importance of needs to be represented in the context of decision tables, a variable can be outputted. When displaying the results in the front end, the variable is then evaluated accordingly. For example, violations of rules and prohibitions can be marked in red, as these are non-permissible. Deviations from principles can be marked in yellow as these may still be permissible and do not necessarily impact negatively on the model quality. The quality of the model can therefore also be weighted in relation to these conditions.

4.8 Conclusion

In the construction industry, numerous standards such as norms and guidelines exist in written form that can be read by humans but not processed by machines. The checking of plans and models against such specifications is currently performed predominantly manually and is therefore a time-consuming and error-prone process. This particularly applies to the railway domain, where numerous regulations must be strictly followed to ensure the safety of rail traffic and passengers.

This study examined the extent to which the Business Process Model and Notation defined in ISO 19510 can be used to graphically represent guideline content and make them executable for model checking. Although applying similar approaches to *Dimyadi et al.*, who focus on buildings [243], the focus of the study is on railway projects and their dedicated guidelines. In the railway domain, the BPMN approach particularly benefits from the precision and clarity of rules originally written in natural language. The BPMN approach was additionally supplemented by the Decision Model and Notation, which is also divergent from other research.

The study presents an analysis of selected Deutsche Bahn AG guidelines and the capability to represent and automate them with BPMN and DMN elements. The analysis showed that 52% of the rule sets examined can be automated, and 46 % of the examined rules were classified as being implementable using BPMN and DMN. By conducting a detailed classification, the rules were categorized into 12 different rule classes. By evaluating the frequency of occurrence of the individual classes, the top three rule classes were identified, and corresponding case studies designed and tested for feasibility. A further case study was also designed for elementary alignment planning within the infrastructure planning. To allow data to be imported from railway modeling software, reference was made to the current state of the IFC schema, and the use of IFC data validated.

Overall, the representation of guideline content using BPMN and DMN is a promising approach. Aside from fulfilling the designated goal, the graphical representation of the process makes the process transparent and enables the user to understand the checking process, which is a significant advantage over hard-coded 'black box' solutions. Due to the standardized representation of the process elements and the widespread use of the notation in the context of IDM, it is conceivable that this method could be widely implemented in the construction industry.

In this study, existing guidelines were manually translated into process diagrams to make them machine-readable. RASE-syntax was used to support the translation of regulation texts into workflow diagrams. Natural language processing methods offer a promising alternative. Future studies could aim to combine both methods, with a view to automatically translating guideline contents into process diagrams using NLP. It is also conceivable that authors could specify future regulations from the outset as BPMN processes, to facilitate the compliance checking of models. Currently there is no connection to a graphical 3D-representation such as a BIM-Viewer. The realized tool analyzes the given parametric of 3D-geometries, but does not calculate e.g. distances between 3D-objects (clearance test). These limitations could be interesting in future studies. Even if the visual representation of guidelines is more a ‘white-box’ than a ‘black-box’ solution, workflow-developers have to have experience in programming. BPMN is primarily a standard for modeling business processes. It is not meant to be a representation for engineering aspects and it is not specific for any engineering design software, which means, it is always necessary to develop import and analyzing functionalities. In this study, it is also shown that BPMN and DMN cannot be used if sophisticated software e.g. for structural analysis or simulation is necessary.

Although the current status of IFC in the field of infrastructure is not yet sufficient to carry out fully-fledged model checks in the railway domain, the recent supplementary studies in the context of IFC-Railway are promising and will likely provide all the information required to enable efficient checking of railway models against current regulations.

5 Knowledge-Based Engineering in the context of railway designs by integrating BIM, BPMN, DMN and the methodology for knowledge-based engineering applications (MOKA)

Previously published as: Marco Häußler, André Borrmann. *Knowledge-based engineering in the context of railway design by integrating BIM, BPMN, DMN and the methodology for knowledge-based engineering applications (MOKA),* Journal of Information Technology in Construction (ITcon), Vol. 26, pg. 193-226, 10.36680/j.itcon.2021.012

Abstract

Designing railway infrastructure is a knowledge-intensive task. Although there are a number of mature design authoring systems available, their support for dynamically incorporating domain-specific engineering knowledge is very limited. At the same time, a standardized digital representation of railway engineering knowledge (such as building codes and best practice) does not exists. To overcome this deficiency, this paper proposes the use of Knowledge Based Engineering (KBE) to automate routine design tasks by considering multiple knowledge sources. In this scenario, KBE is used to support a Railway design authoring system. To ensure maximum transparency in the design of the developed KBE application, graphical ‘Business Process Model and Notation’ (BPMN) has been used in combination with ‘Decision Model and Notation’ (DMN) to formalize the underlying engineering knowledge. The KBE application has been developed according to the Methodology for Knowledge-Based Engineering Applications (MOKA). An evaluation of the BPMN/DMN approach shows that it meets up to 58% of the acceptance criteria found in the literature. In addition, BPMN and DMN can already be used in the early capture phase of MOKA and its workflows can be developed into an executable KBE application in the subsequent phases. The results of the test example discussed here show that time savings of up to 97.5% can be achieved in the execution of the KBE application.

5.1 Introduction

‘Digital methods like building information modeling (BIM) offer considerable advantages in the construction industry over the conventional methods that still largely prevail in current practice’ [244]. With conventional methods, we refer to workflows based on 2D drawings for design, handover to clients and production on site. This applies both to compliance checking, as shown by *Häußler et al.*, as well as to the design of buildings and the associated model creation. Design authoring systems which support object-oriented workflows combined with an parametric modeling approach (also referred to as BIM modelers) are used widely for the design of built structures, see [63,245–247] . The systems on the market already support design engineers by providing component libraries that also define a certain building logic or interdependencies between components in the software, see also [34–39]. In the design of infrastructure, there are numerous such dependencies. The most important element is typically the alignment as nearly all other infrastructure facilities are aligned to it [231]. Railway design authoring systems therefore already provide a framework within which the designer or specialist engineer can work in the course of the design process.

The individual calculation and design steps are often based on knowledge that is already set out in standards and guidelines in the form of rules, see [244]. At present, however, the construction industry lacks standardized methods for digitally mapping existing knowledge in such a way that it is machine-readable, so that it can be evaluated and reused, compare [5]. The extension of the standardized data model Industry Foundation Classes (IFC) by railway infrastructure objects (IFC rail) recently reached the candidate status [248]. While this standardized object-oriented description will provide a good basis for representing and exchanging railway facilities with rich semantics, it does not provide capabilities for representing and applying the knowledge required for designing them.

Especially the digital integration of different knowledge sources is where knowledge-based engineering (KBE) comes in [111]. KBE stands at the intersection of diverse fundamental disciplines, such as artificial intelligence (AI), CAD and computer programming [112]. KBE systems can be applied to achieve a variety of goals, the most important of which is to increase efficiency by using automated systems for routine tasks, as described, for example, in [111]. ‘Although various research approaches have been developed in the area of KBE since the 1980s, there is no uniform and universally applicable description for the industrial environment, with which a KBE application can be implemented and operated’ [249]. *Häußler et al.* present a promising approach to mapping guideline contents in a machine-readable form with the help of ‘Business Process Model and Notation’ (BPMN) in connection with ‘Decision Model and Notation’ (DMN) and demonstrate its use for code compliance checking. In this paper the extent to which BPMN and DMN can be used as basis for a KBE application is investigated.

The paper is structured as follows: Section 5.2 gives an overview of the state of the art with regard to the terms “knowledge” and “engineering”, and proceeds to discuss KBE as a technological symbiosis of these two domains. Following on from this, the Methodology for Knowledge-Based Engineering Applications (MOKA), as well as BPMN and DMN are presented. Section 5.3 describes the methodology used in the present study and the separate MOKA phases: Identify, Justify, Capture, Formalize and Package. Finally, Section 5.4 presents its use in a case study.

5.2 State of the art

KBE synthesizes aspects from the domains of “knowledge” and “engineering”. Both are first discussed independently and KBE is presented afterwards.

5.2.1 Knowledge and knowledge management

ISO 30401:2018 defines knowledge as a ‘human or organizational asset enabling effective decisions and action in context’ [250]. A common means of representing how knowledge arises is the DIKW hierarchy (**D**ata, **I**nformation, **K**nowledge, **W**isdom) as described in [251] and depicted in the pyramid in Figure 44. Information is created by processing and interpreting data. The processing and combination of information creates knowledge, which in turn results in wisdom [251]. *Bellinger et al.* define the terms as follows:

‘Data represents a fact or statement of event without relation to other things. **Information** embodies the understanding of a relationship of some sort, possibly cause and effect. **Knowledge** represents a pattern that connects and generally provides a high level of predictability as to what is described or what will happen next. **Wisdom** embodies more of an understanding of fundamental principles embodied within the knowledge that are essentially the basis for the knowledge being what it is. Wisdom is essentially systemic’ [252].

According to *Schreiber et al.* [253] the definitions of data, information and knowledge are widespread, as can be seen for example in *Rezgui et al.* [254], *Premkumar et al.* [255], *Girodon et al.* [256] and *Roth et al.* [257]. From a philosophical point of view, the term “knowledge” is not as simple or clear-cut to define as *Bolisani & Bratianu* [258], among others, have shown, but an epistemological study of the philosophical basis of knowledge is not relevant in the context of this paper. Rather, the observations made here are based on the interplay of data, information and knowledge, since there is consensus on this in the literature [253]. In the literature there is only ‘limited discussion of the nature of wisdom, and even less discussion of the organizational processes that contribute to the cultivation of wisdom’ [251], as also confirmed by *Liew* [259].

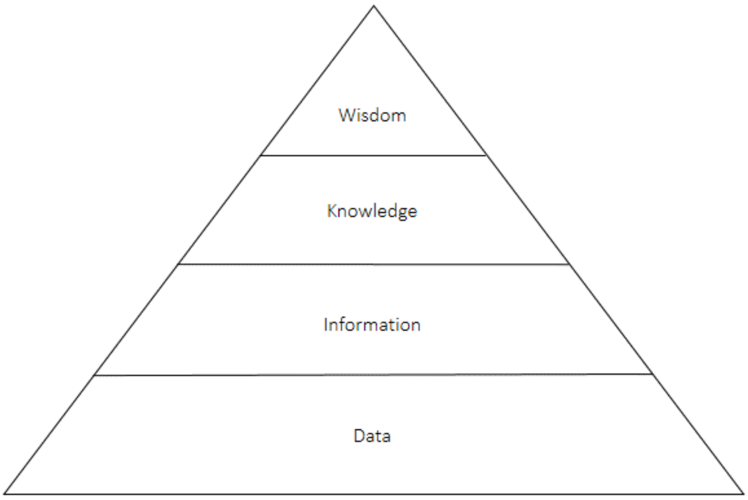


Figure 44: DIKW hierarchy (Data, Information, Knowledge, Wisdom) as described in [251]

Aside from their terminological definition, *Roth et al.* describe knowledge in terms of type, character, form, location and knowledge quality. Figure 45 shows relationship between the types of knowledge identified by *Roth et al.* and lists the different types of knowledge (explicit/implicit, structured/unstructured, etc.).

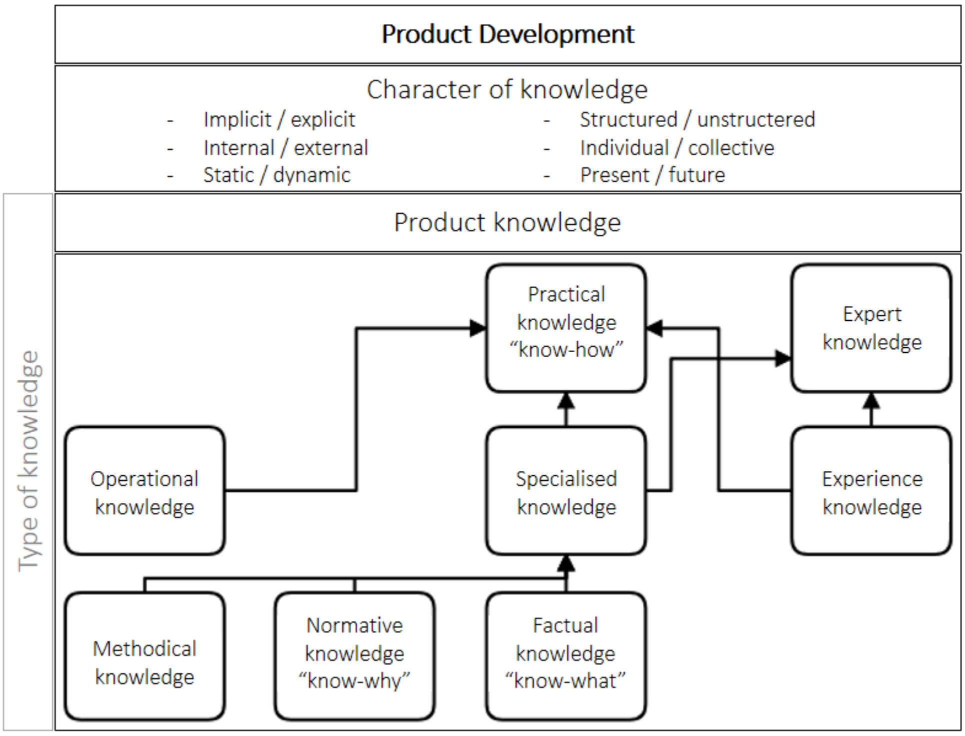


Figure 45: Types of knowledge based on [257]

Storing knowledge and making it available is the subject of many studies. The works of *Haller* [260], *Premkumar et al.* [255], *Girodon et al.* [256], *Cheng et al.* [261], *Wu et al.* [262] as well as research on the topic of “Design Structure Matrices” (DSM) [263,264] are examples of this. A DSM can help clarify dependencies, connections and interfaces of system elements. The integration of knowledge management and BIM is the subject of the investigations of *Liu et al.* [265]. *Chassiakos et al.* describe an approach of using a knowledge-based system (KBS) for the maintenance of bridge structures [116].

5.2.2 Engineering design process

Alongside the concept of knowledge, it is important to understand how engineering and the associated design process works. According to *Ertas*, design can be investigative, creative, rational (logic-based) or decision-oriented (value-based) [266]. According to *Calkins et al.* the design of a product is ‘an ordered set of steps that are performed to accomplish a task. The design of a product traditionally proceeds through a series of well-defined stages or phases including:’ [267].

- conceptual design (concept exploration and development)
- preliminary design
- detail design (production design)

The design process of products is a gradual and iterative process within which different participants solve different tasks and design individual components of the product [215].

5.2.3 KBE

The preceding sections discussed knowledge and the engineering design process independently of one another and placed them in the context of this study. KBE links the two domains and can be used to automate the routine parts of the work steps. ‘A prevalent definition of KBE emphasizes “the capture and systematic reuse of the product and process engineering knowledge” to automate “repetitive and non-creative design tasks” and to support “multidisciplinary design optimization in all phases of the design process”’ [112,268]. *La Rocca* states that ‘Knowledge-based engineering (KBE) is a technology based on the use of dedicated software tools called KBE systems, which are able to capture and systematically reuse product and process engineering knowledge, with the final goal of reducing the time and costs of product development’, see also [111]. In contrast to KBS, as used in the context of KM, KBE applications have the capacity to influence the geometries of a design, see also [111,112]. ‘The basic objectives that have to be supported by KBE are: solve a particular design problem using a KBE application (short-term), and retain the domain knowledge required for solving design problems in the same domain (long-term)’ [269].

According to *La Rocca* and *Cooper et al.* KBE systems have “generative” and “integrated” modelling capabilities (see Figure 46), which means that ‘a set of input values is assigned to the parameters that are used in the product model, the KBE system applies the rules to process the input values and, finally, the engineered design is generated, with little or no human intervention’ [112]. ‘A generative model differs from a geometric model, which is a typical output of advanced CAD systems. Where a geometric model is a model of a designed product with fixed features (dimensions and configuration), a generative model is a generic representation of the product. The generative model […] is built on the basis of a geometric model and is enhanced by the engineering rules that determine its design' [270].

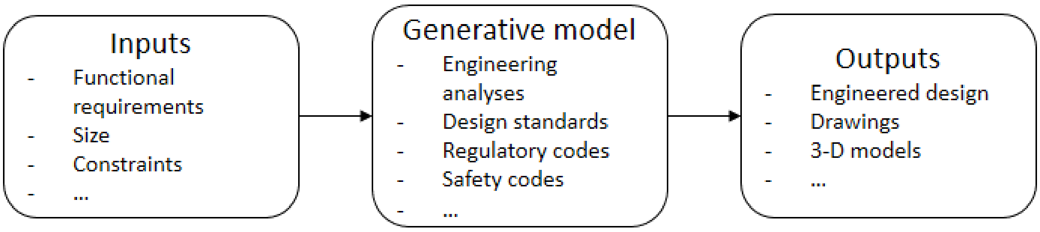


Figure 46: The generative model of a KBE application takes input specifications, applies relevant rules and automatically produces an engineered design (based on [112]*)*

KBE systems can either be integrated directly into a design authoring system or be kept separate from the design authoring system and coupled via interfaces, as shown in Figure 47.

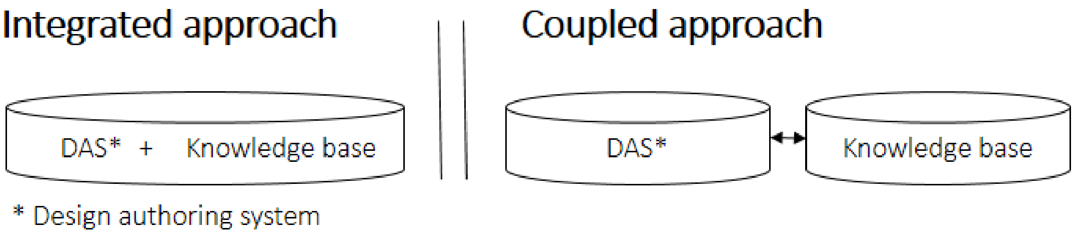


Figure 47: Integrated and coupled approach of KBE and Design authoring system [249]

There are numerous practical examples of the development of KBE applications. These range from KBE systems in the field of aerospace [271–273] or aerodynamics [269,274], for cost estimation [268,275], building design [276], building operation [277–279], building analysis [280,281], sustainability [282], safety planning [94], structural design [246,283] and bridge modelling [114,117,284–286]. *Johansson et al.* present an approach to representing the knowledge contained in a KBE application using graph theory [33,158]. The ability of a KBE system to support decisions is particularly important, as the investigations of *Chen et al.* [287], *An et al.* [288], *Egemen & Mohamed* [289] and *Wang et al.* [290] show.

Garcia & Ip-Shing [269] and *Sainter et al.* [291] state that long-term problems in development of KBE systems exist. These are:

- Knowledge loss, due to poor modelling of the applications
- Knowledge loss, due to the development language
- Knowledge misuse, due to incorrect selection of the applications being developed
- Increased maintenance costs, due to the lack of standardization of applications, and
- Knowledge underutilization, due to the difficulties in sharing and reusing knowledge.

According to *Sainter et al.,* many of these long-term problems can be mitigated using standardized development and management methods, languages and frameworks [292].

According to *La Rocca* [112] and *Tripathi* [293] KBE systems are also referred to as expert systems. *Tripathi* defines the components of a KBE system as follows (see also Figure 48): ‘A rule-based expert system contains a knowledge base, inference engine, knowledge acquisition, explanation facility and user interface’ [293].

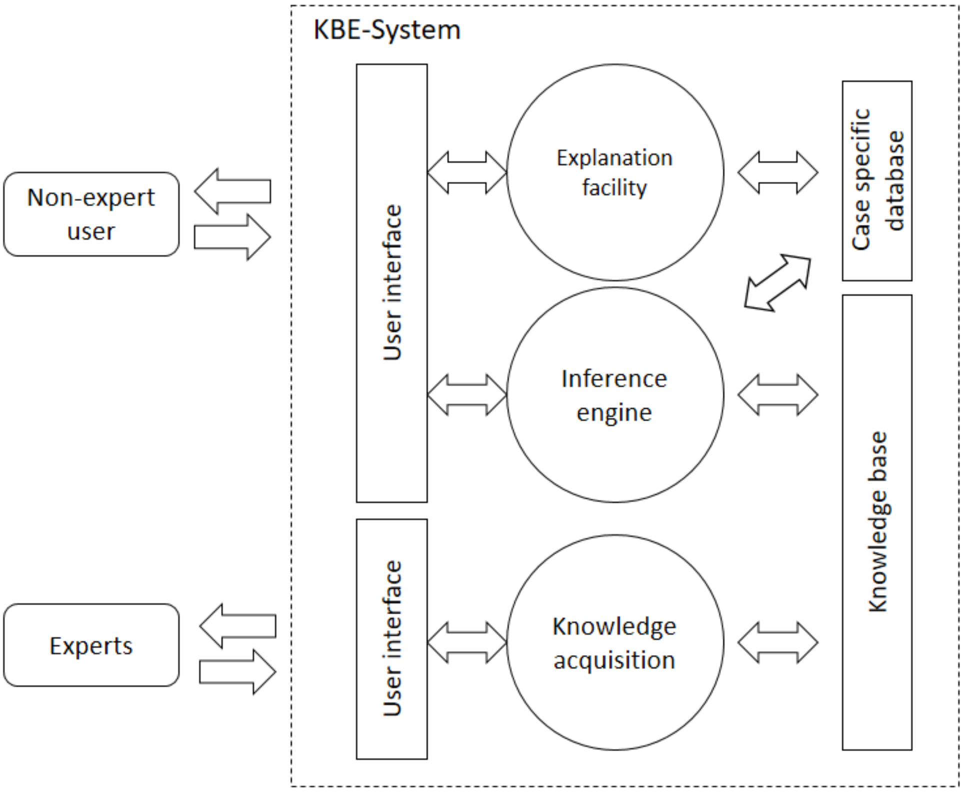


Figure 48: Structure of a KBE system according to Tripathi and Singer [284,293]

5.2.4 MOKA

MOKA was developed to reduce the risks and investment costs associated with the development of KBE systems [111]. The method is a modular system consisting of six steps, as shown in Figure 49.

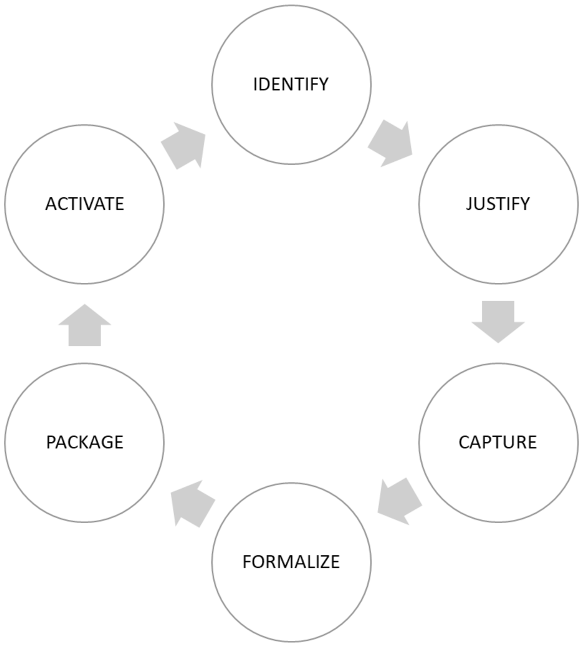


Figure 49: The KBE lifecycle [111]

The contents of the individual steps are described in detail by *Stokes*. In this paper the focus lies on “Identify”, “Justify”, “Capture”, “Formalize” and “Package”.

MOKA provides a method for collecting and structuring knowledge from different sources (Capture) and transferring it step by step into a formal representation (Formalize), so that a KBE application can be developed from it (Package), as illustrated in Figure 50. The methodology also indirectly supports communications between the active participants (e.g. expert, knowledge engineer, software developer). Through its generic approach, the modules of the MOKA process, such as notations or forms of presentation, can be exchanged or supplemented by equivalent elements if necessary.

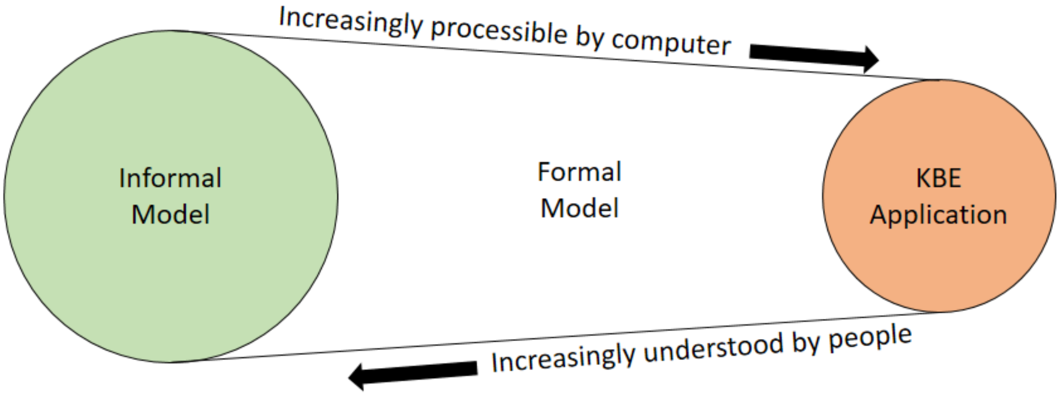


Figure 50: Assessment of the informal and formal model of human readability and computer processability [111]

MOKA is used in numerous studies as a method for the development of KBE applications. There are examples in aerospace [294], mechanical engineering [295] and architecture [296,297], and MOKA is also used in connection with inspection and manufacturing [298–300]. In certain fields of application, MOKA has gaps, which have been identified in various investigations and closed by domain-specific extensions, for example for the automated planning of processes as MOKA understands a product model as a physical object [301,302]. *Chan* has extended the method to develop simulation workflows automatically with the help of KBE mechanisms [97]. *Skarka* has used Protégé to develop knowledge ontologies based on MOKA, see [270,303]. Due to the use of MOKA in aerospace and mechanical engineering, Dassault CATIA is often used as a software package for developing KBE applications using MOKA, see [270,304,305].

Sandberg has examined literature in the field of KBE, MOKA and construction and concluded that while individual MOKA phases are adopted, concrete details on their implementation are usually lacking: "[…] Design automation research focuses too much on the application itself rather than describing the way the application was developed" [306]. Among other things, the definition of acceptance criteria for the successful implementation of a KBE application is criticized. With the help of acceptance criteria the success or failure of the KBE application shall be judged [111].

The literature research revealed that the MOKA method is often used to develop KBE applications, but the examples were predominantly hard-coded "black box" solutions. MOKA aims to minimize the language barriers between Experts, Knowledge Engineers and Software Developers, but MOKA focusses only up to the formal model and describes these steps detailed. The package and activation step is not in focus of MOKA [111].

For the present study, the advantages of the basic principles (collecting and structuring knowledge) of MOKA outweigh the disadvantages (need for domain-specific extensions). The literature research shows that MOKA is generic and extensible, which means that the disadvantages of pure MOKA are manageable. MOKA is therefore elected to be used but instead of developing a “black box” solution, an approach based on a graphical notation is employed, as described in the next section.

5.2.5 BPMN and DMN

A primary objective of the MOKA method is the creation of the formal model, consisting of a product and activity model. *Stokes* recommends the Unified Modeling Language (UML) for the development of product models and the corresponding UML activity diagrams for the design process model [111]. The focus of this research lies on the presentation of the design process model. MOKA does not detail the transfer of the formal model into an executable KBE application in any detail and only recommends converting the formal model into a neutral data format such as Extensible Markup Language (XML) to standardize the transformation process. In contrast to the MOKA recommendation, this study instead looks at how BPMN can be used in place of UML activity diagrams.

‘An activity diagram is a UML behavior diagram which shows flow of control or object flow with emphasis on the sequence and conditions of the flow. The actions coordinated by activity models can be initiated by other actions finishing execution, by objects and data becoming available, or by the occurrence of some event external to the flow’ [307].

Alongside the UML activity diagrams, workflows can also be formally represented with the help of the BPMN. BPMN is an international standard (ISO/IEC 19510) that is maintained by the Object Management Group [208]. Recker et al. describe BPMN as a ‘[…] structured, coherent and consistent way of understanding, documenting, modeling, analyzing, simulating, executing, and continuously changing end-to-end business processes and all involved resources in light of their contribution to business performance’ [209]. ‘BPMN gained popularity because it is a standardized graphical notation, adopted by the Object Management Group (OMG), and it is easy to understand, also for non-IT experts. Since version 2.0, BPMN also includes execution semantics and a standardized XML-based syntax’ [97].

Chan compares various process modelling languages and evaluates them in the categories ‘graphical notation’, ‘designed for execution’, ‘standardized’, ‘widely adopted’ and ‘easy to use’ (see also Table 24). *Chan*’s results show that BPMN is better suited than UML activity diagrams due to the ability to make workflows executable with the help of a corresponding workflow engine. A workflow engine ‘performs the process step by step in accordance with the modelled logic and can react to events during runtime as well as trigger events itself. Such an event might be a data input by the user, the calculation of mathematical formulae, a decision resulting from an if-then condition, or the execution of text-based source code. By virtue of its ability to automate the developed processes by means of a workflow engine, BPMN can be classified as belonging to the category of visual programming languages’ [244]. In practice, however, *Chan* only uses BPMN for visualizing the workflow but does not make it executable.

Table 24: Trade-off of various workflow modelling languages; score: 2 plus points (++), 1 plus point (+), no point (0), 1 minus point (-), 2 minus points (--) [97]*.*

Criteria	**Weight**	**UML**	**BPMN**
Graphical notation	1	++	++
Designed for execution	1	-	+
Standardized	1	++	++
Widely adopted	1	++	++
Easy to use	1	+	+
Score	**Max: 10**	**6**	**8**

BPMN can be used in many ways. Recker states that '"classical" process management applications such as documentation, redesign, continuous improvement and knowledge management dominate the application areas of BPMN, while more technical application areas such as software development, workflow management or process simulation are not (yet) widespread' [222]. In the AEC industry, BPMN is used for the development of IDMs [211,214,215], which is also recommended by [213]. *Alreshidi* combined UML diagrams with BPMN workflows to develop a cloud platform [212]. *Dimyadi et al.* and *Häußler et al.* used BPMN with corresponding workflow engines to verify building data models in the context of code compliance checking [105–107,244].

It is specifically the executability of BPNM workflows that set them apart from other methods, as *Borrmann et al.* confirm: 'In particular, the option to use BPMN in the context of workflow management systems as an implementation language […] is an important criterion for use in construction projects' [217]. 'The most frequently used BPMN elements for representing business processes in the AEC industry were sequence flow, pool, lane, task/activity (exclusive), gateway, and message flow, which are all basic BPMN modelling elements' [216]. A graphical representation of the individual elements is shown in Figure 51.

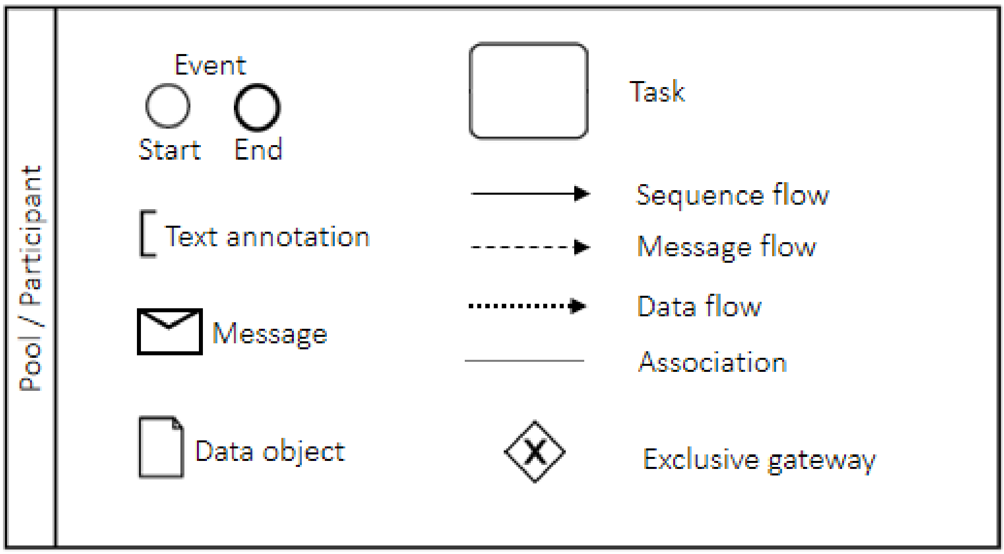


Figure 51: Proposal for an 'Essential Subset' of BPMN for IDM Development, based on [216].

As discussed in Section 2.3, KBE applications must have the capacity to support decisions or even make decisions independently. To this end, DMN augments BPMN. 'The primary goal of DMN is to provide a common notation that is readily understandable by all business users, from the business analysts needing to create initial decision requirements and then more detailed decision models, to the technical developers responsible for automating the decisions in processes, and finally, to the business people who will manage and monitor those decisions. DMN creates a standardized bridge for the gap between the business decision design and decision implementation. DMN notation is designed to be usable alongside the standard BPMN business process notation' [225]. With the help of Decision Engines (comparable with Workflow Engines) DMN decision tables can be made executable. Through the integration of DMN decision tables in BPMN workflows, decision paths can be incorporated directly into the workflow engine, see also [244].

With the help of BPMN and DMN, knowledge, processes and associated decision paths can be represented graphically and thus made more readily understandable than in text-based languages. The special feature of both notations compared to UML is their executability. Compared to MOKA, where the formal model, once complete, has to be manually translated into a specific KBE language by a software engineer, BPMN and DMN do not require any further translation or transformation into text-based code. Instead, the graphical notation can be used directly to develop the informal model in order to visualize process sequences. In the course of developing the formal model, and ultimately also the KBE application itself, the first draft can be developed further without having to switch between different forms of presentation.

5.2.6 Designing in track construction

The aim of this study is to automate design steps in the field of rail infrastructure design with the help of KBE. The focus lies on design of new buildings. According to *Borrmann et al., a* fundamental distinction must be made between explicit and implicit modelling when creating three-dimensional models. 'Explicit modelling, […] describes a volume in terms of its surface […]. Implicit modelling by contrast employs a sequence of construction steps to describe a volumetric body, and is therefore commonly termed a procedural approach' [62]. Of particular relevance for this study is parametric modelling, which *Borrmann et al.* describes as a possible implicit approach. 'Feature-based parametric CAD is currently the industry standard technology to create geometric models and assemblies, and is widely used across many engineering fields' [63]. The importance of parametric modelling is also described by Tang et al. [308], Brown et al. [309] and Zou & Feng [247].

For this research the term "design authoring system" is used as a synonym for software which provides capabilities for the object-oriented and parametric design of geometric models in context of railway infrastructure facilities. The term BIM is used as a synonym for the object-oriented workflow during the building lifecycle, it is not considered as a specific software product. The tools used for this research are presented in Section 5.3.5.

The definition of the parametric 3D-model in software products like Civil 3D [34], OpenRail Designer [35], card_1 [36], Vestra Rail [37], ProVI [38] or Ferrovia [39] is managed with the help of a 'drawing-oriented view – split into site plan, cross-section and elevation [see also Figure 52]. This type of model is referred to as an implicit geometry description. With implicit models, the governing design parameters become significantly more accessible than with explicit models' [176]. The description of the design of railway infrastructure in the underlying guidelines is predominantly parameter-oriented [244].

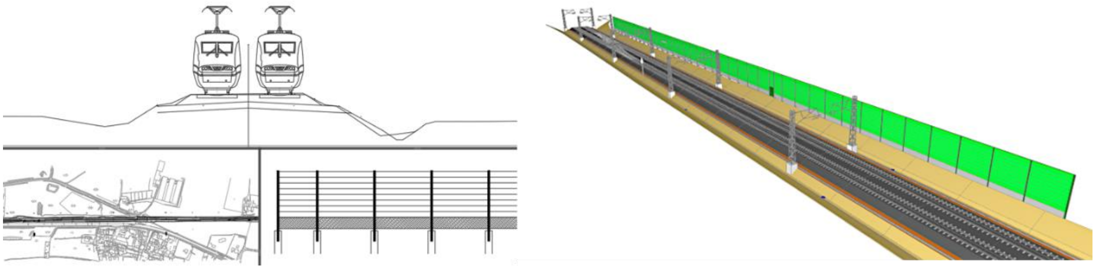


Figure 52: Comparison of implicit and volumetric 3D models: while implicit models (drawing-oriented view) are used during the design process (parametric modeling), explicit models are used in the context of BIM-based analysis [176].

'The procedural approach provides the user of the system with the possibility to easily modify an existing model by going back in the construction history and adapting the corresponding parameter […]' [32]. This is a particular advantage in the context of KBE. For the automation of design steps, the use of a design authoring system which supports the parametric modeling approach is indispensable, as this offers the greatest degree of freedom compared to other modelling philosophies, especially explicit modelling.

5.3 Methodology

In this paper, a KBE application is developed based on the BPMN and DMN notations standardized by the OMG. The intention is to minimize the break that arises when developing an informal model into a finished KBE application using the MOKA method. BPMN and DMN are used not just to represent technical knowledge but also process flows graphically, making them transparent for those involved in development as well as for later users. The ability to make BPMN and DMN executable using a workflow engine makes this approach interesting for the development of KBE applications.

This section describes the following steps of the MOKA method: Identify, Justify, Capture, Formalize and Package.

5.3.1 Identify

This paper focuses on the development of a KBE application to support the design of new buildings along railway infrastructure. The suitability of BPMN and DMN for representing guidelines in the railway design section was discussed in earlier work by *Häußler et al.* which concluded that up to 68 % of the examined rules can be digitally mapped using BPMN and DMN. A commercially available Design authoring system was used, and the KBE application aims to support the work steps of an engineer using this specialist software.

The following key sources of knowledge can be identified:

- Expert knowledge (human sources)
- Contents of guidelines (documents)
- Requirements and modes of operation of the computer application (computer files)

The classified objectives for the development of a KBE system are as follows:

- The structure of a KBE system is fulfilled (very important)
- Use of a graphical notation to provide transparency (very important)
- Use of a standardized development language (very important)
- Combination of different knowledge sources is possible (important)
- Combination with commercially available Design authoring systems, the KBE system generates parameters for Design authoring system which is done usually by users (important)
- The KBE system decreases design time and ensures design quality (important)

5.3.2 Justify

The lack of universal definitions for acceptance criteria has been criticized in the literature, most notably by *Sandberg. Stokes* recommends the definition of (acceptance) criteria to verify the success or failure of KBE systems. As a result, criteria are compiled and clustered for the successful implementation of KBE applications in the course of literature research. These were then used to evaluate the approach of a BPMN and DMN-based KBE system developed in this study. Taking the objectives of this work into account not all criteria are necessarily equally important, the criteria were additionally weighted. The criteria which meet the very important objectives are weighted three times, the important ones two times and the one which are non-important compared to the objectives are weighted with one.

The criteria, their weighting, the corresponding sources, comments and evaluation result are shown in Table 25. The evaluation was carried out as follows:

- Requirement not fulfilled “--” (2 minus points)
- Requirement is not sufficiently met “-” (1 minus point)
- Requirement is largely met “+” (1 plus point)
- Requirement is fully met “++” (2 plus points)

Table 25: Criteria for the successful integration of BPMN and DMN into KBE development

Criteria	**Source**	**Note**	**Evaluation**	**Weighting**
System configuration				
Separate knowledge base	[112,271]	BPMN and DMN	+	3
Separate reasoning mechanism/ inference engine	[112,271,284,293]	Workflow engine	+	3
Provides user interface	[284,293]	Graphical representation and possibility to interact with webpages via interfaces	++	3
Provides explanation facility	[284,293]	Graphical notation and reporting of process steps is possible	++	3
Capability to capture, formalize and provide knowledge	[271,284,293,310,311]	Process knowledge with BPMN, decision support with DMN	++	3

Criteria	**Source**	**Note**	**Evaluation**	**Weighting**
Supported Features				
Support of different rules types	[112]			
- logic rules	[112]	Possible: JavaScript and java available	++	2
- math rules	[112]	Possible: JavaScript and java available	++	2
- geometry manipulation rules (parametric rules)	[112]	The parametric rules of the Design authoring system cannot be manipulated, since the KBE system is not integrated. The geometric rules have to be developed in the coupled KBE system.	--	1
- configuration selection rules (or topology rules; combination of mathematical and logic rules)	[112]		++	2
- communication rules (interfaces)	[112]	Possible, but interfaces have to be implemented	+	2
ReconFigure rules and outputs based on new inputs	[310]	Possible, but only manually	-	1
Derive new rules automatically from old rules based on input changes	[310]		--	1
Intelligently control rule sequencing and execution	[310]		++	1
Provides a mediating language between experts, knowledge engineers and software developers, simple yet expressive and preferably graphical	[311]	Graphical notation	++	3

Criteria	Source	Note	Evaluation	Weighting
Expert/end user involvement	[271]		+	2
User extensible and customizable	[311]		+	1
Formal enough to support code generation	[311]	BPMN and DMN are developed to be executable	++	3
Allow models to be organized according to local style	[311]		+	1
Automate processes in a product development lifecycle	[310]	Is possible using workflow engine	++	2
Verify designs against standards	[310]	Verification is limited to the designed process, no additional checks are executed automatically	-	1
Handle new known and unknown problems	[310]	Unknown problems cannot be solved	-	1
Provide high level commands that invoke a number of sub-processes	[310]	Due to the representation of workflows, it is possible to invoke tasks and sub-processes	++	1
Adequate definition of activities	[111]		++	3
Benefits of KBE				
Decrease time and costs for new product development	[111,128,271,272,310]		++	2
Improve quality of design analysis, decisions and ensure consistent quality of outputs	[271,272,310]		+	2
Documentation				
Personalization and codification of knowledge, “learning by doing” (the knowledge must be personalized: it must be geared towards the end	[271]	Graphical representation makes it possible to convey knowledge to the end user in an understandable manner. It is also possible to update	+	1

Criteria	**Source**	**Note**	**Evaluation**	**Weighting**
user(s), who must be able to retrieve, understand and if necessary, update the knowledge used for design and analysis)		the knowledge easily. It is not possible for the system to update knowledge automatically.		
Users can inspect the steps in the design or analysis process, and can see the associated knowledge through the related knowledge component(s)	[271]		++	3
Consistency of representation	[111]		++	2
Methodology worthy becoming a standard	[111]	BPMN and DMN are already standards	++	3
Provision of case reports (input, output and used knowledge is listed)	[111,271]	Standardized reports have to be implemented in the system	+	1
Goes beyond “black box” processes and applications by supporting categorization, accessibility, traceability and subsequent sourcing of knowledge	[271,312]		++	2

A total of 31 evaluation criteria were identified in the course of literature research and have been applied to evaluate the approach tested in this paper. Comparing the evaluated score of 36 points against the maximum achievable score of 62 points, the approach discussed in this paper fulfils 58% of the requirements that are not weighted against each other. Taking into account the weighting of the criteria applied for this study, the BPMN/DMN approach in this study fulfils 73 % of the requirements, which can be considered a good value due to the large number of different requirements.

5.3.3 Capture

‘The capture step involves the collection of raw knowledge and transforms it into the first level of MOKA representation – the Informal Model’ [111]. After the sources of knowledge have been identified (see Section 5.3.1), the knowledge they contain must be compiled, analyzed and put into context. According to *Roth et al.* there are different types of knowledge. As described by *Tripathi* [293] knowledge can be collected using various methods, for example using interview techniques, through text analyses, observation techniques and review techniques.

The types of knowledge were systematically examined with the help of interviews. Table 26 shows the structure of the interviews conducted on the respective knowledge types. Open, unstructured interviews were conducted with domain experts, but also with the software developers of the selected design authoring system. The suitability criteria of the domain experts are described in Section 5.4.2.

In addition to the results of the interviews, the interviewees provided additional documents that can be considered as further sources of knowledge.

Table 26: Structure of interviews and knowledge sources

Type of knowledge			**Interview**	**Additional documents/tools**
Practical knowledge ('Know-how')	Specialized knowledge	Methodical	Domain Expert	
			Software developer	Descriptions concerning Design authoring system, Design authoring system
		Normative ('know-why')	Domain expert	Deutsche Bahn AG Guidelines
		Factual ('know-what')	Domain expert	Deutsche Bahn AG Guidelines
	Experience knowledge		Domain expert	–
	Operational knowledge		Software developer	Descriptions concerning Design authoring system, Design authoring system

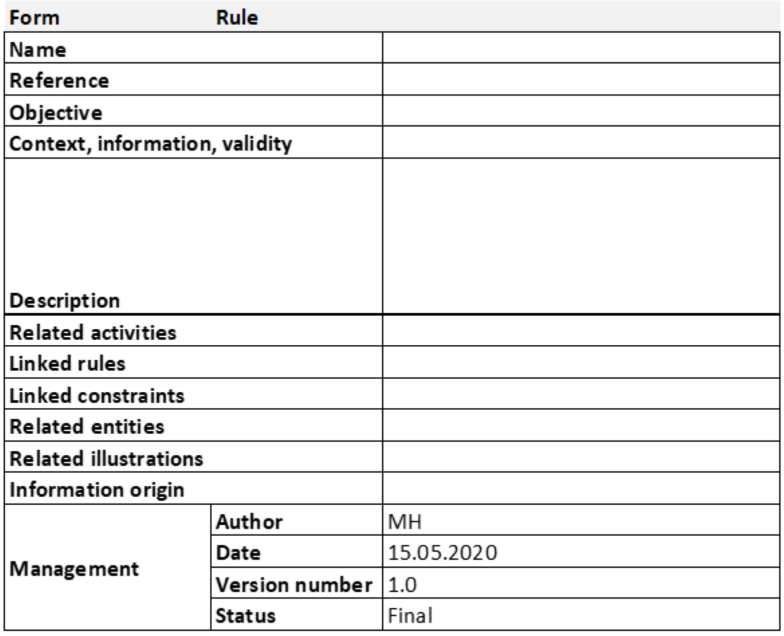
The objective of the interviews with experts was to gather existing technical knowledge for the proposed KBE application. This includes methodological, normative and fact-based knowledge, but also knowledge from experience gained. No pre-existing documents could be provided for methodological knowledge as well as knowledge from experience. In order to make existing knowledge usable for the present study, a scenario for a designing task was developed and the experts were asked to explain the processing and solution path of this step by step. This explanation served as the basis for a first version of a BPMN diagram.

The interviews with the software developers served to understand the software used. They provided insights into the methodical procedure during modelling, but also into the interfacing requirements for connecting the design authoring system to the proposed KBE application (operational knowledge).

The documents made available were then screened and analyzed, and the designing tasks to be solved with the help of KBE were extracted and grouped into knowledge types. Most of the guidelines examined describe the "know-why" and "know-what" while the documents for the Design authoring system describe the operational knowledge.

Based on the steps performed, one can state that the design of railway infrastructure and its associated structures is very strongly parameter-oriented and that numerous geometric dependencies (vertical distance, horizontal distance, height, length, etc.) exist between the individual component and/structures, see also [244]. The most important basis for the design is the alignment, which serves as the defining element for all associated structures. To determine the dependencies between components and assemblies, DSM were developed in conjunction with the experts.

An essential aspect for structuring the collected knowledge is the so-called ICARE forms (Illustrations, Constraints, Activities, Rules and Entities) in MOKA (see Figure 53). These can be used to represent the informal model. All the information for proceeding forward towards the KBE application is thus available. In the study, special attention is paid to how the Design authoring system and the KBE system can be connected and which inputs and outputs are understood or will be generated.



Form	**Rule**	
Name		
Reference		
Objective		
Context, information, validity		
Description		
Related activities		
Linked rules		
Linked constraints		
Related entities		
Related illustrations		
Information origin		
Management	**Author**	MH
	Date	15.05.2020
	Version number	1.0
	Status	Final

Figure 53: An (empty) example of a rule form containing, for example, general information (name, reference number), rule description, and links to other ICARE forms

5.3.4 Formalize

MOKA distinguishes in the formal model between “Product Model” and “Design Process Model”. These are derived from the ICARE forms and both are intended to be represented with the help of UML diagrams. This means, however, that in the next “Package” step, the newly created formal model has to be translated into a KBE language because UML is not developed for direct execution. The Design authoring system used is already a domain-specific specialist application. The UML diagrams serves to visualize the underlying product model and as a basis from which to develop the KBE application. This study, however, focuses on automating the design steps, which can be represented by a “Design Process Model”. This is created by the expert in the “Capture” phase and used and developed to support the transformation process from the informal to the formal model as well as to be transparent for the experts involved. To develop the process model, the contents of the ICARE forms are systematically transferred into BPMN and DMN. In a first step, the Activity forms are evaluated in combination with the created workflow diagrams. These contain links to the Rule and Entity forms. This procedure has been corroborated by *Stokes* [111].

In the course of the expert interviews, numerous references to directives in the railway design sector were communicated. The requirements were documented and recorded in the ICARE forms. For the most part, these guidelines are only available in human-readable form and the requirements are presented in various forms, e.g. continuous text, graphics, tables and formulae, see also [98,244]. To translate them into a machine-readable language, the RASE syntax (**R**equirement, **A**pplies, **S**elect, **E**xception) was used. This has been described comprehensively by *Hjelseth & Nisbet* [90] and *Häußler et al.* also used it to translate the corresponding guidelines. This process has been adapted in the present study using the BPMN elements script task, parallel and exclusive gateway, sequence flows and DMN tasks. Sub-processes are used to visually summarize related activities. As far as multiple but similar activities are included, the sub-processes are designed so that they can be executed several times with the help of loop criteria.

Since the focus is the automation of the design process, it is important to depict not just the technical content of the knowledge sources, but also the methodical approach of the experts and the necessary procedure within the Design authoring system through the BPMN diagrams. An example BPMN diagram is shown in Figure 54.

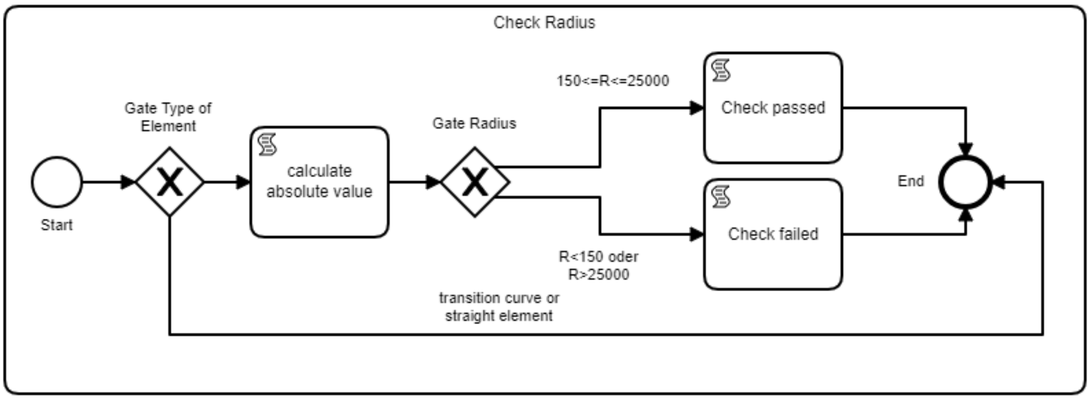


Figure 54: BPMN-Process in the context of code compliance checking [244]

The A (Activities) and R (Rules) forms contain appropriate information. The main objective of the KBE application is to process data from upstream processes in such a way that new data is generated through the execution of the KBE application which can then made available to subsequent processes (see Figure 46). As the aim is to support the user, the interface of the KBE application has been kept to a minimum so that only native data of the Design authoring system is used. Although the interfaces are defined during the formalization process, they are not developed. This is the content of the “Package” step.

The resulting “Design Process Model” was then shown to the experts for joint verification of correctness (see Section 5.4.3).

The advantage here is that no further steps are required to translate the created workflows and decision tables into a neutral data format such as XML, as these are already saved as XML data. As such, this step from MOKA is not required.

5.3.5 Package

In the “Package” step, the formal model is usually translated by a software developer into the programming language of the target platform. Ideally, the formal model saved as XML is used as a basis. Since BPMN and DMN are already designed as executable languages and are stored in XML format, no further translation is required. The package step is therefore used to test the functionality and interaction of the applications used. In particular, the definition of variables is validated and adjusted if necessary.

In addition, a web-based user interface was developed as a direct connection to the KBE system along with the necessary interfaces to the Design authoring system.

The individual components of a KBE application were described earlier in Section 5.2.3. In our case, the following software components were used (see also Figure 55):

Knowledge base: Camunda Modeler [241]
Inference engine: Camunda Community Platform (workflow engine) [241]
Explanation facility: Web services – webpage and Camunda Community Platform [241]
User interface: ProVI 6.2 [38] and webpage (interface to KBE system)

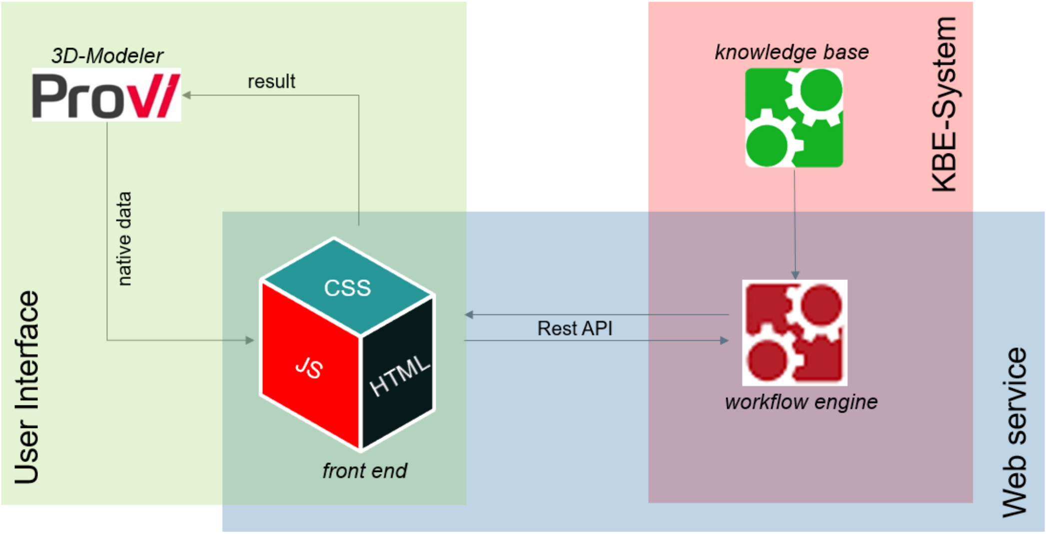


Figure 55: Software configuration

The communication between the user interface and the KBE system running on a server is facilitated by a Rest API.

5.4 Case study: The design of noise barriers

5.4.1 Identify and justify

A key aspect in the design of railway facilities is to reduce the noise produced by rail traffic [313,314], for example by erecting noise barriers alongside railway lines. The construction and the individual components of noise barriers are highly standardized, which lends itself towards automation of the design process. The KBE application proposed here as a case study is intended to support the geometric design of a noise barrier by automating the steps that are carried out manually. Existing automation steps already available in specialist applications do not need to be replicated. For example, the specialist app already models the noise barrier as a parametric model so that the KBE application only needs to determine the required input parameters. The KBE applications should nevertheless be generic enough so that different variants of the same noise barrier can be generated, but also different noise barriers for different projects and project areas.

The software configuration has already been described in Section 5.3.5, and the acceptance criteria for this study were evaluated in Section 5.3.2. The evaluation discussed there applies unchanged for the specific task at hand.

5.4.2 Capture

For knowledge acquisition, five domain experts from different companies were consulted in order to ensure a cross-company approach. The domain experts have more than 5 years of professional experience and are classified as senior engineers in their companies. In addition, a software developer of the Design authoring system employed was interviewed for the capture phase.

The interviews took between two and three hours per domain expert (only one expert per interview). The experts were asked to explain their design workflow without any interposed question first to get an unaffected description. The experts described the constraints in design, used input data and desired outcome. All of the experts use the same Design authoring system and expressed that no comparable system for the design of noise barriers is known to them. The parametric logic of a noise barrier was described as well as the interdependency to other facilities along railway tracks. In a second step, the idea of a KBE system was introduced by the knowledge engineer and the experts were asked which part of the design process is worth to automate. The experts provided different documents concerning the design of noise barriers. For this study the same Design authoring system was used as the one indicated by the domain experts. The interview with the software developer focused on the operation of the Design authoring system, provided interfaces were discussed as well as possible extensions for the software.

Based on the interviews, the relevant guidelines were reviewed.

Various disciplines are involved in the design of a noise barrier, such as sound calculations, structural analysis and geometric design. The interviews showed that the result of noise calculations is a key input variable for the geometric design of noise barriers. The calculation is carried out by environmental engineers using specialized software and essentially provides information on the height and length of a noise barrier. The structural analysis is a downstream process for the geometric design of the noise barrier and is carried out by structural engineers. The aim of the test case outlined here is to support the design engineers in the routine tasks of geometric design.

The main results of the knowledge acquisition are summarized below. A noise barrier consists of four main components: post (1), wall element (2), base (3) and foundation (4), as shown in Figure 56. In addition, the noise barrier is dependent on various input variables. The distance from the front edge of the noise barrier relates to the alignment of the track as set out in Deutsche Bahn guideline 800.0130 [240]. An explanation is given in Section 5.4.3.2.

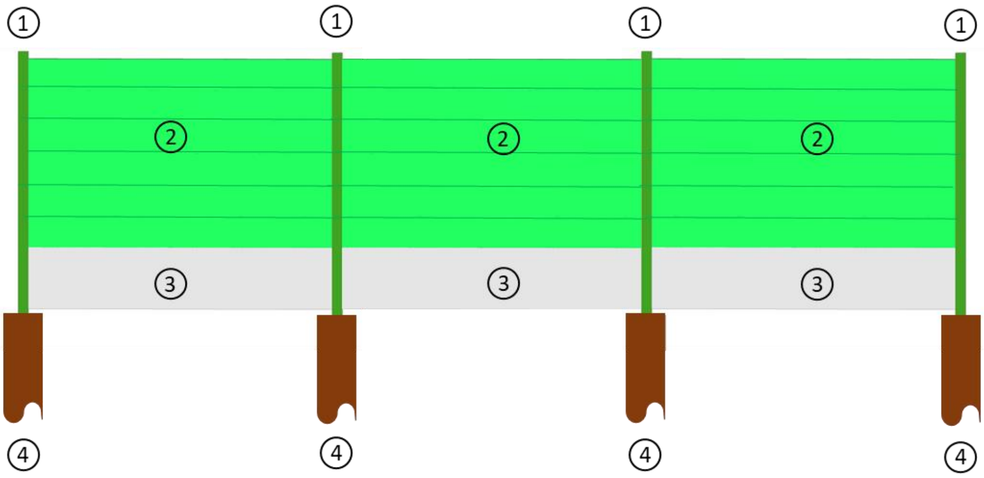


Figure 56: Design of a noise barrier

Various structures exist along railway tracks that influence the design of noise barriers. These include drainage, underground cable conduits, overhead line systems and components for signaling and safety technology, as shown in Figure 57 situation 1. These can obstruct the continuous construction of a noise barrier parallel to the track (Figure 57 situation 2). Table 27 shows the interdependencies at the level of the installation in the form of a DSM. For all the subsections considered, the alignment therefore serves as an input variable, and for the noise barrier in particular, the subsections listed are likewise input variables. The design of a noise barrier must therefore be able to respond flexibly to other subsections by means of so-called by-passes that avoid collisions with installations from other subsections. Figure 57 situation 3 shows how that noise barrier steps back to avoid colliding with a catenary mast and manhole.

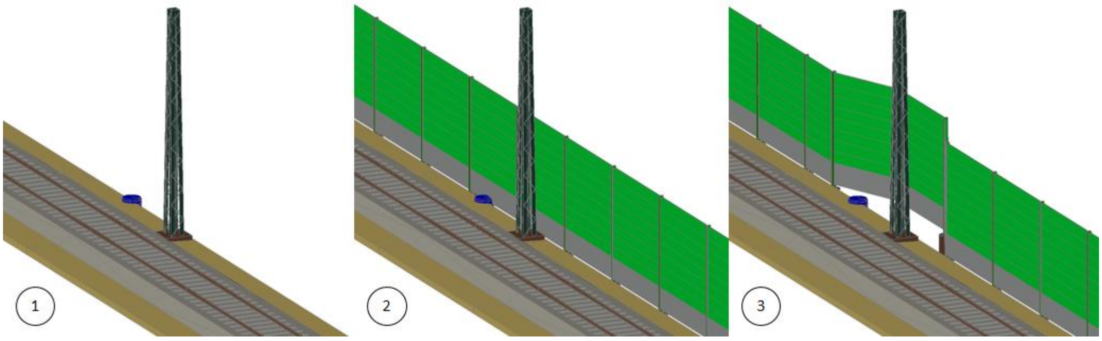


Figure 57: Railway track model; situation 1 – railway with associated installations; situation 2 – railway with installations and a parallel noise barrier; situation 3 – with installations and noise barrier by-pass.

Table 27: DSM for noise barriers

		Output					
		alignment	noise barrier	catenary	signaling	drainage	cable system
Input	alignment		x	x	x	x	x
	noise barrier						
	catenary		x		x	x	x
	signaling		x	x		x	x
	drainage		x				x
	cable system		x			x	

The Design authoring system can work with parametric model data in software-specific formats for alignment as well as for the installation of overhead line and drainage systems. For formalizing the informal model, the approach discussed here is limited to the consideration of catenary masts and manholes.

A methodical description of the procedure was developed in cooperation with the experts as a draft BPMN workflow shown in Figure 58. The design engineer begins by evaluating the route alignment, checking the constraint points and the distance of the noise barrier from the parallel track lines. In a second step, the engineer checks the structural data of the influencing subsections (here: overhead line and drainage), determines the positions of the overhead line masts and manholes, and plans any necessary by-passes at these points. Since these work steps have to be carried out repeatedly for numerous instances of these objects, the tasks must be conceived as a loop.

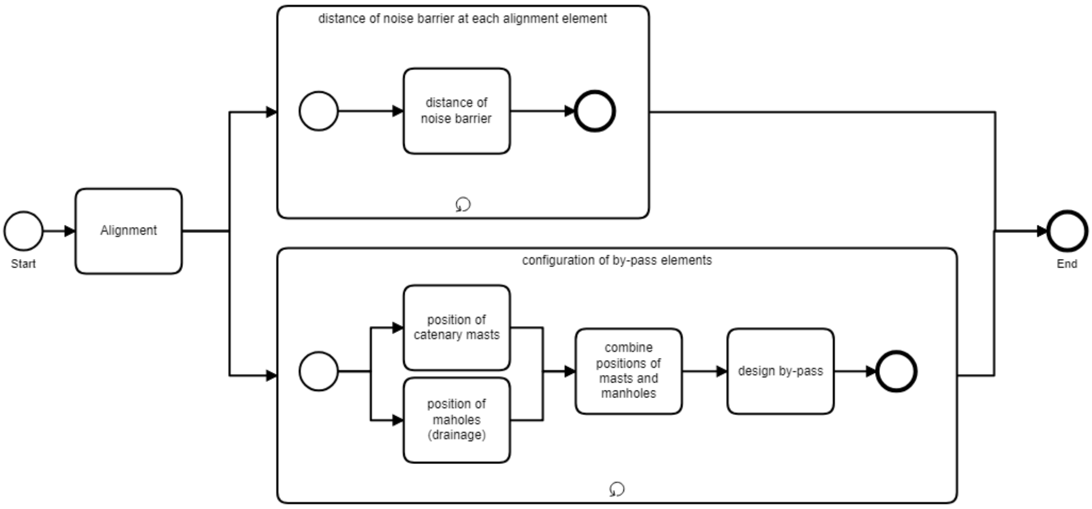


Figure 58: BPMN workflow of the design process of a noise barrier, created by an expert

This workflow serves as a basis for the Activity forms, of which the "distance of a noise barrier" is shown in Table 28 by way of example. In this study, the combination of standardized forms and the process-related representation using BPMN proved to be helpful as a means of facilitating communication between the participants.

Table 28: Example of MOKA activity form, definition of noise barrier distance

Form	**Activity**	
Name		Definition of noise barrier distance, import data
Reference		A_D_1
Trigger		Completion of alignment
Input		Alignment data
Output		List of parameters with noise barrier distance at each alignment element
Potential failure modes		Needed data is incomplete
Objective		Designing the distance of a noise barrier
Input requirements		Input data from design authoring system must include information about: station, radius, cant, pace, number of tracks, side of noise barrier
Context, information, validity		▨
Description		This activity aims to describe the import data for alignment.
Related Activities		A_D_2
Rules involved		R_D_1
Entities involved		ES_NB_1
Related Illustrations		▨
Information Origin		Guideline 800.130 A07, description of design authoring system
Management	**Author**	MH
	Date	15.05.2020
	Version Number	1.0
	Status	Final

5.4.3 Formalize

The ICARE forms created above, along with the outline workflow are developed in greater detail in the MOKA "Formalize" step. First, the product model is designed as a UML diagram. Two sub-processes are derived from the workflow, both of which are based on the alignment data of the project, but can also be viewed one after the other, as per the methodical procedure of the experts as well as within the Design authoring system. The sub-processes "Determining the distance" and "Defining the by-passes" are described in the following sections.

5.4.3.1 Product model

The product model of a noise barrier was developed as a UML diagram, shown in Figure 59. The UML diagram shows the structural design of a noise barrier and its individual components (as per Figure 56) and assigns attributes to the classes that need to be considered when designing a noise barrier. To clarify the origin of the data, the attributes are given a color. In this paper, the focus lies on the components relevant for the geometric design and their parameters, which are shown in red.

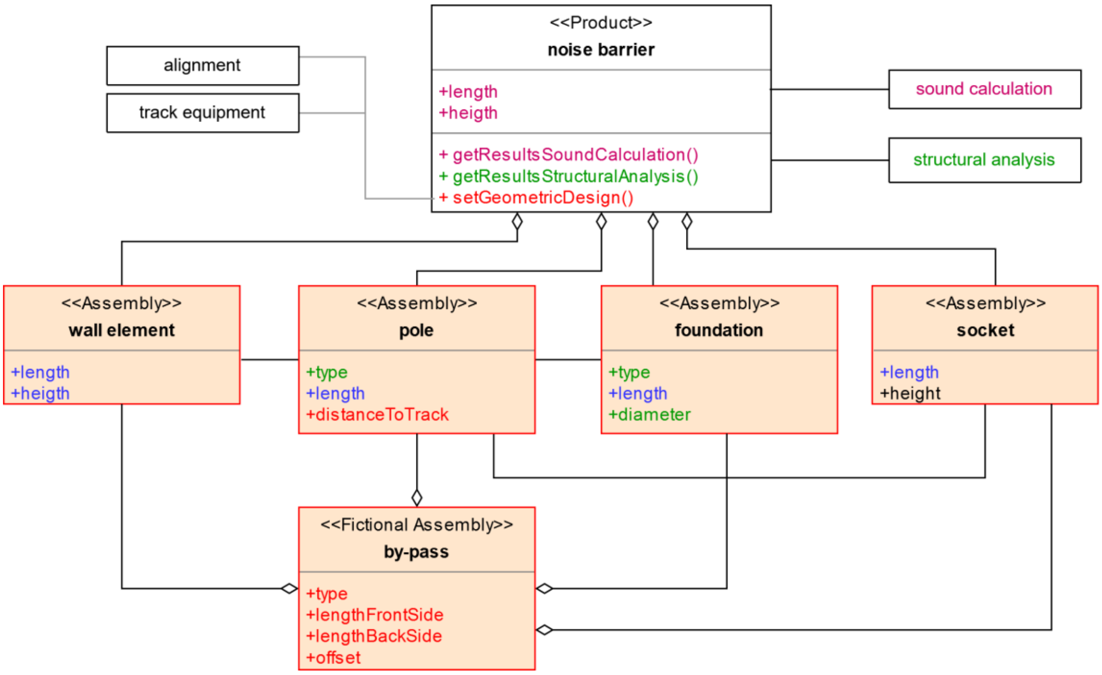


Figure 59: UML diagram of a noise barrier. Sound calculation (purple) and structural analysis (green) are inputs for the geometric design (red). The blue attributes are calculated by the Design authoring system automatically, and the red attributes by the KBE application.

5.4.3.2 Design process model: Definition of distance to track

The most important input variable for determining the distance of a noise barrier to the track is the track alignment. The relevant reference object for determining the position is the axis which is comprised of straight line, circular arc and transition curve segments as shown in Figure 60.

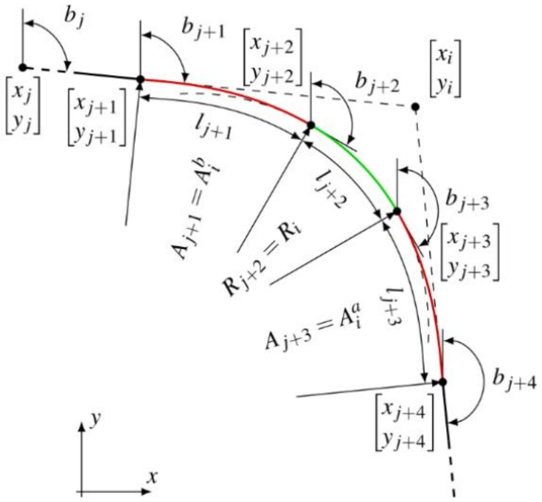


Figure 60: Different types of segments along a horizontal alignment: straight elements in black, transition curves in red, circular arc in green (after an original in [315]*)*

Directive 800.0130 Annex 07 defines the necessary distance between noise barriers and track alignment axes, and this has already been recorded as a Rule form in the informal model. The table contained in the directive is evaluated as a decision table based on its structure and the clear definition of input and output variables, which makes it possible to formalize it as a DMN decision table shown in Table 29.

Table 29: Decision table for the distance between a noise barrier and alignment axis

Input				Output
Cant **[mm]**	**Number of tracks**	**Pace** **[km/h]**	**Side of arc** **[-]**	**Distance** **[m]**
[0 .. 160]	1	<= 160	-	3.3
[0 .. 160]	1	> 160	-	3.8
[0 .. 160]	2	<= 160	inside	3.3
[0 .. 20]	2	<= 160	outside	3.3
[25 .. 50]	2	<= 160	outside	3.4
[55 .. 100]	2	<= 160	outside	3.55
[105 .. 160]	2	<= 160	outside	3.7
[0 .. 160]	2	> 160	inside	3.8
[0 .. 20]	2	> 160	outside	3.8
[25 .. 50]	2	> 160	outside	3.9
[55 .. 100]	2	> 160	outside	4.0
[105 .. 160]	2	> 160	outside	4.2

This was used as basis for checking the design input data, e.g. to determine if the necessary input variables have been provided. The alignment data includes information on the cant and speed of the axis elements, but the number of tracks can only be specified by the user as there are no other data sources for this. A corresponding input interface was developed as part of the subsequent “Package” step. By using the Activity and Rule forms as well as the decision table, it became clear that a further key variable for automating the execution of the decision table was the position of the noise barrier in relation to the axis, e.g. on which side of the arc the noise barrier lies (left or right of the track). This, too, must be defined by the user. Figure 61 shows the possible situations. A special case in this context are straight sections of track: these have no curvature and are thus not explicitly detailed in the guidelines. The interviews with experts revealed that these are typically treated as “inside arcs”. As the route alignment data also contains information on the direction of curvature, the following distinction can be made:

- Radius < 0 → left curved
- Radius > 0 → right curved
- Radius = 0 → not curved

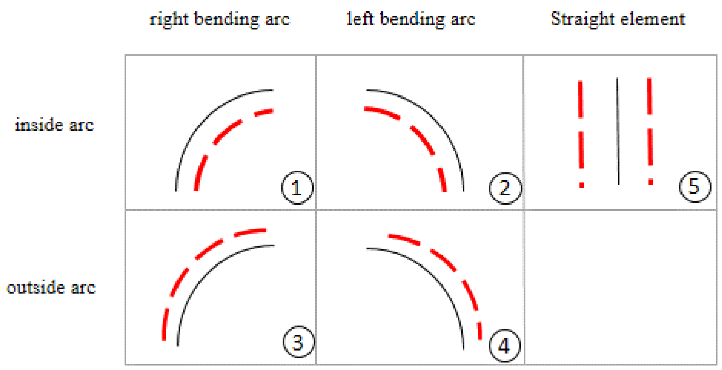


Figure 61: Relation of noise barrier (red, broken line) to arc element (black, continuous line)

These decision processes can be mapped using BPMN as shown in Figure 62, where the numbers correspond to the different situations shown in Figure 61. The DMN task “Define necessary distance” represents the decision table shown in Table 29 as a BPMN node.

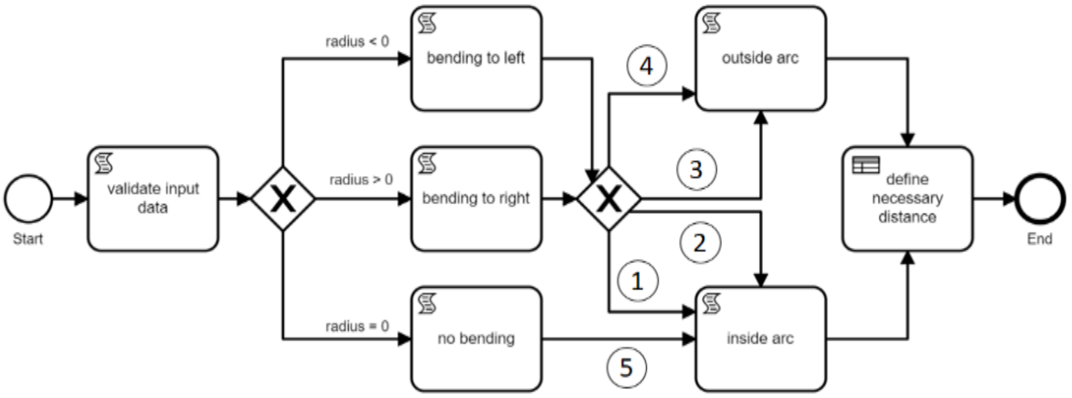


Figure 62: Workflow for defining the necessary distance at each alignment element

5.4.3.3 Design process model: Definition of by-pass parameters

Alongside determining the distance between the noise barrier and the track alignment axis, the “Capture” phase and the DSM also made it clear that design of noise barriers is subordinate to other subsections such as the design of overhead lines or drainage. These must therefore also serve as input variables for the design of the noise barrier. The next step in the design of noise barriers is therefore to ensure it does not collide with existing installations along the railway line.

The logic of drainage systems means that the data provided by the Design authoring system includes the entire sewer network. Manholes do not necessarily have to collide with the noise barrier. The same also applies to catenary masts for overhead line design. The masts and drainage elements can be positioned on either side of the track and in some cases may not clash with the noise barrier at all. The first step is, therefore, to filter out only those objects that may influence the course of the noise barrier, i.e. those elements located on the same side of the tracks as the noise barrier (see Figure 63).

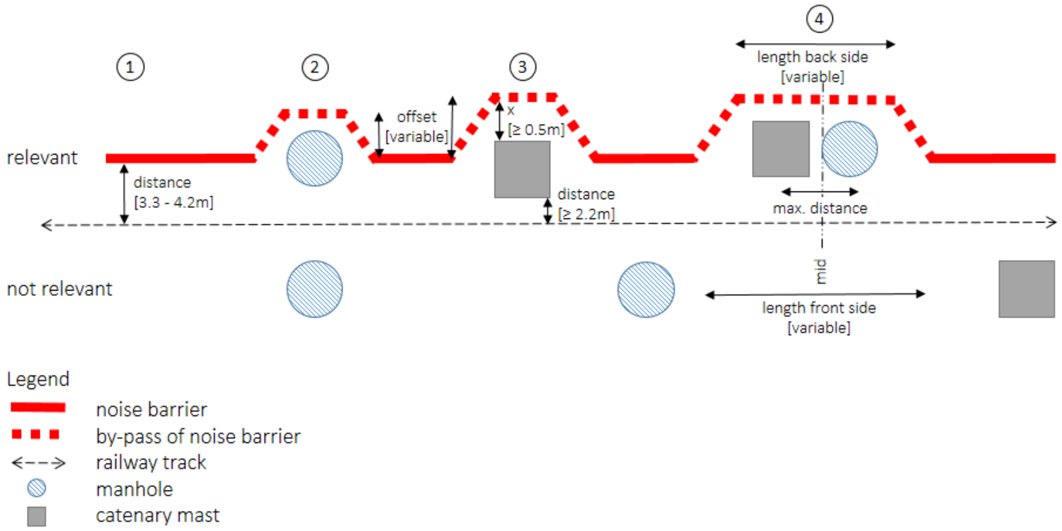


Figure 63: Sketch of a noise barrier with by-passes. Four situations are shown: 1) without obstacle, 2) collision with a manhole, 3) collision with a mast, 4) by-pass around several obstructions.

The next step is to check which of the filtered objects actually influence the design of the noise barrier and to what extent. The Design authoring system can provide the following parameters on manholes and catenary mast objects:

- Object type
- Station on the track
- Distance of the center of the object to the track
- Dimensions of the object (height, length, width, diameter etc.)

The objects are initially considered independently of each other and it is assumed that each of the objects will result in a single by-pass segment in the noise barrier. The aim of the automation is to determine the length of the respective by-pass around the objects encountered. Catenary masts restrict the space available next to the tracks. The interviews with experts revealed that space must be left for the safe passage of maintenance personnel next to the tracks. The process must therefore determine whether the space between the mast and tracks is sufficient or whether service staff will need to pass behind the mast (see Figure 63 situation 3). This can be determined with the help of a simple decision table (see Table 30) and requires only the “distance between track and the front edge of the catenary mast” as an input value. If this value is less than 3.3 m, the distance between the rear edge of the mast and the noise barrier must be at least 0.8 m, otherwise 0.5 m (the dimension “x” in Figure 63 situation 3). In the case of the manhole, this is not relevant for the staff maintenance route as the manhole cover is at ground level and does not obstruct the path passing over it (see Figure 63 situation 2). Section 5.4.3.2 describes how the distance of the noise barrier to the track axis is determined (Figure 63 situation 1). This is in relation to the main elements of the underlying track alignment. In addition to the determined position of the individual relevant objects and the regulatory information on the space to be kept free, the distances from the axis to the front edge of the noise barrier is calculated at each catenary mast or manhole using the same logic as described in Section 5.4.3.2. With this information, the necessary offset of the noise barrier around manholes and catenary masts can be determined. The corresponding section of the BPMN workflow is shown in Figure 64.

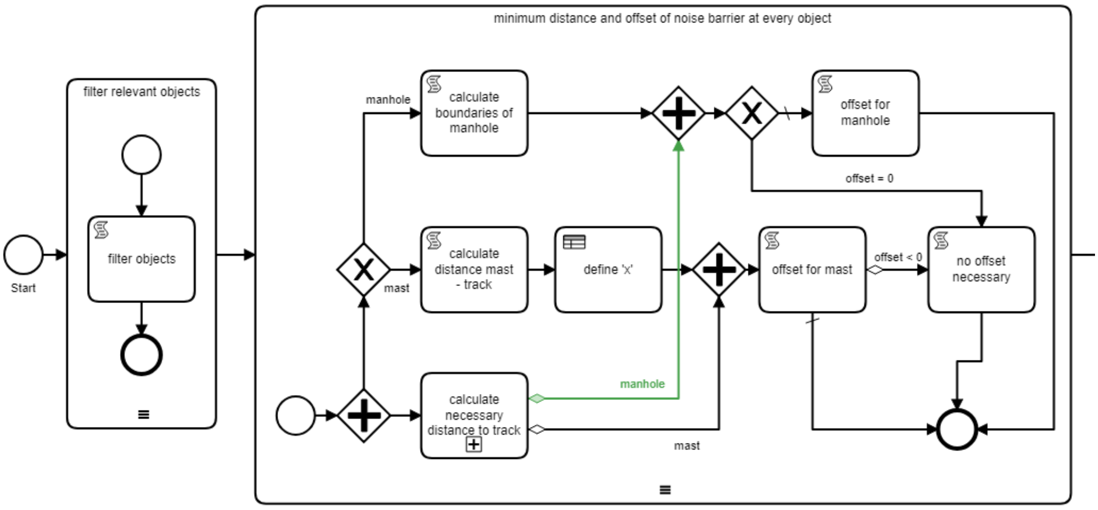


Figure 64: BPMN diagram for the design of by-passes part I: Filter relevant objects, minimum distance and offset of noise barrier at every object

Table 30: Decision table for the definition of ‘x’ as necessary distance between foundation and wall

Input	**Output**
Distance track to foundation [m]	**Distance foundation to wall ‘x’ [m]**
< 3.3	0.8
≥ 3.3	0.5

Using the steps described, the data available from the implicit model was used to generate information and knowledge for the course of the noise barrier. The assumption up to now has been that each manhole or mast will result in an independent bypass. The next step is to determine whether several small by-passes can be combined to form a single larger by-pass (Figure 63 situation 4), for example because geometrically expedient or for cost reasons. To this end, an additional script task was integrated into the process to identify the spatial proximity of different objects. This script task checks the distance between objects based on a maximum distance specified by the user. By varying this input variable, different variants of the same noise barrier can be generated. The requisite operators for this were combined in a single task to avoid a complex and incomprehensible BPMN diagram. *Preidel* calls this form of representation the “atomic method” [4]. Once close-by objects – i.e. objects that can be combined in a single by-pass – have been identified, the actual parameters of the by-pass can be determined in an independent sub-process. First, the mid-point of the by-pass is determined from the minimum and maximum station of the objects to be considered. Based on whether closed or open by-passes are to be designed (a user-specified input), additional parameters are determined for “length front” and “length rear”. This enables the user to consider further variants. The corresponding section of the BPMN workflow is shown in Figure 65.

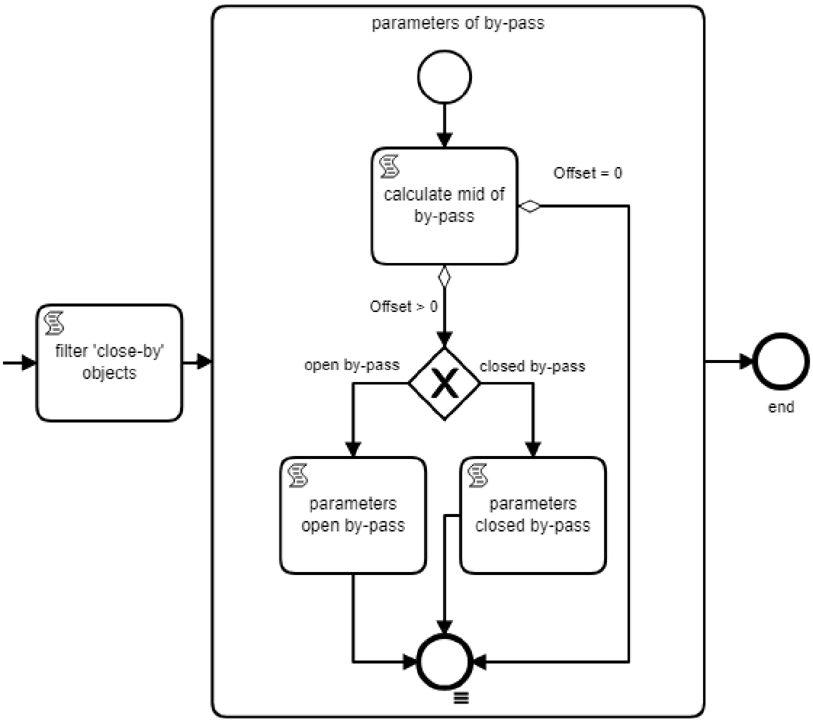


Figure 65: BPMN diagram for by-pass design Part II: Filter “close-by” objects, parameters of by-pass

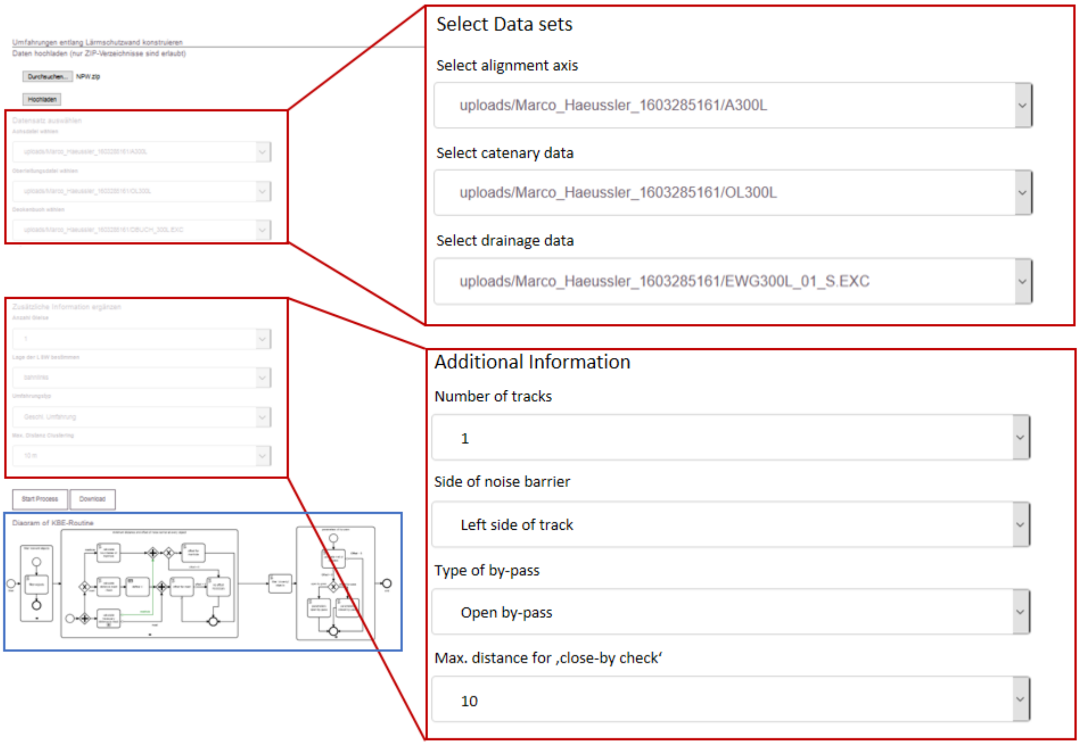
The result of this process provides the parameters necessary for the design of by-passes. For this purpose, input data was processed, and new data, information and knowledge was generated. Data required for the Design authoring system can also be passed to it via corresponding interfaces.

Although in MOKA, the “Capture” and “Formalize” phases follow each other, the authors found that questions arising in the course of formalization can make it necessary to repeat individual steps of the capture phase. The authors therefore see these phases as being iterative.

5.4.4 Package

With the help of BPMN and DMN, the knowledge identified in the “Capture” phase could be digitally mapped in the “Formalize” phase. In this final step, automated systems need to be developed to extract relevant data from the underlying design input data, which is available as tabular data in text files. Since both the Design authoring system and the KBE system have a web interface, the programming language JavaScript can be used for data extraction. As the workflow engine can also interpret JavaScript, the entire KBE application has a consistent programming language. The web-based user interface is shown in Figure 66.

The user must first select the data sets to be evaluated and then define boundary conditions applicable to the process, such as the number of tracks, and on which side of the tracks the noise barrier is needed, as described earlier. The website allows the user to choose using simple select dropdowns. The BPMN process is also shown.



Select Data sets

Select alignment axis: uploads/Marco_Haeussler_1603285161/A300L

Select catenary data: uploads/Marco_Haeussler_1603285161/OL300L

Select drainage data: uploads/Marco_Haeussler_1603285161/EWG300L_01_S.EXC

Additional Information

Number of tracks: 1

Side of noise barrier: Left side of track

Type of by-pass: Open by-pass

Max. distance for ‚close-by check‘: 10

Figure 66: Web-based user interface for choosing the data to submit to the KBE system (server-side) and setting additional user inputs via dropdowns (red). The BPNM process diagram is also displayed to the user (blue).

In addition, the generated process diagrams are extended in the “Package” step so that they can be executed by the workflow engine used. The variable definitions made need to be checked and adapted if necessary, and the generated data for passing back to the Design authoring system must be converted into the required Comma-separated values format (CSV). For each station (one line per axis element or by-pass) the corresponding parameters are arranged in a pre-defined column sequence. This data can be downloaded from the server by the user and imported back into the Design authoring system. Due to the lack of an API, it is not possible to connect the Design authoring system directly to the KBE application.

Over the course of developing the executable KBE application, one can also see that the “Formalize” and “Package” phases are also closely related and influence each other. This is primarily because using

BPMN and DMN makes it possible to produce a consistent logic with an executable language so that the conditions for successful execution can already by determined during the "Formalize" phase. The interaction between the different phases is a positive characteristic in the authors' view. Figure 67 shows the interrelationships of the "Capture", "Formalize" and "Package" phases of the MOKA method in the case study project.

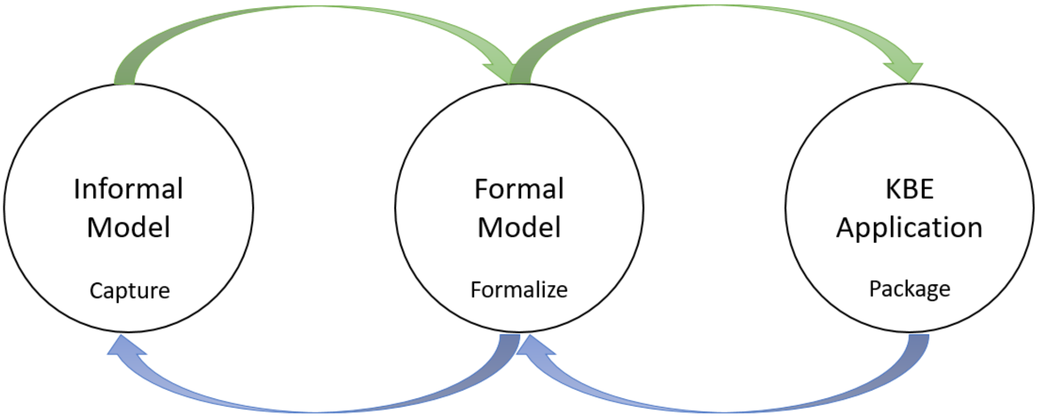


Figure 67: Relationships between the MOKA phases in the case study

5.4.5 Results

In this study, the aim was to use a uniform presentation form for the activity diagrams from the "Capture" to "Package" phases. As BPMN workflows in combination with DMN decision tables can be made executable by means of a workflow engine, this approach could be used for these phases of this study. The acceptance criteria outlined for this study in Section 5.3.2 were evaluated for the method used. One of the most important challenges in connection with KBE is that the automation of routine tasks should reduce the processing time and, in turn, production costs. To quantify these results, the processing times for the semi-automated (by using functionalities of the Design authoring system) design of a noise barrier were compared with the KBE-based approach. To determine the time required for semi-automated design, we assumed the user would use the same Design authoring system used in the KBE-based approach. The initial steps for generating input data are the same in both approaches and were therefore not compared. The following data was used as a basis for the design:

- Route axis with a length of 2,700 m
- Sewer system with 102 manholes
- Catenary data with 21 catenary masts

The noise barrier should extend the length of the railway track, i.e. also have a length of 2,700 m.

Table 31 compares the times required to design a noise barrier semi-automated and with the help of the automated KBE approach. The processing steps compared correspond to the procedure described above and the semi-automated processing time are empirical values estimated by five domain experts.

The experts came up with close-by estimates independently of each other. According to their estimates, the semi-automated design of the noise barrier takes about 145 min on average. These times are based on routine tasks only; special situations such as designing noise barrier that pass over a bridge structure, are not considered here. By comparison, the KBE-based approach is significantly faster, taking only 3 minutes 40 seconds, which corresponds to a ratio of 2.5 % of the time.

Table 31: Comparison of the semi-automated and automated design of a noise barrier

Phase	Description	Time needed for semi-automated design	Time needed for automated design	Ratio
1	Analysis of underlying data	17 min	Included in 2-4	-
2	Import to KBE system	-	1 min	-
3	Definition of noise barrier distance	20 min	20 sec	1.5%
4	Design of by-passes	108 min	2 min	1.9%
5	Export to Design authoring system	-	20 sec	-
	Total	**145 min**	**3 min 40 sec**	**2.5%**

Variants are correspondingly quicker to produce: the user can, for example, decide at the beginning of the process whether the noise barrier should be designed with open or closed by-passes. With the KBE-based approach, the “type of by-pass” selection can be adjusted accordingly, and the system determines the configuration within 3 minutes 20 seconds. A design engineer would require a further 40 minutes to adapt the previously created design. This corresponds to an approx. 92 % improvement in efficiency when developing variants or adjustments to the design.

The entire process of designing the executable KBE application from the “Capture” to the “Package” phase took about 4 weeks in the described example.

5.5 Conclusion

This paper has presented a study for a KBE system based on the graphical notation BPMN in combination with DMN. The development of the application followed the general principles of MOKA which is already well established in practice. MOKA aims to collect knowledge in different ways in successive phases and to transfer it from an initially unstructured form to a KBE system. A disadvantage of MOKA in the opinion of the authors is that there is no uniform notation. The results of each individual phase always need to be converted into a different data format for the next, i.e. from ICARE forms to UML diagrams and from UML diagrams to a KBE language. The aim of this study was to use a consistently uniform presentation form for the activity diagrams. To this end, BPMN and DMN were used as two notations that can be used for the graphical representation of knowledge and are also executable with the help of an associated workflow and decision engine. The case study examined an application in the field of railway infrastructure design for new buildings. As part of the “Justify” phase, 31 acceptance criteria were compiled from the literature and the approach presented here was evaluated against them: the results show that the approach fulfils 58 % (unweighted) and 73 % (weighted) of the criteria. During the “Capture” phase, the available sources of knowledge (expert knowledge, guidelines, functionality of the applications used) were collected and structured with the help of ICARE forms, DSM and BPMN diagrams. In the next step, the informal model was formalized using methods described in earlier work by *Häußler et al.* to formalize the knowledge contained in guidelines with the help of BPMN and DMN. The methodical knowledge of the experts and the requirements of the specialized applications can likewise be formalized with the help of BPMN and DMN.

The case study examined the automation of the geometric design of a new noise barrier along railway tracks and demonstrates that BPMN and DMN can be applied consistently from the “Capture” phase to the actual KBE application. By automating the processing steps, the time required for the design of noise barriers under the given boundary conditions can be reduced many times over: in the case study, the KBE approach requires only 2.5 % of the time required for semi-automated design. The creation of different variants is also significantly faster when using automated processes. In the example shown, 8 % of the semi-automated processing time was required to develop an additional variant of an existing noise barrier design.

In addition to significantly shortening the required generation time, the approach developed here also demonstrates the graphical representation of different types of knowledge in connection with the process-based approach to the design of railway infrastructure facilities. The process is transparent, meaning that the KBE system is not a “black box” solution, which can in turn benefit user acceptance. In addition, the system can be adapted to meet different requirements.

Apart from these advantages, however, there are also some aspects that the approach presented here cannot solve, or only insufficiently address. The KBE application is developed as a system decoupled from the Design authoring system. This makes it necessary to develop appropriate interfaces for importing and exporting data, which may need to be updated as the Design authoring system is developed. Currently, the data must be actively transferred to the KBE solution by the user, which introduces a potential point of error. With the help of an API, the KBE system could in future be directly linked to the Design authoring system. This would also eliminate a further disadvantage: the user would not have to leave the Design authoring system and the result would be displayed in the Design authoring system without further action. In the current prototype, the approach is limited to the knowledge provided by the engineer and the routines devised by the developer. The system is not able to learn independently from the tasks set and find new solutions to problems. For highly standardized structures – such as the example of the noise barrier – this approach is, in the authors' opinion, quite sufficient. But if more complex, highly specific solutions need to be developed, e.g. escape routes and emergency services access in dense inner-city environments, or the fitting of railway platforms, this approach may currently be too limited.

The case study presented here is limited to the geometric design of new noise barriers. Future investigations should aim to develop cross-domain solutions. Using the noise barrier as an example, this would entail linking together sound calculations, structural analysis and geometric design. This could be achieved, for example, by triggering sophisticated calculation and simulation applications from a workflow or by modelling the calculations and decision paths using BPMN and DMN.

As the field of railway infrastructure design is particularly well defined by standards and regulations, the approach shown here offers very good and valuable support for routine tasks. Further fields of application could be the automated design of superstructures, underground cable routing, platforms and engineering structures such as bridges and tunnels. Applications are also conceivable in road construction, e.g. in the design of road cross sections (road superstructure, road widths). BPMN and DMN methods could also be used for the design of road junctions, though here stronger interaction with Design authoring system would be necessary.

6 Discussion and Outlook

Quality improvement is one of the objectives associated with the introduction of the BIM method in the construction industry. The aim of this dissertation has been to develop methods for the model-based quality assurance of railway infrastructure design. Two approaches were followed, as presented in chapters 3 to 5:

Approach 1 – Checking existing models	(Chapter 3 and 4)
Approach 2 – Creation new models	(Chapter 5)

The dissertation focusses on the design of traffic facilities and civil engineering structures in the context of railway infrastructure.

This chapter summarizes the main findings of the research work. The conclusion is divided into the approaches of the work and the investigated domains. Further research needs or possible extensions of the methods are discussed in section 6.3.

In practice, the quality of railway models is still predominantly assured by manual methods in a downstream checking process. This approach is labor-intensive and only contributes to efficiency gains to a limited extent. Practical, partially automated approaches such as 3D clash detection illustrate the potential that the automation of processes can bring in context of quality assurance. A high degree of automation allows routine tasks to be processed and completed more efficiently. For this purpose, it is necessary for both the model information and the rules to be checked to be available in a machine-readable and thus digitally analyzable form. The dissertation investigated how both aspects can be fulfilled.

6.1 Conclusions

The BIM method undoubtedly contributes to improving the quality of design. However, this goal is not achieved by applying the BIM methodology alone, but rather through consistent and (partially) automated checking of the model information, where the model was created manually. The goal of downstream model checking is to find errors that occurred during the modeling process. In addition to automating the checking process, digital methods were also introduced to create models in compliance with guidelines (upstream process). This allows models to be quality assured at an early stage in the design process.

The approach-related findings of the work are listed in the following sections. Here, the objectives along with corresponding hypotheses and their treatment in the thesis are discussed. These are:

Objective 1: Development of a concept for model-based quality assurance.

> **Hypotheses 1:** Several domains must be considered when checking models.

Objective 2: Model checking based on explicit geometry description.

> **Hypotheses 2:** Checking of model semantics is a central aspect.

Objective 3: Model checking based on implicit geometry description.

> **Hypotheses 3:** Using implicit geometry description allows easier compliance checking.

Objective 4: Creation of geometric models using methods of downstream quality assurance.

> **Hypotheses 4:** Compliance checking methods of parameterized models can also be used for model creation.

In summary, the work conducted in this dissertation confirmed all the hypotheses, and the objectives were achieved.

6.1.1 Approach 1 – Model checking

6.1.1.1 Quality assurance concept

The literature research in the context of BIM and quality assurance has shown that numerous approaches to model checking exist. Existing studies focus predominantly on the checking of building models, both in terms of compliance with guidelines and safety on construction sites. Especially in the context of construction site safety, the combination of 3D and 4D models has been investigated.

A study of the existing literature reveals that investigations into the checking of models have different objectives. The studies of *Eastman* [87] and *Preidel* [4] are particularly noteworthy here. From this, it can be deduced that several aspects must be considered in quality assurance.

Only a few studies exist in the field of infrastructure design. To date, there is no overarching approach that incorporates different aspects and perspectives into model checking. The work in this dissertation closes this gap for the field of railway infrastructure and considers different aspects in the context of 3D, 4D and 5D models. Hypothesis 1 could be confirmed, and Objective 1 has been achieved with the development of a cross-domain quality assurance concept.

The concept is modular, so that further quality checks can be added or omitted as required.

6.1.1.2 Clash detection

Automated clash detection is a very common method for checking models and various software products are available on the market for this purpose. Clash detection has therefore also been integrated into the present quality assurance concept. While the available software products already offer a high degree of automation with regard to the identification of clashing objects, the evaluation of the identified clashes is not adequately supported because the evaluation of test results is carried out manually.

In practice, different authoring software is used for the creation of 3D models. In the current state of the art, this means that clashes between the models produced by different software products are partly unavoidable. At the same time, the available software is not always able to produce a clash-free output of 3D models. With the currently available methods for clash detection, all clashes, i.e., component overlaps, are identified, regardless of their importance or their cause.

The work in this dissertation provides methods for the automated evaluation of clash detections. With the help of a clash matrix, component combinations are identified for which software-dependent clashes can be automatically identified and evaluated. The evaluation can be controlled individually. In addition, clashes can be declared as unimportant from a planning point of view, for example because they will not be processed until a later point in the project or because they are completely irrelevant for the project.

Clash tests are usually carried out on a static 3D model. Furthermore, an important objective for the construction realization is the clash-free sequence of the construction technology. Here, however, static

3D models do not provide any information. Using a 4D model it is possible to realize a clash detection of the construction process. This check is also part of the quality assurance concept. It also shows that cross-domain aspects play an important role. The model-based check offers opportunities to find and eliminate such errors early in the project.

6.1.1.3 Semantic checking

In Chapter 3, it was shown that the semantics of models play a central role in the BIM methodology. With the help of semantics, various information is stored, transported and made available. The inspection of properties, respectively their designation and their property values, is the most obvious check. On the basis of existing property definitions, also called object type lists, property lists, object and attribute lists or catalogs, semantic structures can be checked for consistency. More demanding, however, are checks for logical relationships between different properties. For this purpose, a database has been developed which represents building structures and thus the expected value for models to be checked. This database is also used to store the expected values of geometrical dimensions. In infrastructure, components are often standardized so that their properties are already known in advance. This makes it possible to conduct automated checks for so-called semantic-geometric coherence. This involves checking whether the dimensions of the three-dimensionally modeled component match the information specified in the properties. The challenge here is to build up an extensive database that holds the components relevant for the infrastructure, including the associated properties. In the work discussed here, this has only been developed as an example. Likewise, the limits and restrictions that arise in the current state of the art of infrastructure models were shown.

Another aspect relevant to the checking of the semantic model is the linking of different model dimensions. Since the linking logic of 3D to 4D or 5D models is mainly created manually, an error-prone process is assumed in the sense of a quality assurance strategy. The fact that such checks are possible was shown in Section 3.4.3.

The semantic model is also used for the automated clash detection of 3D and 4D models mentioned above.

Since the semantics of models is used in many ways and linking and evaluation processes are based on it, Hypothesis 2 is considered confirmed. The investigations are based on explicit geometry descriptions, except for the detached domain “Construction Design”. Thus, Objective 2 has also been achieved.

6.1.1.4 Code Compliance Checking

Probably the most demanding domain is the module of the quality assurance concept defined as “Construction Design”. This is understood to mean the checking of the 3D model for compliance with existing guidelines. In building design, there are some studies dedicated to this topic, but there are very few examples of infrastructure design.

The challenges in this topic are complex. The current situation is as follows:

There are numerous guidelines, standards, and laws for railway infrastructure. These are available in human-readable form as continuous text, tables, and graphics or similar. Numerous studies have examined the translation of human-readable texts into machine-readable formats and different methods can be used for this purpose, among them logic-based or language-based approaches, query languages, ontologies, natural language processing, text-based (hard-coded) or visual programming languages.

A goal of the investigations in this dissertation aim is to give the user insight into the developed test routines through the greatest possible transparency. Since it must be assumed that the users come from the construction design industry and typically have limited IT or programming knowledge, a visual programming language must be used to meet this requirement. For this purpose, the Business Process Model and Notation, standardized in ISO 19510, was used in combination with the Decision Model and Notation. Both are developed as executable languages with the help of workflow and decision engines, respectively.

The extent to which these modeling languages are suitable for representing guideline content was investigated. The investigations show that up to 68 % of the rules relevant for the railway model can be represented with the help of BPMN and DMN. The suitability of these two modeling languages is therefore rated as good. The examined rules of the guideline were classified and corresponding proofs of the feasibility of the proposed method were provided for the three most frequently occurring rule classes.

There are various software products that support the design of railway structures. These specialized applications mainly facilitate implicit geometry description as a parametric model in the so-called 3-sheet projection (also called 2.5D). Here, the data model is divided into cross-section, site plan and longitudinal section. Standardized data exchange formats for these data models exist (IDM^{VU}) but are rarely used in practice. The internationally recognized and standardized IFC format is currently still under development for elements of the infrastructure and the currently published part that concerns alignment can only be used for railway infrastructure with restrictions. While the transfer of explicitly described geometries is possible via generic classes of the IFC, it results in the loss of the parametric model information.

In the work presented in this dissertation, the proofs of the developed methods were therefore mainly carried out using native software formats where the parametric model information can be accessed. Where the IFC format is applicable, this was also used.

The methods developed achieve a reduction in testing times of up to 80 %, which represents an enormous increase in efficiency.

In summary, it can be stated that checking for code compliance can be achieved better using implicit geometry descriptions than with explicit geometries, since the parametric model information can be accessed. Hypothesis 3 is thus confirmed, and Objective 3 has been achieved.

6.1.2 Approach 2 – Model creation

Code Compliance Checking focuses on checking existing models for compliance with guidelines. This represents a downstream quality check.

Knowledge Based Engineering pursues the goal of automated model creation or the automation of routine tasks. In this dissertation this is understood as accompanying quality assurance, since the guideline knowledge must already be represented in the algorithms to produce a compliant result. In addition to knowledge about the content of guidelines, it is also necessary to identify which steps are performed by the modelers to design a product or building.

With the help of the MOKA method, a KBE application was developed for the field of infrastructure design. MOKA is a multi-stage system that aims to successively formalize initially unstructured and possibly inaccurately documented knowledge so that it can ultimately be applied automatically in a KBE application.

As in the earlier code compliance checking, the modeling languages BPMN and DMN were used to combine knowledge from different sources in such a way that routine tasks in the context of design of railway structures can be automated. Guidelines as well as results from interviews with experts in the respective disciplines were utilized.

The data exchange is also based on native software data.

The results of the investigations show that over 90% of the time required for manual design can be saved. This frees up resources for other tasks.

It could be demonstrated that the methods suitable for downstream model checking can also be used for the guideline-compliant creation of models. Hypothesis 4 is thus confirmed, and Objective 4 has been achieved.

6.2 Research contributions

This dissertation contributes to the standardized quality assurance of BIM models in the context of railway infrastructure.

No cross-domain quality assurance concept exists in currently available literature and so far only individual aspects have been examined. The developed concept is modular, so that necessary tests can be supplemented and those that are not necessary can be omitted.

In addition to the presented concept, a procedure for the evaluation of model quality was introduced. This makes it possible to carry out evaluations across models, projects, teams, and companies based on qualitative aspects. The standardized evaluation of model quality using several criteria is thus made possible, and subjective perceptions lose significance.

Automatisms were developed for the individual domains, the background of which is explained in this thesis. The methods are generic, so that they can be used not only for railway infrastructure but also for other infrastructure facilities, such as road construction or civil engineering. Some of the methods can also be used in the field of building design.

In particular, the work in this dissertation makes an important contribution to infrastructure design in the context of code compliance checking. The central questions here are: (1) How can guideline content be mapped in a machine-readable way? (2) What kind of models can be used for checking? The work discussed here provides corresponding statements and proves them qualitatively.

In addition to reviewing existing models, the work in this dissertation closes the gap in automated model generation using different knowledge sources. Examples for KBE applications exist in the automotive and aerospace sector, and partially also in the construction sector. For the design of railway structures this approach is new. This dissertation makes statements about the possible increase in efficiency when using such methods.

Following recent literature, a visual programming language was used for code compliance checking and knowledge-based engineering. This creates a level of transparency for the user that is not possible with text-based languages. It is assumed that transparency also contributes to acceptance among users, in turn promoting the use of digital methods in the construction planning industry. Both notations BPMN and DMN are standardized, software neutral, visual and therefore easy to understand, exchangeable and executable. BPMN is not only used in context of code compliance checking or Knowledge Based Engineering, but also for creating IDM. Furthermore it can be integrated in other processes between different stakeholders outside the BIM context. So people can use one language for different aspects. This is an advantage over other visual programming languages which are coupled to software products or not generic and thus not able to support different processes among a building design project.

6.3 Outlook

Digitalization will increasingly transform the construction planning industry in the coming years. Digital methods can lead to efficiency gains as well as to an increase in quality within construction projects.

This dissertation presents a comprehensive approach to ensuring quality in model-based projects. However, it cannot encompass all aspects and there is therefore further scope for extending the possibilities of the methods presented. These are discussed in the following sections.

6.3.1 Software and data exchange formats

Currently, there are still some limitations to the checking of railway models. On the one hand, this is due to the available model checking software, which mainly originate from and are designed for building design. On the other hand, necessary data exchange formats are not yet fully available, so that a lot of information is lost or cannot be evaluated in a standardized way.

The current lack of adequate data exchange formats looks set to be resolved in the short to medium term through current developments in the context of IFC as well as OKSTRA.

Future research could investigate which differences exist between building and infrastructure design with regard to model checking. On this basis, domain-specific extensions could then be developed and integrated into model checking software. Although the currently available software products usually offer programming interfaces, this offers no added value for users without programming knowledge.

6.3.2 Code Compliance Checking

The integration of BIM, BPMN and DMN is an essential aspect of the work in this dissertation. It could be proven that up to 68% of the rules relevant for modeling can be represented with BPMN and DMN, a result that can be classified as good.

Subsequent research projects could investigate whether other approaches could lead to a higher degree of implementation. Since BPMN and DMN were developed for a different purpose, it is also important to investigate whether domain-specific extensions could contribute to a better result.

Even though BPMN can be classified as a visual programming language, process development by users without programming knowledge is typically possible only to a limited extent. Future research could investigate the development of standardized node elements that can be used in individually developable processes without the user having to program the concrete contents.

In particular, the translation process from human-readable to machine-readable formats is currently a manual and frequently time-consuming process. Automated methods such as NLP can already be used in the process development phase to reduce manual effort.

The approach presented here provides table-based result reports for each inspection operation. A graphical connection to the three-dimensional model is currently not implemented. In line with the model-based approach, test results could in future also be visualized on the model.

Subsequent research activities could investigate the extent to which the presented method can be applied to other areas of infrastructure design, such as road or civil engineering design.

6.3.3 Knowledge Based Engineering

The methods presented here for automated model generation in connection with the planning of railway structures are only a first step in this area of research.

Since the approach used here is the same as that used for Code Compliance Checking using BPMN and DMN, the same development possibilities as outlined in Section 6.3.2 apply. These include the investigation of alternative approaches to overcoming limitations of the method, but also its extension to other disciplines such as road and civil engineering design.

Furthermore, it could be investigated whether the Case Management Model and Notation (CMMN) modeling language, which is related to BPMN and DMN, can be integrated. Standalone BPMN might be too restrictive and static in case different scenarios have to be considered or new knowledge, which is gained while designing, changes the requirements. As CMMN is developed to handle different cases it could be a valuable addition.

Knowledge-based engineering represents the interface between CAD and artificial intelligence. The development of KBE routines with the help of machine learning methods, i.e., by observing design processes or by evaluating parametric models, brings this a step closer.

In the research presented here, a simple form of inference engine was used. A higher-level inference engine would allow for the generation of new knowledge. Future research could pursue this approach.

Railway structures are subject to numerous dependencies across different disciplines. In the research presented here, only one primary discipline was discussed. The cross-domain integration of knowledge of various kinds should be studied in future research with the aim of facilitating efficient interaction between disciplines that are treated separately in practice. Further increases in efficiency can be expected by cross-disciplinary approaches in computer-aided applications.

7 References

[1] C. Eastman, P. Teichholz, R. Sacks, K. Liston, BIM Handbook - A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers, and Contractors, John Wiley & Sons, Inc., (2008). ISBN:978-0-470-18528-5.

[2] Bundesministerium für Verkehr und digitale Infrastruktur, Stufenplan Digitales Planen und Bauen, (2015). https://www.bmvi.de/SharedDocs/DE/Publikationen/DG/stufenplan-digitales-bauen.pdf?__blob=publicationFile.

[3] Bundesministerium für Verkehr und digitale Infrastruktur, Abschlussbericht der Reformkommission Großprojekte, (Final Report of the Reform Commission of Large-Scale Construction Projects). (2015) pp. 1–112. http://www.bmvi.de/SharedDocs/DE/Publikationen/G/reformkommission-bau-grossprojekte-endbericht.pdf?__blob=publicationFile (accessed June 20, 2019).

[4] C. Preidel, Automatisierte Konformitätsprüfung digitaler Bauwerksmodelle hinsichtlich geltender Normen und Richtlinien mit Hilfe einer visuellen Programmiersprache (Automated code compliance checking of digital building models with regard to applicable standards and, (2020) pp. 1–266. https://mediatum.ub.tum.de/doc/1534486/1534486.pdf (accessed October 20, 2020).

[5] S.M. İlal, H.M. Günaydın, Computer representation of building codes for automated compliance checking, Automation in Construction. 82 (2017) pp. 43–58. doi:10.1016/j.autcon.2017.06.018.

[6] Bundesministerium für Justiz und Verbraucherschutz, Allgemeines Eisenbahngesetz, (2021) p. https://www.gesetze-im-internet.de/aeg_1994/__4.html (accessed April 15, 2021).

[7] Bundesministerium für Verkehr und digitale Infrastruktur, Sicherheit im Eisenbahnbetrieb, (2021) p. https://www.bmvi.de/DE/Themen/Mobilitaet/Schiene/Sicherheit-Schienenverkehr/sicherheit-schienenverkehr.html (accessed April 15, 2021).

[8] DB Netz AG, Richtlinie 820 - Grundlagen des Oberbaues, (Guideline 820 - Basics of Superstructure). (2013) p. https://mediendienste.extranet.deutschebahn.com/DIBS/.

[9] Deutsche Bahn AG, BIM Methodik bei der DB Station & Service AG, (BIM Methodology at DB Station & Service AG). (2020) p. https://www1.deutschebahn.com/sus-infoplattform/start/Vorgaben-zur-Anwendung-der-BIM-Methodik (accessed March 22, 2020).

[10] DVV Media Group GmbH, BIM4RAIL, (2019) p. https://www.bahn-manager.de/bim4rail/.

[11] DB Netz AG, BIM@DB Netz, (2021) p. https://www1.deutschebahn.com/db-netz-bim (accessed April 13, 2021).

[12] China Railway BIM Alliance, Railway BIM Data Standard, (2015) pp. 1–233. https://www.buildingsmart.org/wp-content/uploads/2017/09/bSI-SPEC-Rail.pdf (accessed July 30, 2020).

[13] G. Ge, China Railway BIM and Standards, (2015) pp. 1–34. https://syncandshare.lrz.de/download/MktrQlVFTm1CTHBLS3lNcW5MQVdz/IFC Rail%2BRoad/2015-12-19/3a_IfcRail_CRBIM_Slides_2015-10-13.pdf?inline (accessed July 30, 2020).

[14] Bentley Systems Inc., BIM project for the Beijing-Zhangjiakou high-speed railway, n.d. p. https://www.bentley.com/en/project-profiles/2018/china-railway-engineering_beijing-zhangjiakou-railway (accessed April 12, 2021).

[15] W. Lee, S. Kang, R. Moh, R. Wu, H. Hsieh, Z. Shu, Application of BIM coordination technology to HSR Changhua station, Visualization in Engineering. 3 (2015) pp. 1–10. doi:10.1186/s40327-014-0008-9.

[16] G. Lee, Railway BIM 2030 roadmap of South Korea, (2018) pp. 1–31. http://big.yonsei.ac.kr/railbim/reports/RailBIM2030Roadmap_Full_Eng_Final.pdf (accessed July 30, 2020).

[17] G. Lee, Rail BIM 2030 Roadmap of South Korea (2), (2018) p. http://big.yonsei.ac.kr/railbim/ (accessed April 13, 2021).

[18] DIGITAL SNCF, La méthode BIM, ou comment bâtir dans l'ère numérique (The BIM method, or how to build in the digital age), (2017) p. https://www.digital.sncf.com/actualites/la-methode-bim-ou-comment-batir-dans-lere-numerique (accessed July 29, 2020).

[19] Bauen digital Schweiz, 3rd Suisse BIM Congress - BIM Roadmap, (2018) pp. 1–51. https://bauen-digital.ch/assets/Downloads/de/BIM-Roadmap.pdf (accessed July 30, 2020).

[20] ITALFERR - Ferrovie dello Stato, Innovating to design the future, (2019) pp. 1–248. http://www.italferr.it/content/dam/italferr/expertise/innovazione/Innovating to design the future.pdf (accessed July 29, 2020).

[21] RailwayPRO Communication Platform, Russian Railways creates BIM-Technology unit, n.d. p. https://www.railwaypro.com/wp/russian-railways-creates-bim-technology-unit/ (accessed April 22, 2021).

[22] Trafikverket, BIM – the work approach of the future, (2019) p. https://www.trafikverket.se/en/startpage/projects/Road-construction-projects/the-stockholm-bypass/bim--the-work-approach-of-the-future/ (accessed July 30, 2020).

[23] Banedanmark, Launching an agreement for a common Nordic approach to BIM, (2018) p. https://uk.bane.dk/en/Supplier/News-for-suppliers/BIM (accessed July 30, 2020).

[24] Assystem, BIM strategy, tests and commissioning for the Doha metro, (2021) p. https://www.assystem.com/en/projects/bim-strategy-tests-and-commissioning-doha-metro/ (accessed April 14, 2021).

[25] High Speed Two Ltd., Digital Engineering and BIM at HS2, (2021) p. https://www.hs2.org.uk/building-hs2/digital-engineering-and-bim/ (accessed April 14, 2021).

[26] RB Rail AS, Rail Baltica, (2021) p. https://www.railbaltica.org/rb-rail-as-bim-documentation/ (accessed April 12, 2021).

[27] V.J.L. Gan, C.L. Wong, K.T. Tse, J.C.P. Cheng, I.M.C. Lo, C.M. Chan, Parametric modelling and evolutionary optimization for cost-optimal and low-carbon design of high-rise reinforced concrete buildings, Advanced Engineering Informatics. 42 (2019) p. 100962. doi:10.1016/j.aei.2019.100962.

[28] S. Sharma, A. Sawhney, M. Arif, Parametric Modelling for Designing Offsite Construction, Procedia Engineering. 196 (2017) pp. 1114–1121. doi:10.1016/j.proeng.2017.08.069.

[29] H. Pottmann, M. Eigensatz, A. Vaxman, J. Wallner, Architectural geometry, Computers and Graphics (Pergamon). 47 (2015) pp. 145–164. doi:10.1016/j.cag.2014.11.002.

[30] J. Amann, Eine objektorientierte Sprache zur Einbettung von Interpretationssemantik in digitale Bauwerksmodelle, (2018) p. http://nbn-resolving.de/urn/resolver.pl?urn:nbn:de:bvb:91-diss-20181218-1453871-1-2 (accessed April 22, 2021).

[31] P. Galiläer, Trassierung von Straßenverkehrsanlagen mit der Finiten Elemente Methode (Dissertation), (2010) pp. 1–158. https://d-nb.info/1009591363/34 (accessed April 16, 2021).

[32] A. Borrmann, T.H. Kolbe, A. Donaubauer, H. Steuer, J.R. Jubierre, M. Flurl, Multi-Scale Geometric-Semantic Modeling of Shield Tunnels for GIS and BIM Applications, Computer-Aided Civil and Infrastructure Engineering. 30 (2014) pp. 263–281. doi:10.1111/mice.12090.

[33] S. Vilgertshofer, A. Borrmann, Using graph rewriting methods for the semi-automatic generation of parametric infrastructure models, Advanced Engineering Informatics. 33 (2017) pp. 502–515. doi:10.1016/j.aei.2017.07.003.

[34] Autodesk Inc., Autodesk Civil 3D, (2021) p. https://www.autodesk.eu/products/civil-3d/rail-design?plc=CIV3D&term=1-YEAR&support=ADVANCED&quantity=1 (accessed February 13, 2021).

[35] Bentley Systems Inc., Bentley OpenRail Designer, (2021) p. https://www.bentley.com/en/products/product-line/civil-design-software/openrail-designer (accessed February 13, 2021).

[36] IB&T Software GmbH, card_1, (2021) p. https://www.card-1.com/en/cross-section-modules/ (accessed February 13, 2021).

[37] AKG Software Consulting GmbH, Vestra Civil, (2021) p. https://www.akgsoftware.com/solutions/railway/ (accessed February 13, 2021).

[38] ProVI GmbH, ProVI 6.2, (2020) p. https://www.provi-cad.de (accessed September 26, 2020).

[39] CGS Labs, Ferrovia, (2021) p. https://cgs-labs.com/ferrovia/ (accessed February 28, 2021).

[40] M. Pasetto, A. Giordano, P. Borin, G. Giacomello, Integrated railway design using Infrastructure-Building Information Modeling. the case study of the port of Venice, Transportation Research

Procedia. 45 (2020) pp. 850–857. doi:10.1016/j.trpro.2020.02.084.

[41] Allplan Deutschland GmbH, Nemetschek Allplan, (2021) p. https://www.allplan.com/de/loesungen/infrastrukturbau-software/ (accessed April 15, 2021).

[42] Autodesk Inc., Autodesk Revit, n.d. p. https://www.autodesk.de/products/revit/overview (accessed January 6, 2019).

[43] Allplan Deutschland GmbH, Nemetschek Allplan Bridge, (2021) p. https://www.allplan.com/de/produkte/bridge/ (accessed April 15, 2021).

[44] Autodesk Inc., Autodesk Infraworks 360, (2021) p. https://www.autodesk.de/products/infraworks-360/advanced-capabilities/bridge-design (accessed April 14, 2021).

[45] M. Nöldgen, BIM im Brücken- und Ingenieurbau, Springer Fachmedien Wiesbaden, Wiesbaden, (2016). doi:10.1007/978-3-658-12875-3.

[46] Dassault Systems, CATiA, n.d. p. https://www.3ds.com/products-services/catia/disciplines/construction/ (accessed April 15, 2021).

[47] DIN Deutsches Institut für Normung e. V., ISO 9000:2015 / DIN EN ISO 9000:2015-11, Quality management systems – Fundamentals and vocabulary, (2015) pp. 1–104.

[48] H. Kulusjärvi, Common BIM Requirements - Quality assurance, Principles and Practice of Modern Radiotherapy Techniques in Breast Cancer. (2012) pp. 1–27. https://asiakas.kotisivukone.com/files/en.buildingsmart.kotisivukone.com/COBIM2012/cobim_6_quality_assurance_v1.pdf (accessed November 30, 2020).

[49] W. Solihin, C. Eastman, Classification of rules for automated BIM rule checking development, Automation in Construction. 53 (2015) pp. 69–82. doi:10.1016/j.autcon.2015.03.003.

[50] R. Müller, Verordnung über die Honorare für Architekten- und Ingenieurleistungen (Fee Structure for Architects and Engineers), (2013). ISBN:978-3-481-03137-4.

[51] Bundeskammer der Architekten und Ingenieurkonsulentenn, Honorar Information Architektur (Fee Information Architecture), (2010) pp. 1–191. https://www.arching.at/fileadmin/user_upload/redakteure/HIA_2010/hia2010_gesamt.pdf (accessed August 31, 2020).

[52] Swiss Society of Engineers and Architects, Ordnung für Leistungen und Honorare der Bauingenieurinnen und Bauingenieure (Regularity and Fee Structure for Civil Engineers), (2020) pp. 1–68. http://shop.sia.ch/normenwerk/ingenieur/sia 103/d/2020/D/Product (accessed August 31, 2020).

[53] Verband Beratender Ingenieure, BIM-Leitfaden für die Planerpraxis, n.d. p. https://www.vbi.de/shop/bim-leitfaden-fuer-die-planerpraxis-1/ (accessed December 5, 2020).

[54] American Institute of Architects, AIA Document E202 ™ – 2008, Building Information Modeling Protocol Exhibit. (2008) pp. 1–9.

[55] BIM Forum, Level of Development (LOD) Specification Part I & Commentary, (2020) p. 277. https://bimforum.org/lod/ (accessed December 7, 2020).

[56] F. Mini, Entwicklung eines LoD Konzepts für digitale Bauwerksmodelle von Brücken und dessen Implementierung, (2016) p.

[57] The British Standards Institution, Specification for information management for the capital / delivery phase of construction projects using building information modelling, PAS 1192-2:2013. (2013) p. 68.

[58] ZukunftBAU, BIM-Leitfaden für Deutschland, (Guideline BIM for Germany), (2014) pp. 1–109. https://www.bbsr.bund.de/BBSR/DE/FP/ZB/Auftragsforschung/3Rahmenbedingungen/2013/BIMLeitfaden/Endbericht.pdf;jsessionid=12F63EC6FEA13FAC9753B52378D265D6.live11292?__blob=publicationFile&v=2 (accessed February 3, 2020).

[59] U.S. Institute of Building Documentation, USIBD Level of Accuracy (LOA) Specification Guide, (2016) p. 30. https://usibd.org/product/level-of-accuracy-loa-specification-version-2-0/ (accessed December 7, 2020).

[60] R. Kaden, C. Clemen, R. Seuss, J. Blankenbach, R. Becker, A. Eichhorn, A. Donaubauer, T.H. Kolbe, U. Gruber, Leitfaden Geodäsie und BIM - Version 1.0, (2017). ISBN:9783000577949.

[61] J. Tulke, 18.4 - Model Checking, in: Building Information Modeling - Technology Foundation and Industry Practice, (2017) p. 322. ISBN:978-3-319-92862-3.

[62] A. Borrmann, M. König, C. Koch, J. Beetz, 2.2 - Solid modeling, in: Building Information Modeling - Technology Foundation and Industry Practice, (2017) pp. 1–597. ISBN:978-3-319-92862-3.

[63] J.D. Camba, M. Contero, P. Company, Parametric CAD modeling: An analysis of strategies for design reusability, CAD Computer Aided Design. 74 (2016) pp. 18–31. doi:10.1016/j.cad.2016.01.003.

[64] buildingSMART International, Industry Foundation Classes (IFC), (2020) p. https://www.buildingsmart.org/standards/bsi-standards/industry-foundation-classes/ (accessed December 8, 2020).

[65] buildingSMART International, IFC Specifications Database, (2020) p. https://technical.buildingsmart.org/standards/ifc/ifc-schema-specifications/ (accessed December 7, 2020).

[66] A. Braun, S. Tuttas, A. Borrmann, U. Stilla, A concept for automated construction progress monitoring using BIM-based geometric constraints and photogrammetric point clouds, Journal of Information Technology in Construction. 20 (2015) pp. 68–79. http://www.itcon.org/2015/5 (accessed December 7, 2020).

[67] IGM GmbH, Datenaustauschformate für BIM, (2020) p. https://www.igmilde.de/datenaustauschformate-fuer-bim/ (accessed December 6, 2020).

[68] G. Schaber, H. Felsinger, CPIxml Schnittstelle zur Interoperabilität zwischen SolidWorks und RIB iTWO, in: R.J. Scherer, S.-E. Schapke (Eds.), Informationssysteme Im Bauwesen 1, Springer Berlin Heidelberg, Berlin, Heidelberg, (2014) pp. 415–427. doi:10.1007/978-3-642-40883-0_17.

[69] Bundesanstalt für Straßenwesen, interactive instruments GmbH, Objektkatalog für das Straßen- und Verkehrswesen, (2020) p. www.okstra.de (accessed December 6, 2020).

[70] J. Tulke, Objektkatalog für das Straßen- und Verkehrswesen, Änderungsantrag A0139 vom 03.01.2018, (2018) p. http://www.okstra.de/ (accessed December 7, 2020).

[71] S. Wunsch, M. Lehnert, J. Krimmling, J. Easton, Datenformate, -modelle und -konzepte für den Eisenbahnbetrieb, EI - Der Eisenbahningenieur. (2016) pp. 8–14.

[72] VDV, Infrastructure Data Manegement (IDMVU) Interface Standard - Data Model and Data Exchange Format, VDV Recommendation. 456 (2014) pp. 1–196. https://www.vdv.de/schriften---mitteilungen.aspx?mode=detail&id=0a23d764-8322-4986-a7f8-3ad7d8f8b629 (accessed December 7, 2020).

[73] International Union Of Railways, Feasibility Report - UIC RailTopoModel and data exchange format, (2013) p. http://www.railtopomodel.org/files/download/RailTopoModel/270913_trafIT_FinalReportFeasibilityStudyRailTopoModel.pdf (accessed December 7, 2020).

[74] V. Augele, Comparative Analysis of Building Information Modelling (BIM) and RailTopoModel/railML in View of their Application to Operationally Relevant Railway Infrastructure, (2017) p. 44. https://www.railml.org/en/public-relations/scientific-papers.html?file=files/download/papers/230117_TUDresdenAugele_TermPaperComparisonRailTopoModelIFCrailML.pdf (accessed December 7, 2020).

[75] A. Hlubuček, Railtopomodel and Railml 3 in Overall Context, Acta Polytechnica CTU Proceedings. 11 (2017) pp. 1–6. doi:10.14311/app.2017.11.0016.

[76] RailML.org, RailTopoModel, (2020) p. http://www.railtopomodel.org/en/homepage.html (accessed December 5, 2020).

[77] A. Nash, D. Huerlimann, J. Schuette, V.P. Krauss, RailML - A standard data interface for railroad applications, Advances in Transport. 15 (2004) pp. 233–240. doi:10.2495/978-1-84564-500-7/01.

[78] C. Rahmig, railML ® Use Case Definition Schematic Track Plan, (2019) pp. 1–21. https://www.railml.org/en/download/schemes.html (accessed December 7, 2020).

[79] B. Reiner, C. Klaus, J. Buder, Neue Werkzeuge in der LST-Planung mit PlanPro, Der Eisenbahningenieur. (2015) pp. 51–54.

[80] C. Klaus, Vom Bleistift zur Maus : LST-Planungswerkzeuge heute und morgen, Der Eisenbahningenieur. (2015) pp. 40–46.

[81] J. Buder, U. Maschek, Fachtagung „Durchgängige Datenhaltung in der LST-Planung“, Der Eisenbahningenieur. (2017) pp. 10–15. www.eurailpress.de/archiv/lst-planung/.

[82] M. Lehnert, B. Jaekel, S. Wunsch, C. Klaus, Automatisierte semantische Validierung von bahnspezifischen Daten – Teil 1, Der Eisenbahningenieur. (2018) pp. 16–20.

[83] M. Lehnert, B. Jaekel, S. Wunsch, C. Klaus, Automatisierte semantische Validierung von bahnspezifischen Daten – Teil 2, Der Eisenbahningenieur. (2018) pp. 20–24.

[84] ISO 19650-2:2019, 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, (2019) p.

[85] A. Borrmann, M. König, C. Koch, J. Beetz, Building Information Modeling - Technology Foundation and Industry Practice, (2017). ISBN:978-3-319-92862-3.

[86] N.J. Fiedler, Modernization scenarios of the building permit procedure in consideration of new technological tools, (2015) p. https://repositum.tuwien.at/bitstream/20.500.12708/2830/2/Modernization scenarios of the building permit procedure in consideration of new technological tools.pdf (accessed November 29, 2020).

[87] C. Eastman, Jae-min Lee, Yeon-suk Jeong, Jin-kook Lee, Automatic rule-based checking of building designs, Automation in Construction. 18 (2009) pp. 1011–1033. doi:10.1016/j.autcon.2009.07.002.

[88] M. Schenke, Logikkalküle in der Informatik Wie wird Logik vom Rechner genutzt?, (2013). doi:10.1007/978-3-8348-2295-6.

[89] W.W.W. Consortium, Extensible Markup Language (XML), (2016) p. https://www.w3.org/XML/ (accessed December 6, 2020).

[90] E. Hjelseth, N. Nisbet, Capturing Normative Constraints by Use of the Semantic Mark-up Rase Methodology, CIB W78-W102 2011: International Conference. (2011) pp. 26–28. https://itc.scix.net/pdfs/w78-2011-Paper-45.pdf (accessed July 15, 2020).

[91] C. Sydora, E. Stroulia, Towards Rule-Based Model Checking of Building Information Models, 36th International Symposium on Automation and Robotics in Construction. (2019) pp. 1–7. https://pdfs.semanticscholar.org/c921/479a5341aac384b16ab3d2d5277f66acb2f8.pdf (accessed February 3, 2020).

[92] J. Busse, B. Humm, C. Lübbert, F. Moelter, A. Reibold, M. Rewald, V. Schlüter, B. Seiler, E. Tegtmeier, T. Zeh, Was bedeutet eigentlich Ontologie?, Informatik-Spektrum. 37 (2014) pp. 286–297. doi:10.1007/s00287-012-0619-2.

[93] M. Venugopal, C.M. Eastman, J. Teizer, An ontology-based analysis of the industry foundation class schema for building information model exchanges, Advanced Engineering Informatics. 29 (2015) pp. 940–957. doi:10.1016/j.aei.2015.09.006.

[94] S. Zhang, F. Boukamp, J. Teizer, Ontology-based semantic modeling of construction safety knowledge: Towards automated safety planning for job hazard analysis (JHA), Automation in Construction. 52 (2015) pp. 29–41. doi:10.1016/j.autcon.2015.02.005.

[95] Y.C. Lee, C.M. Eastman, W. Solihin, An ontology-based approach for developing data exchange requirements and model views of building information modeling, Advanced Engineering Informatics. 30 (2016) pp. 354–367. doi:10.1016/j.aei.2016.04.008.

[96] C. Preidel, A. Borrmann, Towards code compliance checking on the basis of a visual programming language, Journal of Information Technology in Construction. 21 (2016) pp. 402–421. http://www.itcon.org/2016/25 (accessed February 3, 2020).

[97] P.K.M. Chan, A New Methodology for the Development of Simulation Workflows: Moving Beyond MOKA, (2013) pp. 1–204. http://resolver.tudelft.nl/uuid:51041c5c-66c2-4d47-bf20-91459a538f7d (accessed October 15, 2020).

[98] C. Preidel, A. Borrmann, Automated Code Compliance Checking Based on a Visual Language and Building Information Modeling, Proceedings of the 32nd International Symposium of Automation and Robotics in Construction, 15-18 June. (2015) pp. 256–263. doi:10.13140/RG.2.1.1542.2805.

[99] C. Preidel, S. Daum, A. Borrmann, Data retrieval from building information models based on visual programming, Visualization in Engineering. 5 (2017) pp. 1–14. doi:10.1186/s40327-017-0055-0.

[100] P. Pauwels, S. Zhang, Semantic Rulechecking for Regulation Compliance Checking: An Overview of Strategies and Approaches, Proceedings of the 32nd CIB W78 Conference. (2015) pp. 619–628.

[101] B. Zhong, C. Gan, H. Luo, X. Xing, Ontology-based framework for building environmental monitoring and compliance checking under BIM environment, Building and Environment. 141 (2018) pp. 127–142. doi:10.1016/j.buildenv.2018.05.046.

[102] S. Daum, A. Borrmann, Checking Spatio-Semantic Consistency of Building Information Models By Means of a Query Language, Proceedings of the International Conference on Construction Applications of Virtual Reality. (2013) pp. 1–9. https://publications.cms.bgu.tum.de/2013_daum_convr.pdf (accessed February 15, 2019).

[103] J. Dimyadi, R. Amor, Automated Building Code Compliance Checking – Where is it at?, Proceedings of the 19th World Building Congress: Construction and Society. (2013) pp. 172–185. doi:10.13140/2.1.4920.4161.

[104] J.P. Martins, A. Monteiro, LicA: A BIM based automated code-checking application for water distribution systems, Automation in Construction. 29 (2013) pp. 12–23. doi:10.1016/j.autcon.2012.08.008.

[105] J. Dimyadi, C. Clifton, M. Spearpoint, R. Amor, Regulatory Knowledge Encoding Guidelines for

Automated Compliance Audit of Building Engineering Design, in: 2014 International Conference on Computing in Civil and Building Engineering, (2014) pp. 1–8. doi:10.1061/9780784413616.067.

[106] J. Dimyadi, R. Amor, Automating Conventional Compliance Audit Processes, J. Ríos et Al (Eds) Product Lifecycle Management and the Industry of the Future, Vol 517. Proceedings of the 14th IFIP WG 5.1 International Conference (PLM 2017), Seville, Spain, 10-12 July 2017. (2017) pp. 324–334. https://www.researchgate.net/profile/Johannes_Dimyadi/publication/321691957_Automating_Conventional_Compliance_Audit_Processes/links/5b263743aca272277fb5d260/Automating-Conventional-Compliance-Audit-Processes.pdf (accessed January 5, 2020).

[107] J. Dimyadi, C. Clifton, M. Spearpoint, R. Amor, Computerizing Regulatory Knowledge for Building Engineering Design, Journal of Computing in Civil Engineering. 30 (2016) pp. 1–13. doi:10.1061/(asce)cp.1943-5487.0000572.

[108] S. Daum, a Borrmann, Definition and Implementation of Temporal Operators for a 4D Query Language, Proc. of the ASCE International Workshop on …. (2013) pp. 1–8. doi:10.1061/9780784413029.059.

[109] S. Daum, A. Braun, BIM-basierte Bauprozessplanung: Automatisierte Ableitung von Precedence Relationships durch eine räumliche Anfragesprache, Proc. Des 26. Forum Bauinformatik. (2014) pp. 1–8. http://mediatum.ub.tum.de/node?id=1232112 (accessed April 15, 2021).

[110] buildingSMART International, BIM Collaboration Format, (2021) p. https://technical.buildingsmart.org/standards/bcf/ (accessed April 20, 2021).

[111] M. Stokes, Managing Engineering Knowledge; MOKA: Methodology for Knowledge Based Engineering Applications, Professional Engineering Publishing Limited, (2001). ISBN:1860582958.

[112] G. La Rocca, Knowledge based engineering: Between AI and CAD. Review of a language based technology to support engineering design, Advanced Engineering Informatics. 26 (2012) pp. 159–179. doi:10.1016/j.aei.2012.02.002.

[113] J. Abulawi, Ansatz zur Beherrschung der Komplexität von vernetzten 3D-CAD-Modellen, Dissertation. (2012) pp. 1–311. https://edoc.sub.uni-hamburg.de/hsu/volltexte/2012/2993/ (accessed April 22, 2021).

[114] D. Singer, A. Borrmann, A novel knowledge-based engineering approach for infrastructure design, Proceedings of the Fourth International Conference on Soft Computing Technology in Civil, Structural and Environmental Engineering", Civil-Comp Press, Stirlingshire, UK. 109 (2015) pp. 1–16. doi:10.4203/ccp.109.17.

[115] H. Adeli, K.V. Balasubramanyam, A Knowledge-Based System for Design of Bridge Trusses, 2 (1988) pp. 1–20. doi:10.1061/(ASCE)0887-3801(1988)2:1(1).

[116] A.P. Chassiakos, P. Vagiotas, D.D. Theodorakopoulos, A knowledge-based system for maintenance planning of highway concrete bridges, Advances in Engineering Software. 36 (2005) pp. 740–749. doi:10.1016/j.advengsoft.2005.03.020.

[117] D. Singer, M. Bügler, A. Borrmann, Knowledge based Bridge Engineering - Artificial Intelligence meets Building Information Modeling, 23rd International Workshop of the European Group for Intelligent Computing in Engineering, EG-ICE 2016. (2016) pp. 1–10. https://publications.cms.bgu.tum.de/2016_Singer_AIB.pdf (accessed October 12, 2020).

[118] J. Stjepandić, H. Liese, W.J.C. Verhagen, P. Bermell-Garcia, Knowledge-Based Engineering, in: Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges, (2015) pp. 255–286. doi:10.1007/978-3-319-13776-6.

[119] M.K. Jha, P. Schonfeld, A highway alignment optimization model using geographic information systems, Transportation Research Part A: Policy and Practice. 38 (2004) pp. 455–481. doi:10.1016/j.tra.2004.04.001.

[120] E. Kim, M.K. Jha, D.J. Lovell, P. Schonfeld, Intersection modeling for highway alignment optimization, Computer-Aided Civil and Infrastructure Engineering. 19 (2004) pp. 119–129. doi:10.1111/j.1467-8667.2004.00342.x.

[121] E. Kim, M.K. Jha, P. Schonfeld, H.S. Kim, Highway alignment optimization incorporating bridges and tunnels, Journal of Transportation Engineering. 133 (2007) pp. 71–81. doi:10.1061/(ASCE)0733-947X(2007)133:2(71).

[122] L. Zhao, Z. Liu, J. Mbachu, Highway alignment optimization: An integrated BIM and GIS approach, ISPRS International Journal of Geo-Information. 8 (2019) p. doi:10.3390/ijgi8040172.

[123] W. Li, H. Pu, P. Schonfeld, J. Yang, H. Zhang, L. Wang, J. Xiong, Mountain Railway Alignment Optimization with Bidirectional Distance Transform and Genetic Algorithm, Computer-Aided Civil and Infrastructure Engineering. 32 (2017) pp. 691–709. doi:10.1111/mice.12280.

[124] H. Pu, H. Zhang, W. Li, J. Xiong, J. Hu, J. Wang, Concurrent optimization of mountain railway alignment and station locations using a distance transform algorithm, Computers and Industrial Engineering. 127 (2019) pp. 1297–1314. doi:10.1016/j.cie.2018.01.004.

[125] H. Pu, T. Song, P. Schonfeld, W. Li, H. Zhang, J. Hu, X. Peng, J. Wang, Mountain railway alignment optimization using stepwise & hybrid particle swarm optimization incorporating genetic operators, Applied Soft Computing Journal. 78 (2019) pp. 41–57. doi:10.1016/j.asoc.2019.01.051.

[126] T. Song, H. Pu, P. Schonfeld, W. Li, H. Zhang, Y. Ren, J. Wang, J. Hu, X. Peng, Parallel Three-Dimensional Distance Transform for Railway Alignment Optimization Using OpenMP, Journal of Transportation Engineering, Part A: Systems. 146 (2020) p. 04020029. doi:10.1061/jtepbs.0000344.

[127] T. Song, H. Pu, P. Schonfeld, H. Zhang, W. Li, J. Hu, J. Wang, J. Wang, Bi-objective mountain

railway alignment optimization incorporating seismic risk assessment, Computer-Aided Civil and Infrastructure Engineering. 36 (2021) pp. 143–163. doi:10.1111/mice.12607.

[128] W.J.C. Verhagen, P. Bermell-Garcia, R.E.C. Van Dijk, R. Curran, A critical review of Knowledge-Based Engineering: An identification of research challenges, Advanced Engineering Informatics. 26 (2012) pp. 5–15. doi:10.1016/j.aei.2011.06.004.

[129] Lino, Lino Design Automation, (2021) p. https://www.lino.de/en/design-automation/ (accessed April 15, 2021).

[130] T.S. AB, Tacton Design Automation, (2021) p. https://www.tacton.com/loesungen/design-automation/?lang=de (accessed April 15, 2021).

[131] Autodesk Inc., Autodesk Design Automation, (2021) p. https://www.autodesk.com/solutions/design-automation (accessed April 15, 2021).

[132] P.J. McGoey, A Hitch-hikers guide to: knowledge based engineering inAerospace (& other industries), in: INCOSE (Ed.), INCOSE Enchantment Chapter, (2011).

[133] D. Bryde, M. Broquetas, J.M. Volm, The project benefits of Building Information Modelling (BIM), International Journal of Project Management. 31 (2013) pp. 971–980. doi:https://doi.org/10.1016/j.ijproman.2012.12.001.

[134] S. Azhar, Building Information Modeling (BIM): Trends, Benefits, Risks, and Challenges for the AEC Industry, Leadership and Management in Engineering. 11 (2011) pp. 241–252. doi:10.1061/(ASCE)LM.1943-5630.0000127.

[135] J. Li, L. Hou, X. Wang, J. Wang, J. Guo, S. Zhang, Y. Jiao, A Project-Based Quantification of BIM Benefits, International Journal of Advanced Robotic Systems. 11:123 (2014) pp. 1–13. doi:10.5772/58448.

[136] K. Barlish, K. Sullivan, How to measure the benefits of BIM - A case study approach, Automation in Construction. 24 (2012) pp. 149–159. doi:10.1016/j.autcon.2012.02.008.

[137] H. Berg, Experiences with BIM for infrastructure implementation - Measured value of today's level - possibilities with future levels of BIM, (2014) p. http://novarc.ee/wp-content/uploads/konverents/Heidi_Berg_Vianova_NOVARC_Tallinn_02.10.14.pdf (accessed January 2, 2019).

[138] Deutsche Bahn AG, BIM bei der DB AG, (BIM at DB AG). (2018) p. https://www.deutschebahn.com/de/bahnwelt/bauen_bahn/bim/BIM-1186016 (accessed January 1, 2019).

[139] 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, (2018) p. https://www.iso.org/standard/68080.html (accessed June 15, 2019).

[140] Autodesk Inc., Autodesk Navisworks, (2018) p. https://www.autodesk.de/products/navisworks/overview (accessed January 6, 2019).

[141] Solibri Inc., Solibri, (2018) p. https://www.solibri.com/de/ (accessed October 5, 2018).

[142] CEAPOINT GmbH, Desite MD Pro, (2018) p. https://www.ceapoint.com/desite-md-md-pro/ (accessed January 6, 2019).

[143] R. Navon, Automated project performance control of construction projects, Automation in Construction. 14 (2005) pp. 467–476. doi:10.1016/j.autcon.2004.09.006.

[144] F.K.T. Cheung, J. Rihan, J. Tah, D. Duce, E. Kurul, Early stage multi-level cost estimation for schematic BIM models, Automation in Construction. 27 (2012) pp. 67–77. doi:10.1016/j.autcon.2012.05.008.

[145] J.H. Garrett, M.E. Palmer, S. Demir, Delivering the Infrastructure for Digital Building Regulations, Journal of Computing in Civil Engineering. 28 (2014) pp. 167–169. doi:10.1061/(asce)cp.1943-5487.0000369.

[146] J.P. Martins, B. Carvalho, V.A. Almeida, Automated rule-checking – a tool for design development, International Association for Housing Science - 41st World Congress on Housing - Sustainability and Innovation for the Future. (2016) pp. 1–8. https://core.ac.uk/download/pdf/143403202.pdf. (accessed January 25, 2019).

[147] A. Charles, Automated rule checking for in-house BIM norms of building models, (2017) pp. 1–78. https://pure.tue.nl/ws/files/58778770/Ayyadurai_Charles_0923390.pdf (accessed February 17, 2019).

[148] V. Getuli, S.M. Ventura, P. Capone, A.L.C. Ciribini, BIM-based Code Checking for Construction Health and Safety, in: Creative Construction Conference 2017, CCC 2017, 19-22 June 2017, Primosten, Croatia, The Author(s), (2017) pp. 454–461. doi:10.1016/j.proeng.2017.07.224.

[149] Ming C. Lin, S. Gottschalk, Collision detection between geometric models: A survey, Proceedings of Institute of Mathematics & Its Applications Conference on Mathematics of Surfaces. (1998) pp. 1–20. http://users.soe.ucsc.edu/~pang/161/w06/notes/cms98.pdf (accessed September 9, 2019).

[150] D. Mainzer, G. Zachmann, Collision detection based on fuzzy scene subdivision, GPU Computing and Applications. (2015) pp. 135–150. doi:10.1007/978-981-287-134-3_9.

[151] J. Schauer, A. Nüchter, Collision detection between point clouds using an efficient k-d tree implementation, Advanced Engineering Informatics. 29 (2015) pp. 440–458. doi:10.1016/j.aei.2015.03.007.

[152] S. Staub-French, A. Khanzode, 3D and 4D modeling for design and construction coordination: issues and lessons learned, Journal of Information Technology in Construction. Vol. 12 (2007) pp. 381–407. doi:http://www.itcon.org/2007/26.

[153] M. Mawlana, F. Vahdatikhaki, A. Doriani, A. Hammad, Integrating 4D modeling and discrete event simulation for phasing evaluation of elevated urban highway reconstruction projects, Automation in Construction. 60 (2015) pp. 25–38. doi:10.1016/j.autcon.2015.09.005.

[154] F. Leite, Automated Approaches Towards BIM-Based Intelligent Decision Support in Design, Construction, and Facility Operations, Proceedings of the European Group for Intelligent Computing in Engineering International Workshop in Advanced Computing Strategies for Engineering. (2018) pp. 276–286. doi:10.1007/978-3-319-91638-5_15.

[155] S. Zhang, J. Lee, M. Venugopal, J. Teizer, C. Eastman, Integrating BIM and Safety: An Automated Rule-Based Checking System for Safety Planning and Simulation, International Council for Research and Innovation in Building and Construction - Proceedings of CIB W99 Conference. (2011) pp. 1–13. https://pdfs.semanticscholar.org/7562/8a5f770262413dd86380b6e4128a0beae761.pdf?_ga=2.259812106.1444231183.1560769556-336857077.1560769556 (accessed January 10, 2019).

[156] L. Chen, H. Luo, A BIM-based construction quality management model and its applications, Automation in Construction. 46 (2014) pp. 64–73. doi:10.1016/j.autcon.2014.05.009.

[157] P. Company, M. Contero, J. Otey, J.D. Camba, M.J. Agost, D. Pérez-López, Web-Based system for adaptable rubrics case study on CAD assessmend, Educational Technology and Society. 20 (2017) pp. 24–41. https://www.researchgate.net/publication/317169762_Web-Based_System_for_Adaptable_Rubrics_Case_Study_on_CAD_Assessment/download (accessed January 27, 2019).

[158] J. Johansson, M. Contero, P. Company, F. Elgh, Supporting connectivism in knowledge based engineering with graph theory, filtering techniques and model quality assurance, Advanced Engineering Informatics. 38 (2018) pp. 252–263. doi:10.1016/j.aei.2018.07.005.

[159] W. Solihin, C. Eastman, Y.C. Lee, Toward robust and quantifiable automated IFC quality validation, Advanced Engineering Informatics. 29 (2015) pp. 739–756. doi:10.1016/j.aei.2015.07.006.

[160] DB Netz AG, Richtlinie 836 - Erdbauwerke und sonstige geotechnische Bauwerke planen, bauen und instand halten, (Guideline 836 - Design, Built and Operate Earthworks and Other Geotechnical Constructions). (2018) p. https://mediendienste.extranet.deutschebahn.com/DIBS/.

[161] The BIM Center, The BIM Center, n.d. p. https://www.thebimcenter.com/2016/03/what-is-clash-detection-how-does-bim-help.html (accessed October 13, 2018).

[162] ProVI GmbH, ProVI 6.1, (2019) p. https://www.provi-cad.de/ (accessed March 15, 2019).

[163] OCCS Development Committee Secretariat, OmniClass, (2017) p. www.omniclass.org (accessed May 19, 2018).

[164] Construction Project Information Committee, UniClass, n.d. p. https://www.cpic.org.uk/uniclass/ (accessed October 5, 2018).

[165] buildingSMART, buildingSMART Data Dictionary, n.d. p. http://bsdd.buildingsmart.org/ (accessed October 5, 2018).

[166] L. Böger, W. Huhnt, S. Wernik, Standardized Names for Object Types and Attributes as Basis for Cooperation Between Planning and Building, Proceedings of the European Group for Intelligent Computing in Engineering International Workshop in Advanced Computing Strategies for Engineering. (2018) pp. 307–324. doi:10.1007/978-3-319-91638-5_17.

[167] R.J. Scherer, S.-E. Schapke, Informationssysteme im Bauwesen 1, (Information Systems in Building Industry Part 1), Springer Vieweg, (2014). ISBN:9783642408823.

[168] M. Fischer, J. Haymaker, K. Liston, Benefits of 3D and 4D Models for Facility Managers and AEC Service Providers, in: 4D CAD and Visualization in Construction, (2003) pp. 1–32. ISBN:90 5809 354 9.

[169] M. Fischer, J. Kunz, The Scope and Role of Information Technology in Construction, Doboku Gakkai Ronbunshu. (2004) pp. 1–31. doi:10.2208/jscej.2004.763_1.

[170] S. Fuchs, Erschließung domänenübergreifender Informationsräume mit Multimodellen, (Development of Cross-Domain Information Systems with Multidimensional Computer Models), (2015). ISBN:978-3-86780-451-6.

[171] A. Stadler, Kohärenz von Geometrie und Semantik in der Modellierung von 3D Stadtmodellen, (Coherence of Geometry and Semantics of 3D City Models), in: Entwicklerforum Geoinformationstechnik 2007, (2007) pp. 167–181. ISBN:978-9-8322-6403-1.

[172] DB Netz AG, Richtlinie 800.0110 - Linienführung, (Guideline 800.0110 - Alignment). (2015) p. https://mediendienste.extranet.deutschebahn.com/DIBS/.

[173] N. Sajadfar, Y. Ma, A hybrid cost estimation framework based on feature-oriented data mining approach, Advanced Engineering Informatics. 29 (2015) pp. 633–647. doi:10.1016/j.aei.2015.06.001.

[174] J. Choi, F. Leite, D.P. de Oliveira, BIM-based benchmarking system for healthcare projects: Feasibility study and functional requirements, Automation in Construction. 96 (2018) pp. 262–279. doi:10.1016/j.autcon.2018.09.015.

[175] Y.C. Lee, C.M. Eastman, W. Solihin, Logic for ensuring the data exchange integrity of building information models, Automation in Construction. 85 (2018) pp. 249–262. doi:10.1016/j.autcon.2017.08.010.

[176] M. Häußler, A. Borrmann, Model-based quality assurance in railway infrastructure planning, Automation in Construction. 109 (2020) pp. 1–15. doi:https://doi.org/10.1016/j.autcon.2019.102971.

[177] J. Dimyadi, R. Amor, BIM-based Compliance Audit Requirements for Building Consent Processing, European Conferences on Product and Process Modeling in the Building Industry. (2018) pp. 1–6.

https://www.researchgate.net/profile/Johannes_Dimyadi/publication/327384647_BIM-based_Compliance_Audit_Requirements_for_Building_Consent_Processing/links/5b99f2f392851c4ba81815c8/BIM-based-Compliance-Audit-Requirements-for-Building-Consent-Processing.pdf (accessed February 3, 2020).

[178] S.J. Fenves, J.H. Garrett, H. Kiliccote, K.H. Law, K.A. Reed, Computer Representations of Design Standards and Building Codes: U.S. Perspective, The International Journal of Construction Information Technology. 3 (1995) pp. 13–34. https://pdfs.semanticscholar.org/3b4e/d73499c6be2a8d07dfeacdc94bb381764c27.pdf (accessed February 3, 2020).

[179] L. Ding, R. Drogemuller, M. Rosenman, D. Marchant, Automating code checking for building designs - DesignCheck, Cooperative Research Centre (CRC) for Construction Innovation. (2006) pp. 1–16. doi:http://ro.uow.edu.au/engpapers/4842/.

[180] H. Kim, F. Grobler, Design Coordination in Building Information Modeling (BIM) Using Ontological Consistency Checking, in: Computing in Civil Engineering (2009), American Society of Civil Engineers, Reston, VA, (2009) pp. 410–420. doi:10.1061/41052(346)41.

[181] A. Yurchyshyna, A. Zarli, An ontology-based approach for formalisation and semantic organisation of conformance requirements in construction, Automation in Construction. 18 (2009) pp. 1084–1098. doi:10.1016/j.autcon.2009.07.008.

[182] J.K. Lee, Building environment rule and analysis (BERA) language and its application for evaluating building circulation and spatial program, (2011) pp. 1–217. https://smartech.gatech.edu/bitstream/handle/1853/39482/Lee_Jin-Kook_201105_PhD.pdf (accessed July 18, 2020).

[183] R. Navarro-Prieto, J.J. Cañas, Are visual programming languages better? The role of imagery in program comprehension, International Journal of Human Computer Studies. 54 (2001) pp. 799–829. doi:10.1006/ijhc.2000.0465.

[184] J.M. Rodriguez Corral, I. Ruiz-Rube, A. Civit Balcells, J.M. Mota-Macias, A. Morgado-Estevez, J.M. Dodero, A study on the suitability of visual languages for non-expert robot programmers, IEEE Access. 7 (2019) pp. 17535–17550. doi:10.1109/ACCESS.2019.2895913.

[185] N.W. Young Jr., S. a Jones, H.M. Bernstein, Interoperability in the Construction Industry, SmartMarket Report. (2007) pp. 1–36. https://vdcscorecard.stanford.edu/sites/g/files/sbiybj8856/f/mcgraw_hill_s_smartmarket_0.pdf (accessed December 10, 2019).

[186] W. Solihin, J. Dimyadi, Y. Lee, C. Eastman, R. Amor, The Critical Role of Accessible Data for Bim- Based Automated Rule Checking Systems, LC3 2017 - Proceedings of the Joint Conference on Computing in Construction (JC3), July 4-7, 2017, Heraklion, Greece. I (2017) pp. 53–60. doi:https://doi.org/10.24928/JC3-2017/0161.

[187] H. Kim, J.K. Lee, J. Shin, J. Choi, Visual language approach to representing KBimCode-based Korea building code sentences for automated rule checking, Journal of Computational Design and Engineering. 6 (2019) pp. 143–148. doi:10.1016/j.jcde.2018.08.002.

[188] T. Kasim, H. Li, Y. Rezgui, T. Beach, Automated Sustainability Compliance Checking Process: Proof of Concept, Proceedings of the 13th International Conference on Construction Applications of Virtual Reality (ConVR). (2013) pp. 11–21. https://itc.scix.net/pdfs/convr-2013-1.pdf (accessed July 15, 2020).

[189] N.M. El-Gohary, R. Zhang, A Machine Learning Approach for Compliance Checking-Specific Semantic Role Labeling of Building Code Sentences, in: T. Hartmann, I. Mutis (Eds.), Advances in Informatics and Computing in Civil and Construction Engineering, Springer Nature Switzerland AG, (2019) pp. 561–568. doi:https://doi.org/10.1007/978-3-030-00220-6.

[190] X. Xu, H. Cai, Semantic approach to compliance checking of underground utilities, Automation in Construction. 109 (2020) pp. 1–21. doi:10.1016/j.autcon.2019.103006.

[191] N. Bus, A. Roxin, G. Picinbono, M. Fahad, Towards French smart building code: Compliance Checking Based on Semantic Rules, Proceedings of the 6th Linked Data in Architecture and Construction Workshop Towards. (2018) pp. 6–15. http://ceur-ws.org/Vol-2159/01paper.pdf (accessed January 10, 2020).

[192] M. Fahad, N. Bus, Conformance checking of IFC models via semantic BIM reasoner, EG-ICE 2019 - Workshop on Intelligent Computing in Engineering. (2019) pp. 1–10. https://pdfs.semanticscholar.org/28ec/e3b8dc96379edb20d1b4ce19604a1f26b142.pdf (accessed January 10, 2020).

[193] J. Zhang, N.M. El-Gohary, Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking, Automation in Construction. 73 (2017) pp. 45–57. doi:10.1016/j.autcon.2016.08.027.

[194] J. Zhang, N.M. El-Gohary, Semantic-Based Logic Representation and Reasoning for Automated Regulatory Compliance Checking, Journal of Computing in Civil Engineering. (2016) pp. 1–42. doi:10.1061/(ASCE)CP.1943-5487.0000583.

[195] K. Kincelova, C. Boton, P. Blanchet, C. Dagenais, BIM-based code compliance checking for fire safety in timber buildings: A comparison of existing tools, CSCE Annual Conference, 12-15 June 2019, Laval, Greater Montreal. (2019) pp. 1–10. https://www.semanticscholar.org/paper/BIM-BASED-CODE-COMPLIANCE-CHECKING-FOR-FIRE-SAFETY-Kincelova-Boton/6c610b35fdd4746999bfba5710b51fa044dc2ceb (accessed January 11, 2020).

[196] P. Patlakas, A. Livingstone, R. Hairstans, G. Neighbour, Automatic code compliance with multi-dimensional data fitting in a BIM context, Advanced Engineering Informatics. 38 (2018) pp. 216–231. doi:10.1016/j.aei.2018.07.002.

[197] B.A. Myers, Taxonomies of visual programming and program visualization, Journal of Visual

Languages and Computing. 1 (1990) pp. 97–123. doi:10.1016/S1045-926X(05)80036-9.

[198] T.R.G. Green, M. Petre, Usability analysis of visual programming environments: A “cognitive dimensions” framework, Journal of Visual Languages and Computing. 7 (1996) pp. 131–174. doi:10.1006/jvlc.1996.0009.

[199] T. Catarci, G. Santucci, Are visual query languages easier to use than traditional ones?: an experimental proof, Proceedings of the HCI’95 Conference on People and Computers X. (1995) pp. 323–338. https://pdfs.semanticscholar.org/815c/4a0a2fc011c91e3351fa2ceef675a176cc8e.pdf (accessed October 29, 2019).

[200] C. Preidel, A. Borrmann, Refinement of the visual code checking language for an automated checking of building information models regarding applicable regulations, 2017 ASCE International Workshop on Computing in Civil Engineering, IWCCE 2017. (2017) pp. 157–165. doi:10.1061/9780784480823.

[201] C. Preidel, A. Borrmann, Integrating Relational Algebra into a Visual Code Checking Language for Information Retrieval from Building Information Models, in: 16th International Conference on Computing in Civil and Building Engineering, Osaka, Japan, (2015) pp. 1–8. doi:10.13140/RG.2.1.4618.5201.

[202] P. Ghannad, Y.C. Lee, J. Dimyadi, W. Solihin, Automated BIM data validation integrating open-standard schema with visual programming language, Advanced Engineering Informatics. 40 (2019) pp. 14–28. doi:10.1016/j.aei.2019.01.006.

[203] F. Ritter, C. Preidel, D. Singer, Visuelle Programmiersprachen im Bauwesen - Stand der Technik und aktuelle Entwicklungen, (Visual Programming Languages in Construction Domain - State of the Art and Current Evolution). (2015) pp. 1–9. https://publications.cms.bgu.tum.de/2015_Ritter_FBI.pdf (accessed October 29, 2019).

[204] T. Bloch, M. Katz, R. Yosef, R. Sacks, Automated model checking for topologically complex code requirements – security room case study, Proceedings of the 2019 European Conference on Computing in Construction. 1 (2019) pp. 48–55. doi:10.35490/ec3.2019.157.

[205] buildingSMART, IFC Rail, (2020) p. https://www.buildingsmart.org/standards/rooms/railway/ifc-rail-project/ (accessed July 19, 2020).

[206] C. Zhang, J. Beetz, M. Weise, Model View Checking: Automated validation for IFC building models, 1 (2014) pp. 1–15. doi:10.1201/b17396-24.

[207] K. Kluza, K. Honkisz, From SBVR to BPMN and DMN Models. Proposal of Translation from Rules to Process and Decision Models, in: L. Rutkowski, M. Korytkowski, R. Scherer, R. Tadeusiewicz, L. Zadeh, J. Zurada (Eds) Artificial Intelligence and Soft Computing. ICAISC 2016. Lecture Notes in Computer Science, Springer, Cham, (2016) pp. 453–462. doi:10.1007/978-3-319-39384-1_39.

[208] ISO/IEC 19510:2013, Information technology — Object Management Group Business Process Model and Notation, 2013 (2013) pp. 1–534. https://www.omg.org/spec/BPMN/ISO/19510/PDF (accessed January 20, 2020).

[209] J. Recker, M. Indulska, M. Rosemann, P. Green, How good is BPMN really? Insights from Theory and Practice, 14th European Conference on Information Systems. (2006) pp. 1–12. https://aisel.aisnet.org/ecis2006/135/?utm_source=aisel.aisnet.org%2Fecis2006%2F135&utm_medium=PDF&utm_campaign=PDFCoverPages (accessed December 30, 2019).

[210] L. Janssens, E. Bazhenova, J. De Smedt, J. Vanthienen, M. Denecker, Consistent integration of decision (DMN) and process (BPMN) models, in: S. Espana, M. Ivanovic, M. Savic (Eds.): Proceedings of the CAiSE'16 Forum at the 28th International Conference on Advanced Information Systems Engineering, Ljubljana, Slovenia, 13-17 June 2016, (2016) pp. 121–128. https://pdfs.semanticscholar.org/2e68/54e24050a62d58657aeaea97620e7caead3d.pdf (accessed October 10, 2019).

[211] S.V. Aram, C. Eastman, Introducing a new methodology to develop the Information Delivery Manual for AEC projects, Proceedings of the CIB W78 2010: 27th International Conference – Cairo, Egypt. (2010) pp. 1–10. https://itc.scix.net/pdfs/w78-2010-49.pdf (accessed January 5, 2020).

[212] E. Alreshidi, M. Mourshed, Y. Rezgui, Cloud-Based BIM Governance Platform Requirements and Specifications: Software Engineering Approach Using BPMN and UML, Journal of Computing in Civil Engineering. 30 (2016) pp. 1–41. https://ascelibrary.org/doi/full/10.1061/%28ASCE%29CP.1943-5487.0000539 (accessed January 8, 2020).

[213] ISO 29481-1:2016, Building information models — Information delivery manual — Part 1: Methodology and format, (2016) pp. 1–40.

[214] M. Weise, T. Liebich, J. Wix, Integrating use case definitions for IFC developments, (2008) pp. 1–9. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.467.9870&rep=rep1&type=pdf (accessed September 30, 2020).

[215] M. Obergriesser, A. Borrmann, Infrastructural BIM Standards - Development of an Information Delivery Manual for the geotechnical infrastructural design and analysis process, (2012) pp. 1–7. https://www.researchgate.net/publication/290353617_Infrastructural_BIM_standards_-_Development_of_an_Information_Delivery_Manual_for_the_geotechnical_infrastructural_design_and_analysis_process (accessed October 8, 2020).

[216] Y.H. Park, C.Y. Cho, G. Lee, Identifying a subset of BPMN for IDM development, Proceedings of the CIB W78-W102, International Conference, Sophia Antipolis, France, 26-28 October. (2011) pp. 1–6. https://www.semanticscholar.org/paper/IDENTIFYING-A-SUBSET-OF-BPMN-FOR-IDM-DEVELOPMENT-Lee/c7e6ec2a662007ac80601475d6917a776f27ae62 (accessed September 9, 2019).

[217] A. Borrmann, M. König, C. Koch, J. Beetz, 4.3 - Process modeling, in: Building Information Modeling - Technology Foundation and Industry Practice, (2017) pp. 1–597. ISBN:978-3-319-92862-3.

[218] A. Awad, M. Weske, Visualization of Compliance Violation in Business Process Models, in: International Conference on Business Process Management, (2009) pp. 182–193. doi:10.1007/978-3-642-12186-9.

[219] A. Ghose, G. Koliadis, Auditing Business Process Compliance, in: Proceedings of the International Conference on Service-Oriented Computing (ICSOC-2007), Lecture Notes in Computing Sicence, 4749, (2007) pp. 169–180. doi:10.1007/978-3-540-74974-5_14.

[220] A. Awad, G. Decker, M. Weske, Efficient Compliance Checking Using BPMN-Q and Efficient Compliance Checking Using BPMN-Q, in: M. Dumas, M. Reichert, MC. Shan (Eds) Business Process Management. BPM 2008. Lecture Notes in Computer Science, 5240, (2008) pp. 326–341. doi:https://doi.org/10.1007/978-3-540-85758-7_24.

[221] A. Awad, BPMN-Q : A Language to Query Business Processes, M. Reichert, S. Strecker & K. Turowski, Enterprise Modelling and Information Systems Architectures – Concepts and Applications. (2007) pp. 115–128. https://dl.gi.de/handle/20.500.12116/22195 (accessed November 11, 2019).

[222] J. Recker, Opportunities and constraints : The current struggle with BPMN, Business Process Management Journal. 16 (2010) pp. 181–201. doi:10.1108/14637151011018001.

[223] F. Kog, A. Dikbas, R.J. Scherer, Verification and validation approach of BPMN represented construction processes, Creative Construction Conference 2014 Verification. (2014) pp. 265–271. https://pdfs.semanticscholar.org/e065/e86f19d160702f3d4b38e651472753ce8b6e.pdf (accessed July 5, 2019).

[224] S. Zolfagharian, J. Irizarry, Current Trends in Construction Site Layout Planning, in: Construction Research Congress 2014, (2014) pp. 1723–1732. doi:10.1061/9780784413517.176.

[225] Object Management Group, Decision Model and Notation Version 1.3, (2019) pp. 1–261. https://www.omg.org/spec/DMN/1.3/PDF (accessed August 14, 2020).

[226] T. Biard, A. Le Mauff, M. Bigand, J.-P. Bourey, Separation of Decision Modeling from Business Process Modeling Using New “Decision Model and Notation” (DMN) for Automating Operational Decision-Making, in: L. Camarinha-Matos, F. Bénaben, W. Picard (Eds) Risks and Resilience of Collaborative Networks. PRO-VE 2015. IFIP Advances in Information and Communication Technology, (2015) pp. 489–496. doi:10.1007/978-3-319-24141-8_45.

[227] S.J. Fenves, E.H. Gaylord, S.K. Goel, Decision Table Formulation of the AISC Specification, (1969) pp. 1–178. https://www.researchgate.net/publication/39067052_Decision_Table_Formulation_of_The_1969_Aisc_Specification (accessed September 5, 2019).

[228] J. Huysmans, K. Dejaeger, C. Mues, J. Vanthienen, B. Baesens, An empirical evaluation of the comprehensibility of decision table, tree and rule based predictive models, Decision Support Systems. 51 (2011) pp. 141–154. doi:10.1016/j.dss.2010.12.003.

[229] G. Lee, R. Sacks, C.M. Eastman, Specifying parametric building object behavior (BOB) for a building information modeling system, Automation in Construction. 15 (2006) pp. 758–776. doi:10.1016/j.autcon.2005.09.009.

[230] A. Borrmann, J. Amann, T. Chipman, J. Hyvärinen, T. Liebich, S. Muhic, L. Mol, J. Plume, P. Scarponcini, IFC Infra Overall Architecture Project Documentation and Guidelines, BuildingSMART. (2017) pp. 1–53. https://www.buildingsmart.org/wp-content/uploads/2017/07/08_bSI_OverallArchitecure_Guidelines_final.pdf (accessed March 8, 2020).

[231] M. Reifenhäuser, R. Klebermass, R. Mautz, BIM IFC-Alignment: Fachliche Anforderungen Rail der DB, ÖBB und SBB, (Technical Requirements Rail of DB, ÖBB, SBB). (2018) pp. 1–12. https://mediendienste.extranet.deutschebahn.com/DIBS/.

[232] A. Borrmann, S. Muhic, J. Hyvärinen, T. Chipman, S. Jaud, C. Castaing, C. Dumoulin, T. Liebich, L. Mol, The IFC-Bridge project – Extending the IFC standard to enable high-quality exchange of bridge information models, Proceedings of the 2019 European Conference for Computing in Construction. 1 (2019) pp. 377–386. doi:10.35490/EC3.2019.193.

[233] S. Esser, K. Aicher, IfcBridge Model Generation using Visual Programming, Forum Bauinformatik 2019. (2019) pp. 1–8. https://publications.cms.bgu.tum.de/2019_Esser_FBI.pdf (accessed March 5, 2020).

[234] S. Markic, J. Schlenger, I. Bratoev, Tangible Alignment Design, 30. Forum Bauinformatik, Weimar, Germany. (2018) pp. 1–8. https://publications.cms.bgu.tum.de/2018_Markic_FBI.pdf (accessed October 14, 2019).

[235] T. Chipman, T. Liebich, M. Weise, mvdXML specification 1.1, Specification of a standardized format to define and exchange Model View Definitions with Exchange Requirements and Validation Rules. By Model Support Group (MSG) of buildingSMART, BuildingSMART Malaysia. 1 (2016) pp. 1–49. http://buildingsmart-tech.org/mvd/XML/1.1 (accessed March 12, 2020).

[236] buildingSMART International, The UML Model Report of IFC Road Project, (2020) pp. 1–124. https://www.buildingsmart.org/wp-content/uploads/2020/06/IR-CS-WP2-UML_Model_Report_Part-5_.pdf (accessed August 14, 2020).

[237] DIN Deutsches Institut für Normung e. V., DIN 820-2:2004-10 Presentation of documents (ISO/IEC-Directives – Part 2, modified) - withdrawn, (2004) pp. 1–139.

[238] DIN Deutsches Institut für Normung e. V., DIN 820-2:2020-03 Presentation of documents (ISO/IEC-Directives – Part 2:2018, modified), (2020) pp. 1–232.

[239] S. Malsane, J. Matthews, S. Lockley, P.E.D. Love, D. Greenwood, Development of an object

model for automated compliance checking, Automation in Construction. 49 (2015) pp. 51–58. doi:10.1016/j.autcon.2014.10.004.

[240] DB Netz AG, Richtlinie 800.0130 - Streckenquerschnitte auf Erdkörpern, (Guideline 800.0130 - Cross Sections at Earthwork). (2018) p. https://mediendienste.extranet.deutschebahn.com/DIBS/.

[241] Camunda Services GmbH, Camunda, (2018) p. https://camunda.com/ (accessed October 23, 2019).

[242] Apache Groovy project, Groovy, (2020) p. https://groovy-lang.org/ (accessed July 28, 2020).

[243] J. Dimyadi, R. Amor, Regulatory Knowledge Representation for Automated Compliance Audit of BIM-Based Models, (2013) pp. 9–12. https://www.researchgate.net/publication/257955455_REGULATORY_KNOWLEDGE_REPRESENTATION_FOR_AUTOMATED_COMPLIANCE_AUDIT_OF_BIM-BASED_MODELS (accessed July 25, 2020).

[244] M. Häußler, S. Esser, A. Borrmann, Code compliance checking of railway designs by integrating BIM, BPMN and DMN, Automation in Construction. 121 (2021) pp. 1–24. doi:10.1016/j.autcon.2020.103427.

[245] K. Amadori, M. Tarkian, J. Ölvander, P. Krus, Flexible and robust CAD models for design automation, Advanced Engineering Informatics. 26 (2012) pp. 180–195. doi:10.1016/j.aei.2012.01.004.

[246] R. Rempling, A. Mathern, D. Tarazona Ramos, S. Luis Fernández, Automatic structural design by a set-based parametric design method, Automation in Construction. 108 (2019) p. 102936. doi:10.1016/j.autcon.2019.102936.

[247] Q. Zou, H.Y. Feng, A decision-support method for information inconsistency resolution in direct modeling of CAD models, Advanced Engineering Informatics. 44 (2020) p. 101087. doi:10.1016/j.aei.2020.101087.

[248] S. Jaud, S. Esser, S. Muhič, A. Borrmann, Development of IFC schema for infrastructure, Proceedings of the 6th International Conference SiBIM. (2020) pp. 1–9. https://publications.cms.bgu.tum.de/2020_Jaud_siBIM.pdf (accessed February 14, 2021).

[249] Verein Deutscher Ingenieure e.V., Wissensmanagement im Ingenieurwesen - Wissensbasierte Konstruktion (Knowlegde management for engineering - Knowledge-based engineering), VDI 5610 (2). (2017) pp. 1–67. https://www.vdi.de/richtlinien/details/vdi-5610-blatt-2-wissensmanagement-im-ingenieurwesen-wissensbasierte-konstruktion-kbe (accessed October 12, 2020).

[250] ISO 30401:2018, Knowledge management systems - Requirements, 11 (2018) pp. 1–28.

[251] J. Rowley, The wisdom hierarchy: Representations of the DIKW hierarchy, Journal of Information Science. 33 (2007) pp. 163–180. doi:10.1177/0165551506070706.

[252] G. Bellinger, D. Castro, A. Mills, Data, Information, Knowledge, and Wisdom, (2004) pp. 1–3. https://www.systems-thinking.org/dikw/dikw.htm (accessed October 21, 2020).

[253] G. Schreiber, H. Akkermans, A. Anjewierden, R. de Hoog, N.R. Shadbolt, W. Van de Velde, B.J. Wielinga, Knowledge Engineering and Management: The CommonKADS Methodology, (2018). doi:10.7551/mitpress/4073.001.0001.

[254] Y. Rezgui, C.J. Hopfe, C. Vorakulpipat, Generations of knowledge management in the architecture, engineering and construction industry: An evolutionary perspective, Advanced Engineering Informatics. 24 (2010) pp. 219–228. doi:10.1016/j.aei.2009.12.001.

[255] V. Premkumar, S. Krishnamurty, J.C. Wileden, I.R. Grosse, A semantic knowledge management system for laminated composites, Advanced Engineering Informatics. 28 (2014) pp. 91–101. doi:10.1016/j.aei.2013.12.004.

[256] J. Girodon, D. Monticolo, E. Bonjour, M. Perrier, An organizational approach to designing an intelligent knowledge-based system: Application to the decision-making process in design projects, Advanced Engineering Informatics. 29 (2015) pp. 696–713. doi:10.1016/j.aei.2015.07.001.

[257] D. Roth, H. Binz, R. Watty, Generic structure of knowledge within the product development process, 11th International Design Conference, DESIGN 2010. (2010) pp. 1681–1690. https://www.designsociety.org/publication/29514/GENERIC+STRUCTURE+OF+KNOWLEDGE+WITHIN+THE+PRODUCT+DEVELOPMENT+PROCESS (accessed October 12, 2020).

[258] E. Bolisani, C. Bratianu, The Elusive Definition of Knowledge, in: E. Bolisani, C. Bratianu (Eds.), Emergent Knowledge Strategies: Strategic Thinking in Knowledge Management, Springer International Publishing, (2018) pp. 1–22. doi:10.1007/978-3-319-60657-6_1.

[259] A. Liew, DIKIW: Data, information, knowledge, intelligence, wisdom and their interrelationships, Business Management Dynamics. 2 (2013) pp. 49–62. https://www.researchgate.net/publication/236870996_DIKIW_Data_Information_Knowledge_Intelligence_Wisdom_and_their_Interrelationships (accessed October 21, 2020).

[260] H. Haller, User Interfaces for Personal Knowledge Management with Semantic Technologies, (2011) pp. 1–203. http://www.heikohaller.de/literatur (accessed August 31, 2020).

[261] M.Y. Cheng, H.S. Peng, C.M. Huang, C.H. Chen, KM-oriented business process reengineering for construction firms, Automation in Construction. 21 (2012) pp. 32–45. doi:10.1016/j.autcon.2011.05.010.

[262] J.W. Wu, J.C.R. Tseng, W. Der Yu, J. Bin Yang, S.M. Lee, W.N. Tsai, An integrated proactive knowledge management model for enhancing engineering services, Automation in Construction. 24 (2012) pp. 81–88. doi:10.1016/j.autcon.2012.02.006.

[263] D. Tang, R. Zhu, J. Tang, R. Xu, R. He, Product design knowledge management based on design structure matrix, Advanced Engineering Informatics. 24 (2010) pp. 159–166.

doi:10.1016/j.aei.2009.08.005.

[264] S. Bhaskara, DSM based approach for managing requirements, rules and design parameters in knowledge based design process, Managing Complexity by Modelling Dependencies - Proceedings of the 12th International DSM Conference. (2010) pp. 237–248. https://www.designsociety.org/publication/30370/DSM+Based+Approach+for+Managing+Requirements%2C+Rules+andDesign+Parameters+in+Knowledge+Based+Design+Process (accessed August 28, 2020).

[265] F. Liu, A.K. Jallow, C.J. Anumba, D. Wu, Building Knowledge Modeling: Integrating Knowledge in BIM, Cib W78. (2013) pp. 9–12. https://www.semanticscholar.org/paper/BUILDING-KNOWLEDGE-MODELING%3A-INTEGRATING-KNOWLEDGE-Liu-Jallow/26e436be170558ac75112afe1edd4ce2352bb5b0 (accessed October 11, 2020).

[266] A. Ertas, Transdisciplinary Engineering Design Process, John Wiley & Sons, Inc., (2018). ISBN:9781119474753.

[267] D.E. Calkins, N. Egging, C. Scholz, Knowledge-Based Engineering (KBE) Design Methodology at the Undergraduate and Graduate Levels, International Journal of Engineering Education. 16 (2000) pp. 21–38. https://www.ijee.ie/articles/Vol16-1/Ijee1118.pdf (accessed October 10, 2020).

[268] X. Zhao, W.J.C. Verhagen, R. Curran, Estimation of aircraft component production cost using knowledge based engineering techniques, Advanced Engineering Informatics. 29 (2015) pp. 616–632. doi:10.1016/j.aei.2015.05.004.

[269] P. Bermell-García, I.-S. Fan, A KBE System for the design of wind tunnel models using reusable knowledge components, Ponencia Presentada En El VI Congreso Internacional de Ingenieria de Proyectos Celebrada En Barcelona En El Ano 2002. (2002) pp. 594–604. https://www.semanticscholar.org/paper/A-KBE-System-for-the-design-of-wind-tunnel-models-García-Ip-Shing/17198e5244e421981afe16834cc2ea868a508a12 (accessed September 1, 2020).

[270] W. Skarka, Application of MOKA methodology in generative model creation using CATIA, Engineering Applications of Artificial Intelligence. 20 (2007) pp. 677–690. doi:10.1016/j.engappai.2006.11.019.

[271] P. Bermell-Garcia, W.J.C. Verhagen, S. Astwood, K. Krishnamurthy, J.L. Johnson, D. Ruiz, G. Scott, R. Curran, A framework for management of Knowledge-Based Engineering applications as software services: Enabling personalization and codification, Advanced Engineering Informatics. 26 (2012) pp. 219–230. doi:10.1016/j.aei.2012.01.006.

[272] A. Corallo, R. Laubacher, A. Margherita, G. Turrisi, Enhancing product development through knowledge-based engineering (KBE)-A case study in the aerospace industry, Journal of Manufacturing Technology Management. 20 (2009) pp. 1070–1083. doi:10.1108/17410380910997218.

[273] D. Steenhuizen, M. Van Tooren, The implementation of a knowledge-based framework for the aerodynamic optimization of a morphing wing device, Advanced Engineering Informatics. 26 (2012) pp. 207–218. doi:10.1016/j.aei.2012.02.004.

[274] T.G. Chiciudean, G. La Rocca, M.J.L. van Tooren, A Knowledge Based Engineering Approach to Support Automatic Design of Wind Turbine Blades, CIRP Design Conference 2008. (2008) pp. 1–5. http://resolver.tudelft.nl/uuid:de72e000-1c1e-4dbe-83ec-a1aef880d60e (accessed October 12, 2020).

[275] A.E. Yildiz, I. Dikmen, M.T. Birgonul, K. Ercoskun, S. Alten, A knowledge-based risk mapping tool for cost estimation of international construction projects, Automation in Construction. 43 (2014) pp. 144–155. doi:10.1016/j.autcon.2014.03.010.

[276] N. Singhaputtangkul, S.P. Low, A.L. Teo, B.G. Hwang, Knowledge-based decision support system quality function deployment (KBDSS-QFD) tool for assessment of building envelopes, Automation in Construction. 35 (2013) pp. 314–328. doi:10.1016/j.autcon.2013.05.017.

[277] A. Motamedi, A. Hammad, Y. Asen, Knowledge-assisted BIM-based visual analytics for failure root cause detection in facilities management, Automation in Construction. 43 (2014) pp. 73–83. doi:10.1016/j.autcon.2014.03.012.

[278] I. Motawa, A. Almarshad, A knowledge-based BIM system for building maintenance, Automation in Construction. 29 (2013) pp. 173–182. doi:10.1016/j.autcon.2012.09.008.

[279] M.J. Dibley, H. Li, J.C. Miles, Y. Rezgui, Towards intelligent agent based software for building related decision support, Advanced Engineering Informatics. 25 (2011) pp. 311–329. doi:10.1016/j.aei.2010.11.002.

[280] T.W. Kim, R. Rajagopal, M. Fischer, C. Kam, A knowledge-based framework for automated space-use analysis, Automation in Construction. 32 (2013) pp. 165–176. doi:10.1016/j.autcon.2012.08.002.

[281] L. Wang, F. Leite, Formalized knowledge representation for spatial conflict coordination of mechanical, electrical and plumbing (MEP) systems in new building projects, Automation in Construction. 64 (2016) pp. 20–26. doi:10.1016/j.autcon.2015.12.020.

[282] M. König, J. Dirnbek, V. Stankovski, Architecture of an open knowledge base for sustainable buildings based on Linked Data technologies, Automation in Construction. 35 (2013) pp. 542–550. doi:10.1016/j.autcon.2013.07.002.

[283] A. Cavieres, R. Gentry, T. Al-Haddad, Knowledge-based parametric tools for concrete masonry walls: Conceptual design and preliminary structural analysis, Automation in Construction. 20 (2011) pp. 716–728. doi:10.1016/j.autcon.2011.01.003.

[284] D. Singer, Entwicklung eines Prototyps für den Einsatz von Knowledge-based Engineering in frühen Phasen des Brückenentwurfs (Development of a KBE-prototype for early bridge-design stages), Master’s Thesis, TU München. (2014) pp. 1–121.

https://publications.cms.bgu.tum.de/theses/singer_2014_amann.pdf (accessed October 10, 2020).

[285] M. Sandberg, P. Jensen, Z. Ramic, P. Simonsson, Knowledge-based bridge design, IABSE Congress Stockholm, 2016: Challenges in Design and Construction of an Innovative and Sustainable Built Environment. (2016) pp. 1–8. https://www.diva-portal.org/smash/get/diva2:1000812/FULLTEXT01.pdf (accessed October 5, 2020).

[286] M. Flurl, D. Singer, R.-P. Mundani, E. Rank, A. Borrmann, A Rule-based Collaborative Modelling System for Infrastructure Design, (2015) pp. 1–10. https://publications.cms.bgu.tum.de/2015_Flurl_Collaboration.pdf (accessed October 12, 2020).

[287] Y. Chen, G.E. Okudan, D.R. Riley, Decision support for construction method selection in concrete buildings: Prefabrication adoption and optimization, Automation in Construction. 19 (2010) pp. 665–675. doi:10.1016/j.autcon.2010.02.011.

[288] S. An, P. Martinez, M. Al-Hussein, R. Ahmad, BIM-based decision support system for automated manufacturability check of wood frame assemblies, Automation in Construction. 111 (2020) pp. 1–15. doi:10.1016/j.autcon.2019.103065.

[289] M. Egemen, A. Mohamed, SCBMD: A knowledge-based system software for strategically correct bid/no bid and mark-up size decisions, Automation in Construction. 17 (2008) pp. 864–872. doi:10.1016/j.autcon.2008.02.013.

[290] M. Wang, Y. Deng, J. Won, J.C.P. Cheng, An integrated underground utility management and decision support based on BIM and GIS, Automation in Construction. 107 (2019) pp. 1–22. doi:10.1016/j.autcon.2019.102931.

[291] P. Sainter, K. Oldham, A. Larkin, A. Murton, R. Brimble, Product Knowledge Management Within Knowledge-Based Engineering Systems, Proceedings of DETC’00 ASME 2000 Design Engineering Technical Conference And Computers and Information in Engineering Conference. (2000) pp. 1–8.

[292] P. Sainter, K. Oldham, A. Larkin, Achieving benefits from knowledge-based engineering systems in the longer term as well as in the short term, Proceedings of the 6th International Conference on Concurrent Enterprising. (2000) pp. 143–150. https://www.researchgate.net/publication/229044325_Achieving_benefits_from_knowledge-based_engineering_systems_in_the_longer_term_as_well_as_in_the_short_term (accessed October 1, 2020).

[293] K.P. Tripathi, A Review on Knowledge-based Expert System: Concept and Architecture, Artificial Intelligence Techniques - Novel Approaches & Practical Applications. 4 (2011) pp. 19–23. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.206.3352&rep=rep1&type=pdf (accessed September 15, 2020).

[294] C.L. Emberey, N.R. Milton, J.P.T.J. Berends, M.J.L. Van Tooren, S.W.G. Van Der Elst, B.

Vermeulen, Application of Knowledge Engineering Methodologies to Support Engineering Design Application Development in Aerospace, 7th AIAA Aviation Technology, Integration, and Operations Conference. 1 (2007) pp. 83–95. doi:10.2514/6.2007-7708.

[295] R. Hunter, J. Rios, J.M. Perez, A. Vizan, A functional approach for the formalization of the fixture design process, International Journal of Machine Tools and Manufacture. 46 (2006) pp. 683–697. doi:10.1016/j.ijmachtools.2005.04.018.

[296] J. Montali, M. Overend, P.M. Pelken, M. Sauchelli, Knowledge-based engineering applications for supporting the design of precast concrete facade panels, in: Proceedings of the International Conference on Engineering Design, ICED, (2017) pp. 131–139. ISBN:978-1-904670-94-0.

[297] J. Montali, M. Sauchelli, Q. Jin, M. Overend, Knowledge-rich optimisation of prefabricated façades to support conceptual design, Automation in Construction. 97 (2019) pp. 192–204. doi:10.1016/j.autcon.2018.11.002.

[298] J. Barreiro, S. Martinez, E. Cuesta, B. Alvarez, P. Fernandez, High level diagrams for identification of knowledge as a basis for a KBE implementation of inspection planning process, III Manufacturing Engineering Society International Congress. (2009) pp. 272–278. https://www.researchgate.net/profile/J_Barreiro2/publication/235792948_High_level_diagrams_for_identification_of_knowledge_as_a_basis_for_a_KBE_implementation_of_inspection_planning_process/links/0912f5138df4477bb5000000/High-level-diagrams-for-identifica (accessed September 20, 2020).

[299] S. Martínez, J. Barreiro, E. Cuesta, B.J. Álvarez, D. González, Methodology for identifying and representing knowledge in the scope of CMM inspection resource selection, AIP Conference Proceedings. 1431 (2012) pp. 250–257. doi:10.1063/1.4707572.

[300] B.J. Álvarez, E. Cuesta, S. Martínez, J. Barreiro, P. Fernández, Implementation of decision rules for CMM sampling in a KBE system, Proceedings of the 36th International MATADOR Conference. 2010-Janua (2010) pp. 335–338. doi:https://doi.org/10.1007/978-1-84996-432-6_76.

[301] S. Ammar-Khodja, N. Perry, A. Bernard, Processing Knowledge to Support Knowledge-based Engineering Systems Specification, Concurrent Engineering Research and Applications. 16 (2008) pp. 89–101. doi:10.1177/1063293X07084642.

[302] M. Helgoson, V. Kalhori, A Conceptual Model for Knowledge Integration in Process Planning, Procedia CIRP. 3 (2012) pp. 573–578. doi:10.1016/j.procir.2012.07.098.

[303] W. Skarka, Constructing model of knowledge base according to MOKA methodology, Diagnostyka. 2 (2006) pp. 37–40. http://yadda.icm.edu.pl/baztech/element/bwmeta1.element.baztech-article-BAR0-0019-0043 (accessed October 12, 2020).

[304] P. Zheng, G. Zhao, V.H. Torres, J. Rios, A knowledge-based approach to integrate Fuzzy

Conceptual Design Tools and MOKA into a CAD system, Proceedings - 2012 7th International Conference on Computing and Convergence Technology (ICCIT, ICEI and ICACT), ICCCT 2012. (2012) pp. 1285–1291. https://www.researchgate.net/publication/259828884_A_Knowledge-based_Approach_to_Integrate_Fuzzy_Conceptual_Design_Tools_and_MOKA_into_a_CAD_System (accessed August 19, 2020).

[305] M. Lohith, L. Prasanna, D.H. Vaderahobli, Translating MOKA Based Knowledge Models into a Generative CAD Model in CATIA V5 Using Knowledgeware, International Conference on Modeling, Simulation and Visualization Methods(MSV). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp). (2013) pp. 1–6. https://www.semanticscholar.org/paper/Translating-MOKA-based-Knowledge-models-into-a-CAD-KnowledgewareLohith-Prasanna/58f5d9ba8829603a0e817d4e83198cf1a14ab2b1 (accessed August 18, 2020).

[306] M. Sandberg, Towards a Knowledge-Based Engineering Methodology for Construction, ICCREM 2015 - Environment and the Sustainable Building - Proceedings of the 2015 International Conference on Construction and Real Estate Management. (2015) pp. 1–8. doi:10.1061/9780784479377.001.

[307] UML Activity Diagrams, UML Activity Diagrams, (2020) p. https://www.uml-diagrams.org/activity-diagrams.html (accessed September 16, 2020).

[308] F. Tang, T. Ma, Y. Guan, Z. Zhang, Parametric modeling and structure verification of asphalt pavement based on BIM-ABAQUS, Automation in Construction. 111 (2020) p. 103066. doi:10.1016/j.autcon.2019.103066.

[309] N.C. Brown, V. Jusiega, C.T. Mueller, Implementing data-driven parametric building design with a flexible toolbox, Automation in Construction. 118 (2020) p. 103252. doi:10.1016/j.autcon.2020.103252.

[310] C. Van Der Velden, C. Bil, X. Xu, Adaptable methodology for automation application development, Advanced Engineering Informatics. 26 (2012) pp. 231–250. doi:10.1016/j.aei.2012.02.007.

[311] R. Brimble, F. Sellini, The MOKA Modelling Language, in: Knowledge Engineering and Knowledge Management Methods, Models, and Tools, (2000) pp. 49–56. ISBN:3540411194.

[312] M. Sandberg, Knowledge Based Engineering - In Product Development, (2003) pp. 1–16. https://www.diva-portal.org/smash/get/diva2:995005/FULLTEXT01.pdf (accessed July 16, 2020).

[313] M. Beier, H. Grütz, K. Jäger, G. Kock, J. Onnich, R. Strube, Headed for the Low-Noise Railway: The DB Noise Reduction Research Programme, n.d. p. http://www.railway-research.org/IMG/pdf/022.pdf (accessed September 26, 2020).

[314] Deutsche Bahn AG, Noise Barriers - Deutsche Bahn, (2020) p.

https://gruen.deutschebahn.com/en/measures/noise-barriers (accessed September 26, 2020).

[315] S. Markic, J. Schlenger, I. Bratoev, Tangible Alignment Design, 30. Forum Bauinformatik, Weimar, Germany. (2018) p. https://publications.cms.bgu.tum.de/2018_Markic_FBI.pdf (accessed March 12, 2020).