SCHOOL OF ENGINEERING AND DESIGN

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master’s Thesis

Assessing parking facility in the city using CityGML 3.0

Hongyu Chu

SCHOOL OF ENGINEERING AND DESIGN

TECHNISCHE UNIVERSITÄT MÜNCHEN

Master's Thesis

Assessing parking facility in the city using CityGML 3.0

Author:	Hongyu Chu
Supervisor:	Univ.-Prof. Dr. rer. nat. Thomas H. Kolbe
Advisors:	Dr. Ihab Hijazi, Christof Beil M.Sc.
Submission Date:	22.07.2024

I confirm that this master's thesis is my own work and I have documented all sources and material used.

Munich, 22.07.2024 Hongyu Chu

Abstract

Parking facilities play a crucial role in urban planning and traffic management, impacting the landscape and residents' quality of life. Regulatory assessments are necessary to ensure compliance with these facilities. In the past, regulatory assessments of parking facilities have focused on the initial phase of their lifecycle, such as building permit approvals, with less attention given to assessing operational parking facilities. Furthermore, most assessments are still conducted manually. With the ongoing digitisation across various fields, automated regulatory assessments have become imperative. Automation not only enhances efficiency and accuracy but also reduces long-term management costs, contributing to more sustainable urban development.

This study develops a checking workflow based on CityGML 3.0 to achieve automated regulatory assessments of parking facilities at the city scale. A case study method is employed, with the city of Ingolstadt selected as the study area due to the data availability. This study mainly consists of three parts.

The first part of the work involves determining the requirements for this case study, including both legal and technical aspects. As the basis for regulatory assessment, relevant regulations are determined. In this study, articles with quantitative properties are selected. Using a logic-based approach, the entities and their attributes in the selected articles are extracted, interpreted and formalised. A regulatory conceptual model is established in the form of a UML diagram. Then, the technical requirements of this use case are identified, and the capabilities of the CityGML 3.0 are evaluated to ensure it can effectively support the regulatory assessment process.

The regulatory conceptual model is mapped to the CityGML Conceptual Model, with a primary focus on the Transportation Module. For attributes that are missing in CityGML 3.0, an extension mechanism is employed to store these properties, i.e., using the Generic attributes. This mapping and extension process constitutes the second part of the work.

In the implementation phase, the checking workflows against the selected articles are developed in FME Workbench. The source datasets contain two parts: the building datasets in CityGML 1.0 and the transportation datasets exchanged in CityGML 3.0, containing detailed 3D information about street spaces. These source datasets are converted into extended CityGML 3.0 data models. The checkable objects and attributes are extracted and stored in these data models. The checking workflows are developed to comply with the regulations. Key parameters, such as the length and width of parking slots, the width of footpaths and driveways, and the number of parking slots are identified and inspected. The assessment results are presented in an HTML report with a 2D map, stored in the extended data model, and visualised in the 3D scene using ArcGIS Pro. The conceptual framework and checking process in this study demonstrate that automated regulatory assessment of parking facilities is feasible using CityGML 3.0.

Acknowledgements

I would like to express my sincere gratitude to my advisors, Dr. Ihab Hijazi and Christof Beil, M.Sc., for their continuous support and expert guidance throughout my thesis. I appreciate Ihab for giving me the opportunity to undertake this thesis and for his many insights into its direction and content. His patience and support during challenging times gave me the courage to persist.

Christof helped me understand the concepts and the relationships between the classes within the transportation module. We discussed in detail how to integrate regulations into CityGML 3.0 during many meetings. His technical assistance was invaluable. I also want to thank Roland Dietrich for always helping me with remote desktop issues.

I am also grateful to my friends, who listened to my worries and encouraged me during moments of self-doubt. Lastly, I owe a special thanks to my family for their unwavering support and unconditional love. Their encouragement has been a solid foundation for me. As a Chinese saying goes, "The journey ahead is long and arduous, but walking it will bring us to our destination."

Table of Contents

List of Figures

List of Tables

Abbreviations

ADE	Application Domain Extension
AEC	Architecture, engineering, and construction
CityGML	City Geography Markup Language
FME Workbench	Feature Manipulation Engine Workbench
GIS	Geographic Information System
LoD	Level of Detail
OGC	Open Geospatial Consortium
UML	Unified Modeling Language
XML	eXtensible Markup Language
XSD	XML Schema Definition

1. Introduction

1.1 Motivation

Traffic areas beyond residential zones predominantly shape urban landscapes. Parking facilities are a critical component of urban traffic and land use infrastructure, potentially encouraging car ownership and usage (Kirschner and Lanzendorf, 2020). Regulations impose various restrictions and requirements on parking facilities to address daily needs. In practice, research on parking facilities often centres on accessibility, traffic management, and parking policies (Christiansen et al., 2017; Marsden, 2006). For effective parking facility management, assessing their compliance with regulatory requirements is also an essential aspect of urban planning and management. Currently, compliance checks for parking facilities face two primary issues: insufficient regular re-inspection and maintenance of existing facilities and dependence on manual expert reviews, resulting in a lack of automation in the process.

Regulation compliance checks are primarily concentrated on the building permit check phase. However, the regulation checks extend throughout the design, construction, and operation phases of the built environment (Amor and Dimyadi, 2021). Post-construction facilities require ongoing maintenance and operation to ensure they fulfil their intended purpose, adhere to design specifications, function efficiently and economically, are used safely, and maintain environmental sustainability throughout their lifecycle. This aspect is particularly underdeveloped in the context of parking facilities (Klementschitz et al., 2007).

Manual reviews are error-prone and inefficiency, contributing to declining productivity within the domain (Dimyadi and Amor, 2013). With the advancement of digitisation, exploring automation in urban planning and regulatory checks is ongoing. One goal is to simplify and translate regulations into machine-readable formats to standardise interpretations and eliminate unnecessary ambiguities (Eastman et al., 2009). Machine-readable data is structured data that computers can automatically read and interpret (Open Knowledge Foundation). Currently, the storage and interpretation of laws and regulations mainly rely on human language. This variability in standards and interpretations across regions poses significant challenges for automating regulatory checks (Eastman et al., 2009). Thus, accurately digitising regulations emerges as a significant focus of this study.

Nowadays, the mainstream exploration of automated regulatory checks is active in building regulations using Building Information Modelling (BIM), leaving some gaps in the regulation checks using 3D city models. BIM is typically applied using the Industry Foundation Classes (IFC) standard at the individual building level. As semantic 3D city models continue to evolve, recent studies have shown interest in embedding regulations within 3D city models, such as urban planning and 3D cadastral exploration at the macro-level (Akahoshi et al., 2020; Nega and Coors, 2022).

However, regulatory checks concerning the traffic aspect are limited by data availability. Research on street space has predominantly focused on areas such as autonomous driving and navigation, closely aligned with the automotive industry (Yurtsever et al., 2020). Detailed semantic representation of roads and street spaces in 3D city models is relatively lacking. Street space datasets face constraints not only from data availability but also from potential conceptual flaws in foundational modelling frameworks and data provision standards or formats (Beil et al., 2020; Labetski et al., 2018). This poses challenges for conducting regulatory compliance checks in street spaces. OpenDRIVE, an important data model for describing and storing road networks, can now be converted to the CityGML format with the release of the r:trån tool (ASAM OpenDRIVE®, n.d.; r:trån, 2020). This conversion increases interoperability and broad applicability of road network data across different data models. With the publication of CityGML3.0, the updated Transportation module can meet diverse application needs to the greatest extent (Beil et al., 2020; Kutzner et al., 2020). As data availability improves and CityGML3.0 updates the Transportation module, advancements in semantics and topology for city objects facilitate regulatory checks at the city scale.

Grounded in this context, this study addresses the gap in automated regulation assessments of parking facilities in the city. It employs the CityGML3.0 data model to achieve this process, taking advantage of its extension capabilities to standardise regulations. The final output aims to assist government officials in verifying compliance of parking facilities with regulatory requirements in real-world scenarios using an automated tool, which is a repeatable process. This approach enhances work efficiency, accelerates processes, and reduces costs through improved management. Additionally, these outcomes can be applied to other construction stages, such as building permit checking.

1.2 Research Objectives

The objective of this research is to achieve a regulatory assessment of parking facilities at the city scale using CityGML 3.0. This approach ensures the accurate translation of regulations to achieve desired outcomes, while also clarifying the entire process and its underlying rationale. Within this context, the thesis therefore aims to answer the following research question:

How can parking facilities be assessed against regulations at the city scale using CityGML 3.0?

To address this main question, several sub-questions are further clarified:

- What are the relevant regulations regarding parking facilities in the city?
- How can the objects and their attributes extracted from the regulations be mapped into CityGML 3.0?
- How can the assessment be realised and automated against the regulations?

1.3 Methodology

The research is accomplished by mapping paper-based regulations into CityGML 3.0, culminating in final regulatory assessments. The overview is illustrated in Figure 1-1.

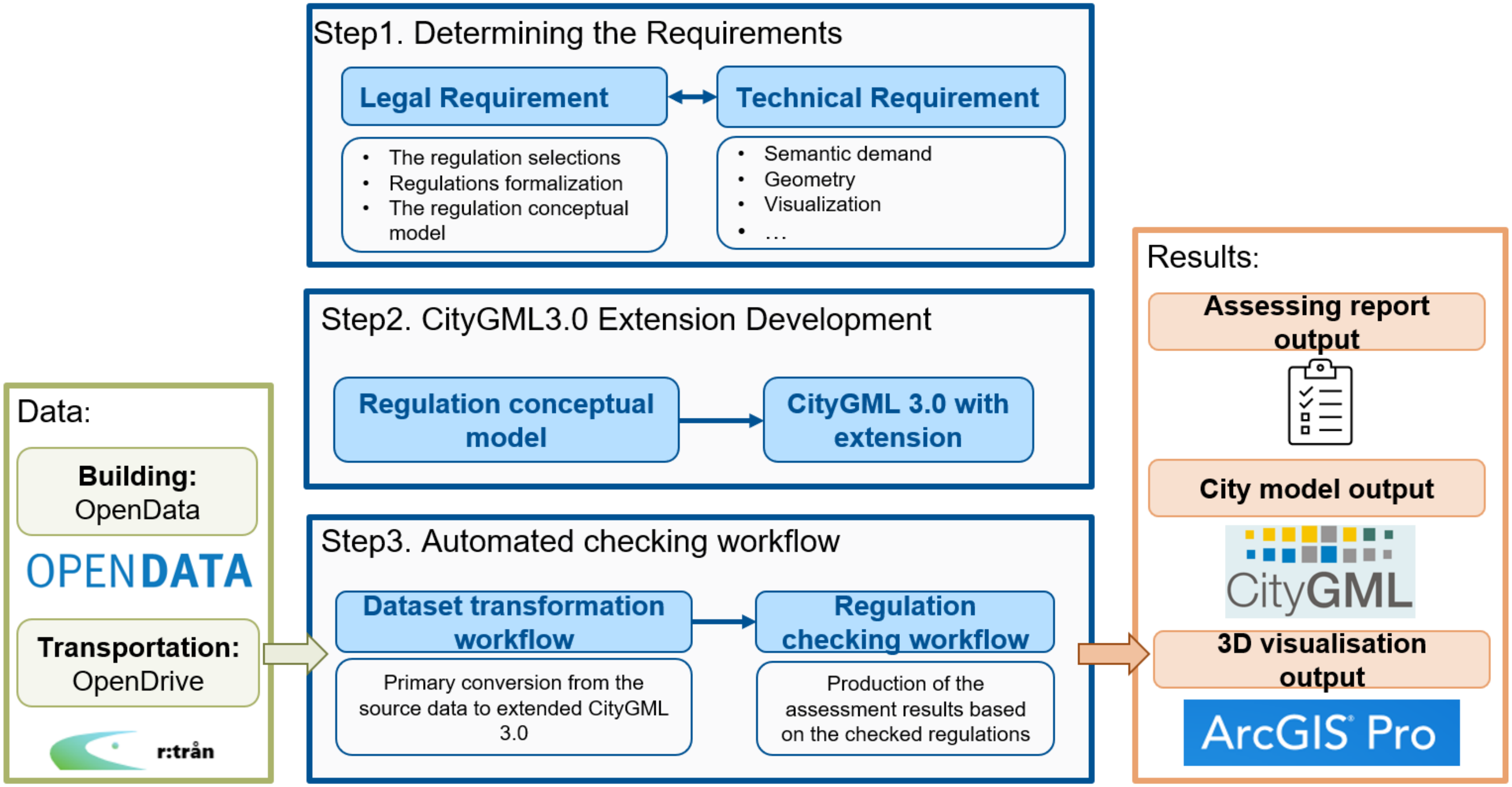


Figure 1-1 Workflow of the methodology

Specifically, the research focuses on three main aspects:

1. Identifying the legal and technical requirements for regulation checks;
2. Mapping the relevant regulations to CityGML and its extensions;
3. Developing a workflow for the regulation-checking process.

The processes use a simplified logic rule-based mechanism (Lee et al., 2016). The first step involves formulating the requirements from legal and technical perspectives that are important for regulatory checking. In the second step, the selected articles from the specific regulations are mapped onto CityGML 3.0, and model extension is carried out as necessary. The data model is then applied to the checking process. Finally, a workflow is developed to automate regulation checks based on specific regulations. This workflow generates a range of outcomes, including an assessment report, a data model storing the check results, and interactive 3D visualisations.

In summary, through this work, I aim to supply an application of the CityGML3.0 Transportation Module. Furthermore, it also shows the great potential of CityGML3.0 in the regulation-checking field to serve as an information resource for urban planning efforts. All steps are mainly implemented in the ETL software Feature Manipulation Engine Workbench (FME workbench). The interactive 3D visualisations are presented in ArcGIS Pro.

1.3.1 Determining the Requirements

It is essential to identify the requirements that support the development of the assessing process for regulation checking. This will determine the specific content and presentation of the assessments. Two types of requirements need to be identified: legal and technical requirements. These are derived from the current Bavarian regulations and the literature review.

For regulatory requirements, the specific regulations need to be identified. Each object and its related properties will be classified under the regulation instead of using the CityGML city model directly, following Lee's methodology in 2016 (Lee et al., 2016).

1. Selection of regulations: This study primarily utilises Bavaria's parking regulations, i.e., the Garage and Parking Slot Ordinance (Verordnung über den Bau und Betrieb von Garagen sowie über die Zahl der notwendigen Stellplätze, abbr. Garagen- und Stellplatzverordnung, GaStellV), along with specific local regulations for the city of Ingolstadt, known as the on the Garages and Parking Slots Provision (Satzung über die Herstellung und Ablösung von Garagen und Stellplätzen, abbr. Garagen- und Stellplatzsatzung, GaStS);
2. Interpretation and standardisation of regulations: This involves deconstructing and reorganising the regulations into a machine-readable format, i.e., creating a regulation conceptual model.

Then, the technical requirements would be explored, including data modelling requirements, such as geometric representations, coordinate systems, and more. The demands of regulation checks and how the data model can fulfil these needs will be examined. The technical requirements are broken down into six aspects as described below:

- Semantic representation

- Geometric representation
- Spatial scope
- Topological relations
- Visualization
- Dynamic management

1.3.2 Mapping and Extending into CityGML

The data model must be conceptually defined following the identification of legal and technical requirements. This involves mapping the identified classes, attributes, and relationships to CityGML 3.0 and extending it with any missing classes or attributes, adhering to CityGML 3.0 standards. The process typically follows the steps illustrated in Figure 1-2:

2. A correspondence is established between semantic objects defined in the regulations and CityGML classes. This constitutes the first mapping, aiming to integrate the classes and attributes from the regulations into the CityGML Conceptual Model as accurately as possible, guided by semantic correspondence.
3. In this step, the specific subclasses where to store the information missing in the CityGML Conceptual Model are chosen. It is determined which extended mechanism applies to enrich the model. Extensions can be made either through generic classes and attributes or by employing the Application Domain Extension (ADE) mechanism.

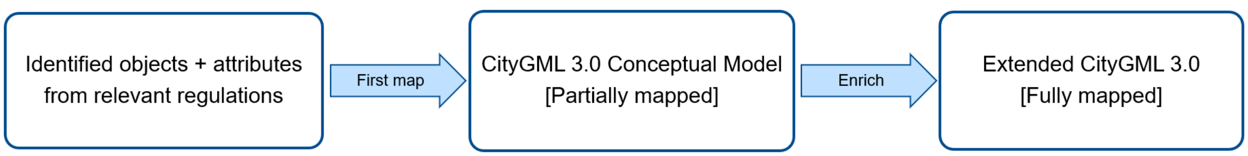


Figure 1-2 The mapping process

1.3.3 Creating a workflow for the automated checking process

The final step is to create a workflow for regulation checks and ultimately generate output. The checking workflow is developed using the ETL software FME Desktop v2023.0.1 (FME Software, 2023). This software supports importing different versions of CityGML and manipulating the data. Furthermore, it allows exporting assessment reports in multiple formats, showcasing the outcomes of the regulation checks. The workflow can automatically complete the entire checking process. The structure is shown following:

- Input: The reader in the FME workbench serves as the workflow's input, including two main datasets: transportation and building.
- Transformer: In FME, various transformer combinations assess different regulations, such as the parameters of the parking slot, the calculation of the parking slot number, and the constraints of ancillary objects.
- Output: The writer in FME produces the final workflow output, displaying the assessment report, the extended data model linked checking results, and the Geodatabase files for further 3D visualisation in ArcGIS Pro v 3.2.2 (ArcGIS Pro, 2023).

While this thesis generally adheres to the outlined workflow, the process is not inherently linear. It allows for identifying and correcting issues or inconsistencies in each step to refine the process in time.

1.4 Reading Guide

The remaining thesis is structured as follows.

Chapter 2 offers extensive insight into relevant concepts, theories, and related work. It focuses on CityGML3.0 and its extensions, the development of the automated compliance checking process, and similar works. Following the description of Section 1.3, Chapter 3 describes the requirements for parking facility assessment, covering the legal and technical aspects. Chapter 4 provides the mapping from the regulation into the conceptual model and extensions of the CityGML 3.0. Chapter 5 describes the implementation process and results, including the conversion of the source datasets, as well as the development of the workflow to achieve the automated regulation check in depth. In Chapter 6, the implications and limitations are discussed. Finally, Chapter 7 provides a summary with the answers to the research questions and outlines future work.

2. Scientific Context and Related Works

This chapter establishes the foundational theoretical knowledge, including relevant frameworks, 3D city models, and related works, alongside a literature review. It begins with Section 2.1, which provides a comprehensive overview of semantic 3D city modelling, with a particular focus on CityGML 3.0. Following this, Section 2.2 discusses the various aspects of regulation checking, detailing the steps and principles involved. The final section reviews similar works in the fields of 3D city modelling and regulation checking.

2.1 Semantic 3D City Modelling – CityGML 3.0

Over the past decade, the adoption of semantic 3D city models has become increasingly widespread. Distinguishing from the regular 3D models, the semantic 3D city models represent more than geometric shapes. City objects within semantic 3D city models, such as buildings, vegetation, city furniture, etc., are defined through a combination of geometry, topology, semantics, and hierarchical structures (Kolbe, 2009). Currently, the CityGML city model is the only available open standardised data model in the city modelling field (Widl et al., 2021).

This section provides an introduction to CityGML, covering its latest 3.0 version, extension mechanisms, revised Transportation module, which is the primary focus of this study, and applications.

2.1.1 CityGML and its version 3.0

CityGML has been an OGC (Open Geospatial Consortium) standard since 2008. As an object-oriented model, it is both an XML-based exchange format and an open data model for semantic 3D city modelling at the city and larger scales (Gröger and Plümer, 2012). The first version was released in 2008, followed by version 2.0 in 2012. Since then, it has been applied in various fields (see section 2.1.4). After nine years of revisions, the fully revised and updated version 3.0 was published in 2021 (Kolbe et al., 2021).

CityGML 3.0 represents a significant advancement in urban modelling and geospatial data management, providing advanced capabilities for the thematic and semantic representation of urban structures and interoperability with other standards like IFC, LADM, and IndoorGML (Kutzner et al., 2020). It defines the graphical appearance, semantic and thematic attributes, classifications, aggregations, and hierarchy of 3D city models. One of its revolutionary strengths is the standardisation of the underlying information model logic via conceptual UML models (Unified Modelling Language) (OGC, 2022a). CityGML 3.0 separates the conceptual model and data encoding formats based on ISO 191XX specifications. The CityGML 3.0 Conceptual Model describes general definitions of the basic entities, their attributes, and relationships for representing 3D city objects. Then, the independent encoding standard allows data to be encoded in GML/XML, JSON or other database schemas, whereas CityGML versions 1.0 and 2.0 only allow encoding data using GML (OGC, 2022a).

The conceptual model defines 17 modules, including the Core module, nine thematic modules, and seven extension modules (as shown in Figure 2-1). It uses the modelling language UML, since a model-driven approach is used to create the data model and exchanged formats for CityGML 3.0

(Kutzner et al., 2020). Some new extension modules have been introduced to enhance applications by adding new information and attributes, these modules can be attached to semantic modules. For instance, the Versioning and Dynamizer modules can inspect and model the qualitative and quantitative changes of a city or city objects as time progresses. The semantic modules now include a Construction module that defines the common classes of all man-made constructions, integrating similar classes from various types of constructions (e.g., buildings, tunnels, and bridges). This new module avoids the multiple definitions of these common classes in each individual module, thereby simplifying the UML model of the Building, Tunnel, and Bridge modules (Kutzner et al., 2020).

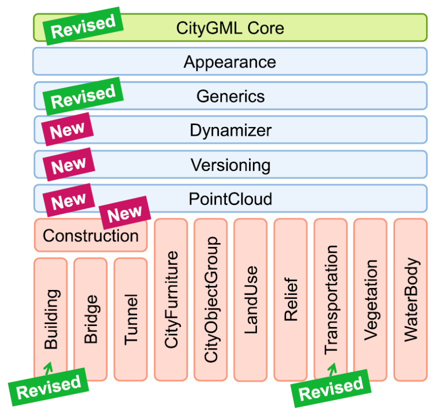


Figure 2-1 CityGML 3.0 module overview. The red boxes represent the different thematic modules, while the blues specify extension modules that define concepts applicable to all thematic modules (Kutzner et al., 2020)

The existing modules in CityGML 3.0, including the Core module and some thematic modules, have been revised and updated. In the Core module, the concept of space has been redefined using two key abstract classes, i.e., *AbstractSpace* and *AbstractSpaceBoundary*, to represent the spatial information for semantics and thematics. The *AbstractSpace* class can then be further subdivided (see Figure 2-2). All thematic modules can derive from these space and boundary classes. This significant change means that all geometric representations only need to be clarified in the Core module and can be inherited by all thematic modules, simplifying the thematic modules and implementing efficiency (Kutzner et al., 2020).

The Core module has also revised the Levels of Detail (LoD), removing LoD4 and retaining only LoD0 to LoD3. In the previous version, LoD 4 is used to represent the interior structures. Now, both the building shell and the interior structures can be modelled and expressed with different LoDs directly, as LoD only describes the geometric resolution instead of the thematic resolution (Kutzner et al., 2020; Löwner et al., 2016). Based on the improved settings of the Core module, improvements have been made to the modelling of street spaces as well. The Transportation module is reviewed in detail in Section 2.1.2.

CityGML is designed to retain the minimum common feature types, keeping the conceptual data model small (Kolbe, 2009). This means additional information and modelling are required for many specific application domains (Biljecki et al., 2018). Therefore, CityGML also provides multiple extension mechanisms. These mechanisms have also been updated in CityGML version 3.0, and their details are reviewed in Section 2.1.3.

Through these updates and revisions in new space concepts, new and revised modules, and enhanced extension mechanisms, more potential applications could be explored. Examples include urban planning, land administration with legal boundaries and spaces, and the analysis of street spaces (Akahoshi et al., 2020; Beil et al., 2020; Saeidian et al., 2023).

3DCityDB (3D City Database) is an open-source database solution designed for storing, managing, and querying 3D city models (Yao et al., 2018). Based on PostgreSQL/PostGIS, 3DCityDB supports the CityGML format, effectively handling complex urban data. The 3DCityDB Web-Map-Client is also provided, based on the Cesium Virtual Globe framework using HTML5 and WebGL technologies, allowing interactive visualisation of 3D city models.

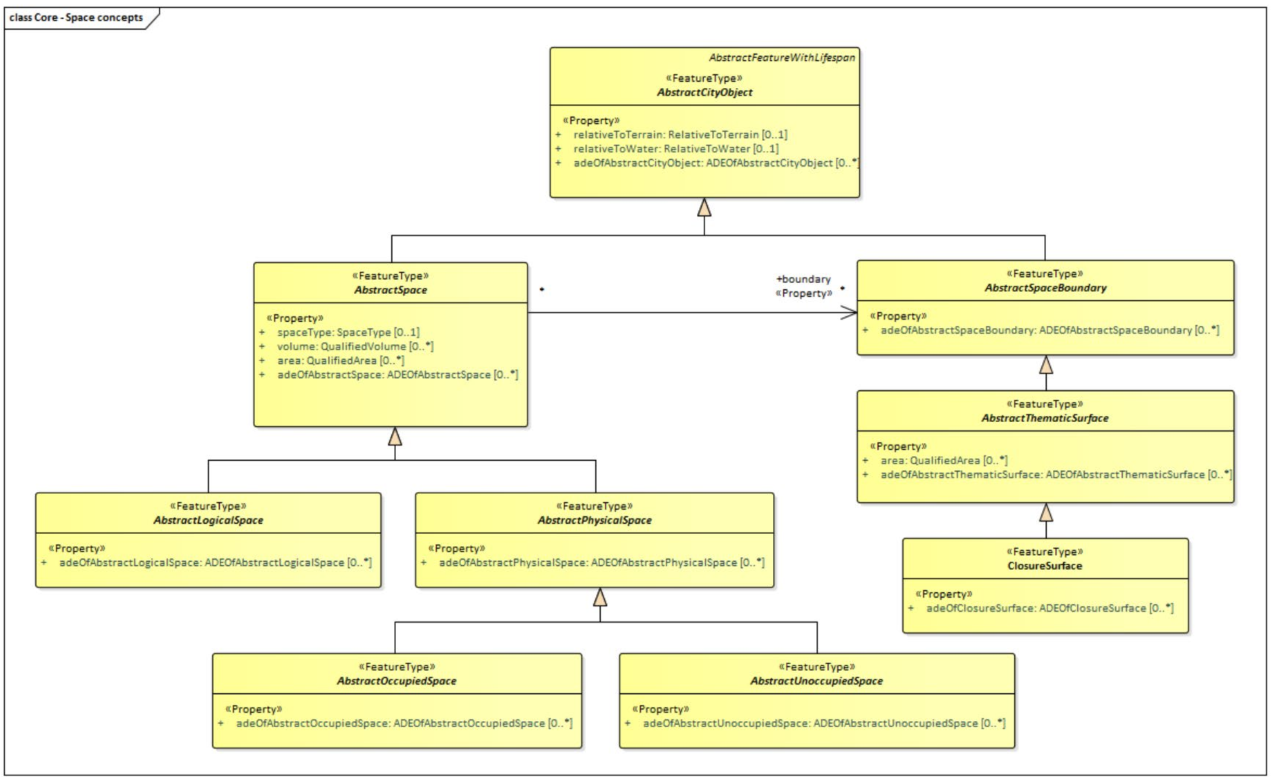


Figure 2-2 The UML diagram of Space Concepts in the Core module (Kolbe et al., 2021)

2.1.2 Revised Transportation Module

With the increasing requirements of applications in street spaces, the original transportation module in CityGML 2.0 has shown the limitations and weaknesses for their needs. In 2013, Tamminga proposed an ADE for transportation and traffic models, providing a generic complement to the transportation module in CityGML 2.0 (Tamminga et al., 2013). This marked the beginning of the transportation application in CityGML. In 2017, Beil and Kolbe identified further development potential for detailed street models based on CityGML 2.0, introducing various potential applications and proposing conceptual guidelines for extending the transportation model (Beil and Kolbe, 2017). In 2020, these guidelines were further refined and extended (Beil et al., 2020). The revised transportation module was released following the official publication of CityGML 3.0 (Kolbe et al., 2021).

In CityGML 3.0, the Transportation module has been significantly enhanced and modified to provide a more detailed and comprehensive representation of urban transportation infrastructures, improving the usability in various fields, such as infrastructure planning and management, land administration, autonomous driving, and environmental simulation (Beil et al., 2020). The Transportation module defines the core elements within the transportation domain, as illustrated in its UML diagram shown in Figure 2-3.

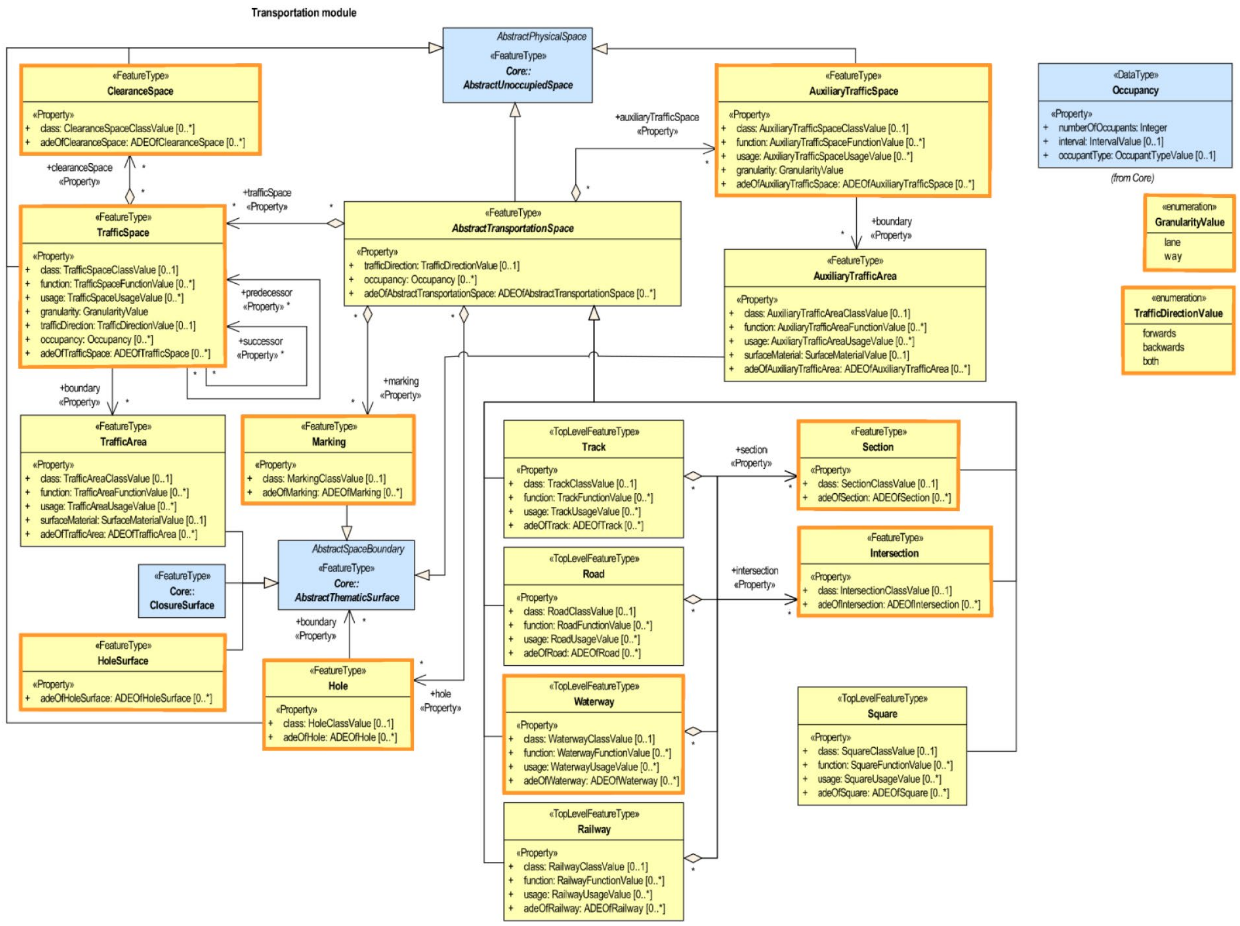


Figure 2-3 UML Diagram of the Transportation module (Kutzner et al., 2020), the new classes in CityGML 3.0 are indicated by boxes with orange lines (Beil et al., 2020)

Concrete transportation objects are derived from the class *AbstractTransportationSpace*, previously known as *TransportationComplex* in CityGML 2.0. These transportation objects include roads, tracks, and even railways and waterways. They can be further semantically decomposed into sections and intersections (classes *Section* and *Intersection*). To avoid redundancy, an i*ntersection* shared between different roads is modelled once and connected using a linking concept through the XLinks (Kolbe et al., 2021). These classes can then be decomposed into *TrafficSpace* and *AuxiliaryTrafficSpace* according to the different traffic functions. They are new classes introduced in CityGML 3.0 to align with the Space concept in the Core module. TrafficArea and AuxiliaryTrafficArea now act as their ground boundaries. Above *TrafficSpace* and *AuxiliaryTrafficSpace*, a class *ClearanceSpace* has been introduced to represent the actual space for safe transportation. The specific geometric representation can be seen in Figure 2-4. Different types of traffic surfaces are defined by the attribute *Function*, *class*, and *usage* from their respective *TrafficArea* and *TrafficSpace*. The newly added classes *Marking*, *Hole*, and *HoleSurface* provide more detailed information about the street space. With the update of LoD (see Section 2.1.1), new attributes have been added, such as the attribute *granularity* to represent different levels of thematic resolution in this module.

Additionally, many technical details have been updated in the module as well. For example, in the highest LoD, each TrafficSpace can be interconnected through ancestor-descendant relationships, which, to some extent, provides potential applications for CityGML in navigation (Beil et al., 2020; Labetski et al., 2018). As mentioned in Section 2.1.1, the new extension modules, such as the Versioning module, allow street spaces to participate in urban version management. Through the linkage with external sensors, dynamic and real-time monitoring can be performed combined with the Dynamizer module, and attributes can be stored to support traffic monitoring and prediction.

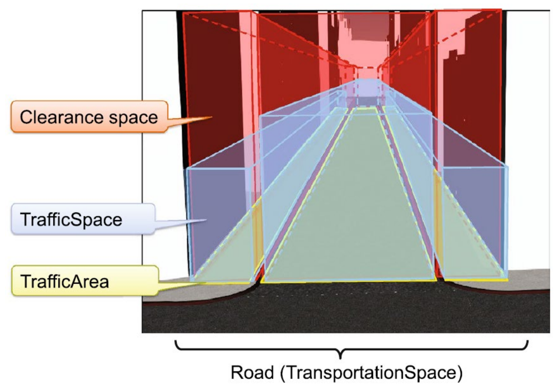


Figure 2-4 The concept explanations for a road in CityGML3.0 (Kutzner et al., 2020)

2.1.3 Extension Mechanism

CityGML is an application-independent and semantically rich data model (Biljecki et al., 2018). However, CityGML is not able to meet the needs of all specific applications. In certain specific application areas, additional information is required, including the objects and attributes not explicitly modelled and defined in the CityGML Conceptual Model thematic modules. In response, CityGML provides a series of official solutions to extend data models with additional objects and attributes to support these applications. CityGML 3.0 introduces three mechanisms for extension: 1. Using generic objects and attributes, 2. Developing Application Domain Extension (ADE), and 3. Creating code lists (Kolbe et al., 2021).

Using the Generics module is the simplest official method for extending the original CityGML data model without changing the default schema. In a running application, new city objects can be defined through generic city objects, or an arbitrary number of new attributes can be added to any predefined city objects via generic attributes. Generic city objects can be derived from the Space Concept in the Core module, forming four top-level feature types to define spaces and surfaces not represented by any existing classes in the CityGML Conceptual Model. The UML diagram is shown in Figure 2-5. Generic attributes are structured as name-value combinations attached to a city object.

They totally have seven data types and can be grouped as a set of generic attributes, as illustrated in Figure 2-6. This mechanism is particularly practical for applications that require only minor extensions to the data model. However, when using the Generics module, naming generic objects and attributes requires extra caution to avoid potential naming conflicts. Furthermore, only classes and attributes that are not represented in any modules in the CityGML conceptual model should be defined as generic city objects and generic attributes to avoid semantic interoperability issues (Kolbe et al., 2021). Figure 2-7 shows a typical workflow in FME.

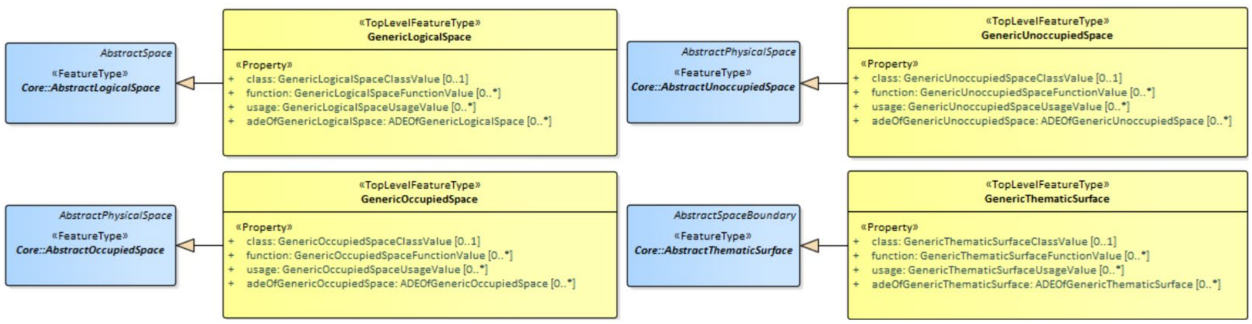


Figure 2-5 The UML diagram of the generic city objects in the Generics Module (Kolbe et al., 2021)

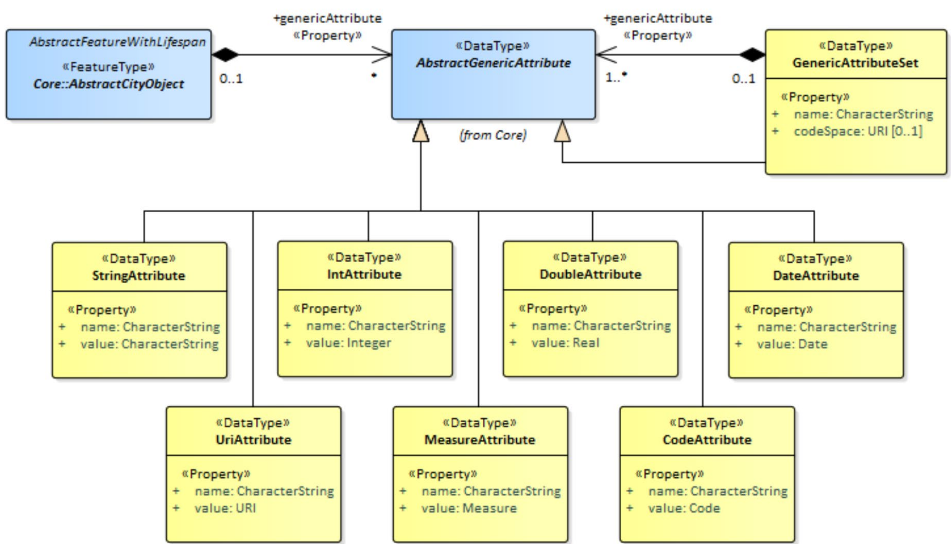


Figure 2-6 The UML diagram of the generic attributes in Generics Module (Kolbe et al., 2021)

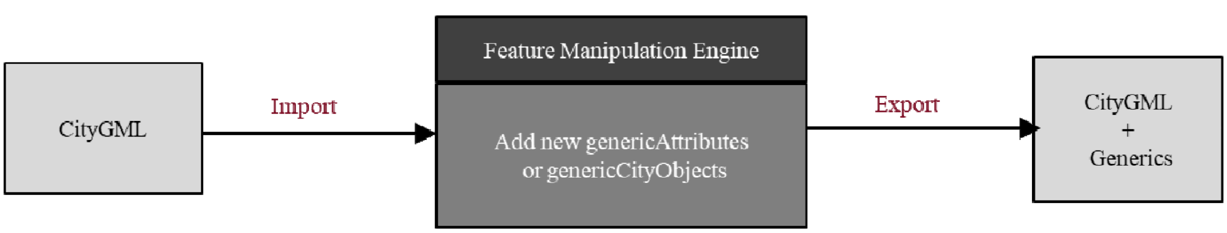


Figure 2-7 A typical workflow to extend CityGML with the Generics module in FME (Padsala et al., 2021)

ADEs represent a systematic and comprehensive method for extension of the CityGML data model for specific applications or domains. They are more practical for applications requiring extensive extensions or those that are standalone systems, such as EnergyADE (Agugiaro et al., 2018). Its extension mechanism introduces domain-specific objects and attributes required by specific applications but not presented in the CityGML Conceptual Model. It allows for the extension of existing objects and attributes within the classes in the CityGML Conceptual Model as well. The main distinction between ADE and the Generics module is whether a new schema is created to store information for specific applications (see Table 2-1).

In CityGML 3.0, a model-driven approach is employed to develop a new ADE, which separates the conceptual model from encoding (Van Den Brink et al., 2013). This approach supports ADEs as an independent model at the conceptual level that can be encoded into various formats rather than limited to outputting only the XML schema as in previous versions (Kolbe et al., 2021). A simplified typical process involves first representing the data model of the specific application in a UML diagram using Enterprise Architect (Sparx Systems, 2024), then automatically deriving GML application schemas using ShapeChange (Interactive Instruments, 2024). Finally, validation is performed to ensure compliance with CityGML standard specifications and definitions.

The typical workflow is shown in Figure 2-8. After the creation of ADE, all relevant elements can be accurately mapped to these additionally defined objects, attributes, and relationships in accordance with CityGML standards. Since an ADE effectively creates a new schema to store information for specific applications, it retains the conceptual and semantic structure of the CityGML standard well. Compared to the extension using the Generics module, ADE has a better ability to ensure semantic interoperability, as the application data can be fully validated according to formal definitions. This makes ADE to be a common mechanism supporting a wide range of specific domains and applications.

Table 2-1 The comparison of the extending mechanism between the Generics model and ADE

Generics Object and Attributes	**ADE**
Create the new objects and attributes without changing the original CityGML schema.	Change the original CityGML schema with domain-specific new objects, attributes, and relationships,
Keep the same XML namespace with CityGML Conceptual model	ADE-specific unique XML namespaces
Can be specified during the implementation phases	Must be specified using UML diagrams and corresponding XSD firstly

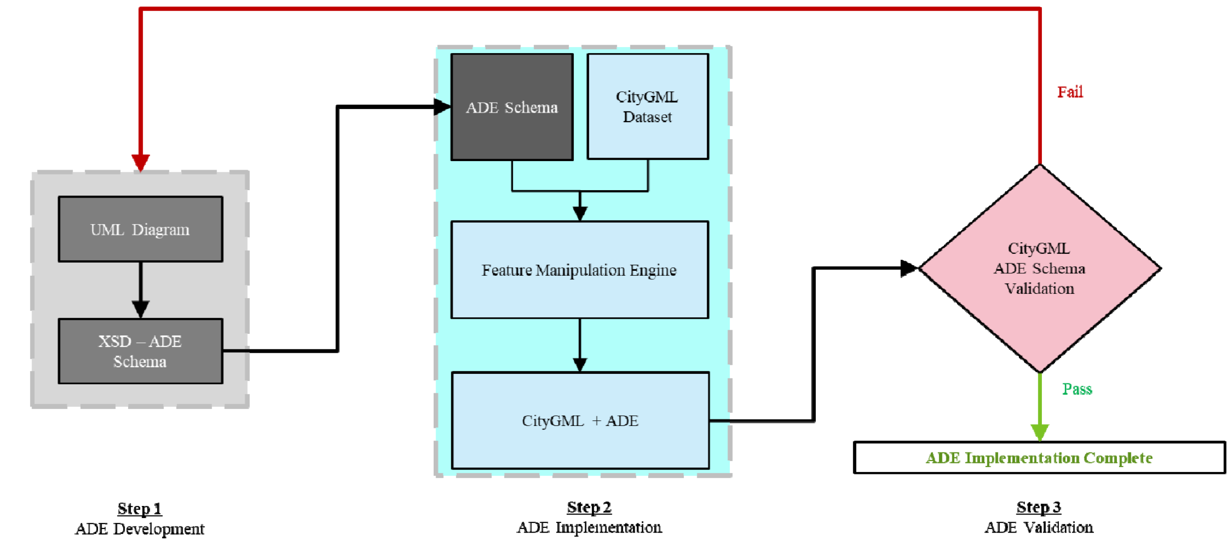


Figure 2-8 A typical workflow to extend CityGML with ADEs (Padsala et al., 2021)

CityGML 3.0 can also be extended using specific code lists. This represents a narrower scope of extension, targeting existing attributes that lack certain value domains. A code list can be used as a data type for some attributes in CityGML, providing a series of codes and their corresponding value. This approach is most commonly used for attributes such as function, usage, and class. It acts as an enhanced version of fixed enumerations but allows for the modification and extension of the value domain. Global, national, or industry-standard classifications can be integrated into CityGML 3.0 through the definition of the code lists (Kolbe et al., 2021).

2.1.4 CityGML Applications

Planning and development of a city requires the combination and assessment of multiple city elements, such as land use patterns, buildings, transportation networks, vegetation cover, and topography (Sinning-Meister et al., 1996). As cities grow and evolve, the relationship between the environment and its human inhabitants is becoming increasingly complex and variable. It requires considering an increasing number of factors, such as energy consumption, pollution levels, network infrastructure, and legal frameworks. Consequently, comprehensive and accurate 3D city models should support not only visualisation but also common applications, including simulation, analysis, and modification.

A key feature of CityGML is the rich semantic foundation and the integration of geometry and semantics, as introduced in section 2.1.1. It enables the effective use of urban geodata in smart cities and urban digital twins (OGC, 2022a). Today, with the rapid development of information technology and GIS, as well as the continuous improvement in data availability, CityGML has been widely adopted and shows increasing activity in various applications.

In urban planning, Japan has introduced the "i-Urban Revitalization" (i-UR) project, an information infrastructure designed to assist municipal authorities in visualising and analysing urban development plans (Akahoshi et al., 2020). One strategy for data standardisation involves developing CityGML and extending an ADE named Urban Planning ADE. This conceptual model supports the storage, simulation, and visualisation of information about urban planning, including urban area functions, resident populations, transportation accessibility, and more (Akahoshi et al., 2020). Generated by CityGML 2.0, the new version has been released to conform with the new CityGML 3.0 standards and its revised ADE mechanism (OGC, 2022b).

In 2022, Greece published GRextADE for storing and exchanging the national standard, highlighting cadastral information (Liamis and Mimis, 2022). The GRextADE not only extends the functionality of base standards but also incorporates the national characteristics of Greek laws and regulations. There are many applications based on national standards similar to GrextADE. Although not designed for a specific purpose, these generic extensions broaden their application scope and enable interoperability with CityGML (Biljecki et al., 2018). For instance, the IMGeoADE in the Netherlands serves as an exemplary case of the CityGML data model adopting national 3D standards (van den Brink et al., 2013). This standard has been widely recognised and actively participated in by various societal sectors, including governments, businesses, and residents. This extension fully integrates CityGML into the existing 2D national information model for large-scale terrain. It is applicable not only to national large-scale terrain databases but also to national data exchange systems further.

Additionally, climate and environmental simulations, including analyses of city energy, noise, and solar, have garnered attention. Introducing new features is essential, and the ability to visualise simulation results and data models is also one criterion for effective modelling. The field of energy management has always been a focal area for CityGML applications. Currently, there are multiple ADEs supporting the analysis of urban energy demand and supply (Agugiaro et al., 2018; Becker et al., 2013; Kutzner and Kolbe, 2016). Among them, EnergyADE fills the gap in the energy data model based on open standards on the city scale, enriching the data model with information necessary for energy simulations, thus facilitating the establishment of Urban Building Energy Modelling (UBEM) (Agugiaro et al., 2018). The conversion of the EnergyADE from CityGML 2.0 to 3.0 was completed in 2023 (Bachert, 2023).

Furthermore, CityGML can also play a role during the built environment process, such as the exploration of urban model versioning and lifecycle management. It is allowed directly to be stored using the Versioning module in CityGML 3.0 now. Also, it can be integrated with other data models, such as BIM. This integration expands the potential application fields of 3D city models. It holds significant ability in areas such as building permit processes and the creation of 3D cadastres, despite facing some technical challenges, details shown in section 2.3 (Uggla et al., 2023).

Research on city digital twins is also intriguing, as it can facilitate real-time remote monitoring, making decision-making more effective (Deng et al., 2021). It requires the visualisation of dynamic sensor information and the analysis of sensor data to integrate urban geospatial data. The most direct application is the monitoring of real-time traffic (Beil et al., 2022).

2.2 Regulation Checking

2.2.1 Current status

The field of compliance checking has been continuously explored. One reason is that buildings and facilities must undergo various regulatory assessments throughout their lifecycle to ensure environmental sustainability and human safety (Dimyadi and Amor, 2013). These regulations consist of zoning information, which includes specific rules and standards at various levels.

For a long time, traditional compliance checking in the industry has been a fully manual process (Dimyadi and Amor, 2013). This method is time-consuming and labour-intensive, characterised by slow and complex workflows. Errors and misunderstandings can occur during the process, leading to delays and increased costs (Eirinaki et al., 2018; Malsane et al., 2015).

Over the past forty years, there has been ongoing exploration into the digitisation and automation of regulation compliance checking. As the architecture, engineering, and construction (AEC) industry has shifted from 2D computer-aided design to more semantically rich building models, along with the development of various algorithms, processes, and platforms, significant progress has been made (Amor and Dimyadi, 2021; Malsane et al., 2015).

Digitisation of compliance checking involves using electronic systems and tools to interpret regulations. When employing 3D modelling, 3D models can provide detailed semantic information and geometric data, allowing for more comprehensive compliance checks (Guler and Yomralioglu, 2022). Currently, the BIM model is preferred for storing, maintaining, and querying building information (Guler and Yomralioglu, 2021). They can also integrate specific regulations with other planning data, such as land use and zoning information (Brasebin et al., 2018).

Many countries and regions are exploring the digitisation of compliance checking, with a primary focus on the building permit process. The EU's 2002 Action Plan recommended the digitisation and automation of public services to improve their efficiency (European Commission, 2002). Subsequently, the EU Digital Building Permit (EU4DBP) and Change toolkit for digital building permit project (CHEK) were proposed, focusing specifically on building permits (CHEK, 2022; EU4dbp, 2022). The former emphasises developing common standards and guidelines for digital building permit systems and providing technical assistance to member states for implementing these systems (EU4dbp, 2022). The latter aims to develop a standardised digital platform for exchanging information related to building permits, including compliance checks (CHEK, 2022).

Automation is a crucial component of digitisation. Specifically, automated code compliance checking, based on regulations established by government agencies, provides computational and platform support for assessing projects (Altıntaş and Ilal, 2021). The CORENET project by the Singapore Building and Construction Authority in 1995 was a pioneering effort in this field. It provided a web-based electronic system for conducting code checks in multiple areas, such as building control and vehicle parking, on submitted building drawings (Liebich et al., 2002). The new generation of this project, called CORENT-X, is currently under development (CORENET X, 2024). In 2015, Malsane et al. created a semantically rich object model specifically for building fire codes, which could meet the requirements for automated enforcement in England (Malsane et al., 2015). Additionally, a pilot project in Italy demonstrated its advantages in building management (Ciribini et al., 2016).

Currently, research on automated regulation compliance checking primarily focuses on building properties (Benner et al., 2010; Olsson et al., 2018), fire safety (Malsane et al., 2015; Nguyen and Kim, 2011), and other aspects of the building, while less attention is given to ancillary facilities around the buildings, such as parking areas and greenery.

2.2.2 Phases of checking process

Approaches for automated compliance checking have been continuously explored and discussed. In 2009, Eastman proposed a notable framework for the rule-based approach to automated compliance checking, generally structured into four steps (Eastman et al., 2009).

1. Regulation interpretation and formalisation, including the corresponding logical structure;
2. Preparation of the data model, involving the extraction and storage of the required information for the checking process;
3. The checking procedure, i.e., the execution stage;
4. The checking result output.

Additionally, many similar and improved approaches have been proposed and developed (Beach et al., 2015; Lee et al., 2016; Peng and Liu, 2023). Through a series of reviews and summaries, Amor and Dimyadi identified two key data sources for a typical automated compliance checking process: data models and normative knowledge, as shown in Figure 2-9 [a] (Amor and Dimyadi, 2021). Normative knowledge is the computable form of regulations and codes, representing what needs to be checked. Data models are the digital representations of the entity being inspected.

Based on this framework, most research focuses on handling these two components, which are also the first two steps in Eastman's framework. Common approaches involve extracting objects and attributes from these data sources, which can access information from each other or mutually map between each other (Amor and Dimyadi, 2021). Currently, the latter is a mainstream research direction, with the typical workflow shown in Figure 2-9 [b].

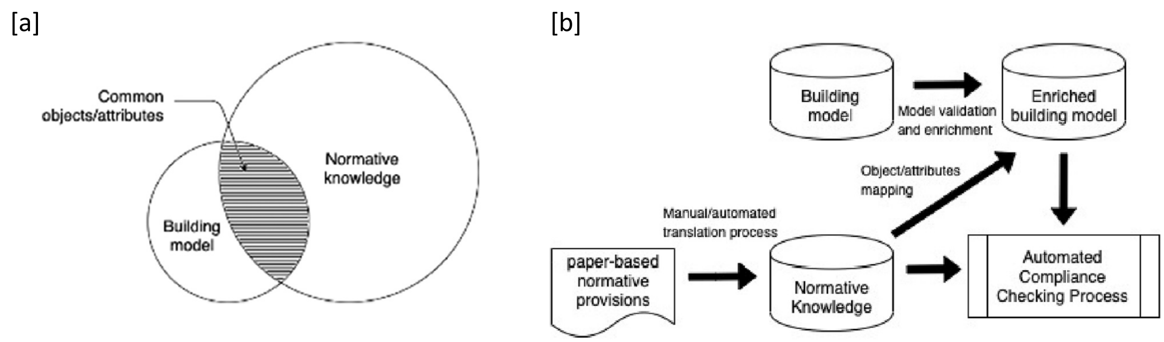


Figure 2-9 [a] Typical data sources for the automated process of the compliance checking; [b] Typical automated process of the compliance checking (Amor and Dimyadi, 2021)

2.2.3 Translation of the Regulations

One of the challenges in compliance checking for existing constructions is extracting valid normative knowledge from natural language. This involves interpreting and formalising legal texts, and translating them into structured data that machines can automatically read and interpret (Amor and Dimyadi, 2021; Zhang et al., 2023). Normative knowledge may include various forms of content, such as natural language text paragraphs, tables, illustrations, and prescribed calculations (Amor and Dimyadi, 2021).

Several approaches are explored to define constraints in regulations and translate them into machine-readable form. Systematising regulatory checks is crucial, requiring the process test and optimisation, and personalising it based on specific use cases (Solihin and Eastman, 2015). Solihin and Eastman analysed the evolution of rule-checking systems, categorising them from simple explicit data checks to advanced checks involving extended data structures or complex compliance checking proofs. This classification helps in understanding the different levels of interpreting and translating regulations (Solihin and Eastman, 2015).

Zhang and El-Gohary adopted a rule-based semantic natural language processing (NLP) approach, developing semantic mapping (SeM) and conflict resolution (CoR) rules (Zhang and El-Gohary, 2016). This approach can extract and convert regulatory text into logical clauses suitable for automated reasoning, thereby reducing time consumption and error rates. Meanwhile, Preidel and Borrmann also emphasized the importance of NLP and visual programming languages (VPL), responding to the progress made in translating regulations for automated compliance (Preidel and Borrmann, 2016). They highlighted the need for domain experts' involvement in this process, especially given the complexity and potential ambiguity of regulations. They discussed the critical process of translating building codes and guidelines into machine-readable language further, contrasting opaque black-box methods with more visible white-box methods, which allow users to read and implement the rules more flexibly (Preidel and Borrmann, 2018).

Lee et al. proposed a logic-based mechanism to translate the natural language content of the Korean Building Code into a computer-executable format (Lee et al., 2016). The rule formalisation and rule-checking processes are separated. First, building objects and related attributes in the Korean Building Code are categorised, and then the natural language sentences are translated based on logic rules. Finally, they are stored in the logical Meta-database called “KBimCode”. Figure 2-10 illustrates the logical process of this study.

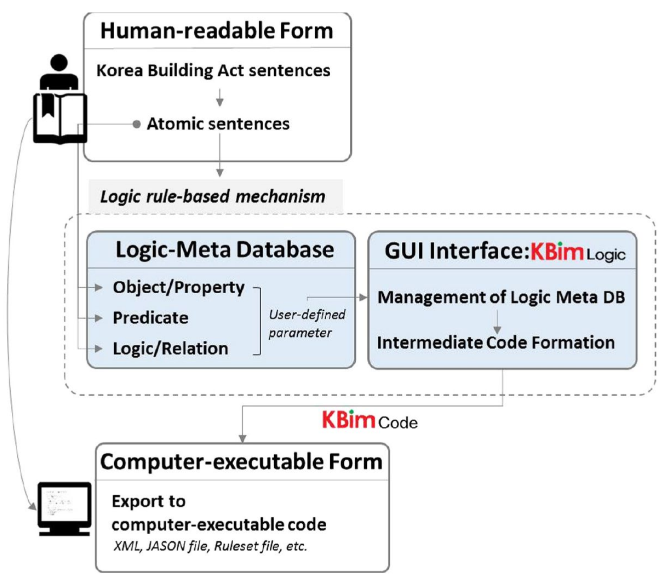


Figure 2-10 The overview of translating regulations from human-readable into computer-executable form (Lee et al., 2016)

2.3 Similar Works

The initial exploration of compliance checking was in the traditional AEC industry (Dimyadi and Amor, 2013). Over the years, BIM has been the primary focus of the compliance-checking systems. Most of the research mentioned above is based on this model (Ciribini et al., 2016; Lee et al., 2016; Malsane et al., 2015; Peng and Liu, 2023; Preidel and Borrmann, 2018, 2016).

BIM provides a detailed symbolic representation, focusing on the specific attributes of building components rather than on abstract spatial entities. This makes BIM particularly useful for compliance checks on individual buildings. In contrast, the semantic 3D city model is capable of representing spatial data at much larger scales, such as entire cities and even nations. There have been attempts to combine BIM and semantic 3D city models for compliance checking. In 2013, Van Berlo et al. proposed a workflow to achieve the 3D checks of building permits based on CityGML and other 3D models at the city scale (Van Berlo et al., 2013). After that, Noardo et al. integrated geoinformation with BIM to digitise and automate the planning permit process with a use case in Rotterdam, particularly focusing on building height, overhang, and tower ratio regulations (Noardo et al., 2020). The interpretation and formalisation mechanisms are similar to those proposed by Lee in 2016. A study also developed a workflow to combine the potential parking area with the building properties to achieve automated building compliance checking (Zhang, 2019).

Recently, there have been studies of compliance checking and regulation embedding based on semantic 3D city models, especially CityGML. A semantic data model of the 3D site plan was developed based on CityGML 3.0 and its ADE (SiteplanADE) in Müller's master thesis (Müller, 2021). Through this SiteplanADE, an algorithm was created to calculate clearance spaces of the planned building within a property parcel automatically. In 2021, Wu proposed a method for automatically checking the regulation of parking slot quantity using the CityJSON data model in her master's thesis (Wu, 2021). However, there is still a gap in the study of the automated assessment of parking facilities using semantic 3D city models.

Although embedding regulations in 3D city models cannot directly perform compliance checking, these models can store legal information and have the potential to be integrated with compliance checks. For example, the applications in 3D cadastral and real estate have been increasingly discussed in recent years. One reason is the vertical extension of building structures and the growing complexity of infrastructure, which require more precise registration based on legal status (Van Oosterom, 2013). Initially focusing on 3D registration and visualisation, research has shifted to legal and economic aspects. Guler and Yomralioglu proposed a tripartite cycle that integrates 3D property ownership, digital building permits, and updated 3D city models. This aims to introduce a reformative framework linking multiple concepts through the use of 3D city models, specifically CityGML and IFC models (Guler and Yomralioglu, 2021). In 2023, Saeidian et al. developed a new ADE in CityGML3.0 that allows for the modelling and data management of underground assets, linking the legal and physical data in practice (Saeidian et al., 2023). Using a case study, they successfully implemented this integrated 3D model in the state of Victoria, Australia.

3. Requirement

This chapter outlines a series of requirements intended to guide and support the development of the parking facility assessment use case, as described in Section 1.3.1. The specified requirements offer critically descriptive information vital for achieving this assessment. In this research, requirements are mainly categorised into legal requirements and technical requirements.

3.1 Legal Requirement

The legal requirements are a fundamental and primary aspect of the parking facility assessment, including parking slots and ancillary elements. These elements are governed by local laws and regulations. Therefore, examining these legal constraints within specific administration areas is necessary to identify the requirements. This study focuses on Bavaria, Germany, particularly the city of Ingolstadt.

The first sub-section of this chapter focuses on the selection of relevant regulations for parking standards. The second sub-section addresses the interpretation and formalisation of these regulations, detailing the process of deconstruction and reorganisation from natural language and classifying each object and its associated properties. Finally, the conceptual model of the regulation is produced. By clarifying and structuring these legal requirements, this section provides a comprehensive understanding of parking regulations in Ingolstadt.

3.1.1 The Regulation Selections

In Germany, the Federal Road Traffic Act (Straßenverkehrsgesetz, StVG) and Federal Road Traffic Regulations (Straßenverkehrs-Ordnung, StVO) establish general guidelines for vehicle parking in public areas. Since the late 1930s, developers have typically been required to prove the availability of sufficient parking slots during the construction or renovation of physical structures (Kirschner and Lanzendorf, 2020). Currently, each state has specific criteria to determine the exact number of required parking slots, allowing for fewer slots when a parking slot serves multiple user groups. State Road Laws (Straßengesetz, StrG) and Building Regulations (Bauordnung) further dictate the construction of parking slots, with local municipal parking regulations (Stellplatzsatzung) offering detailed guidelines.

Article 47 of the Bavaria Building Regulations (Bayerische Bauordnung, BayBO) mandates, "If facilities are built where access or departure traffic is to be expected, parking slots must be provided in sufficient number and size under a suitable condition." This establishes a legal framework for parking facility construction. Bavaria issued the Garage and Parking Slot Ordinance (Verordnung über den Bau und Betrieb von Garagen sowie über die Zahl der notwendigen Stellplätze, Garagen- und Stellplatzverordnung – GaStellV) to specify detailed provisions for parking slots. Local municipalities in Bavaria have the autonomy to make independent decisions on parking slot obligations, with the legal framework allowing them to reduce, exempt, or withhold parking slots as deemed necessary. Since 1995, the city of Ingolstadt has implemented the Garages and Parking Slots Provision (Satzung über die Herstellung und Ablösung von Garagen und Stellplätzen, Garagen- und Stellplatzsatzung – GaStS) to further regulate parking slots based on the GaStellV. Thus, these regulations are interrelated in a hierarchical, nested structure.

This study is framed within these legal contexts and frameworks. The primary focus is on GaStellV, complemented by the GaStS as additional restrictions imposed by municipal authorities in response to local circumstances.

GaStellV consists of six sections: general provisions, building regulations, operating rules, building specifications and testing, necessary parking slots, and final provisions. The specific article of general provision, part of building regulations, and necessary parking slots are chosen as the test for automated regulation checking. Selection criteria are based on quantitative properties of the related objects, detailed as follows:

- General provision: Garage classification (§1 GaStellV)
- Building regulations:
    - Entrance and exit (§2 GaStellV)
    - Ramps (§3 GaStellV)
    - Parking slots and driveways (§4 GaStellV)
    - Clear height (§5 GaStellV)
- Necessary parking slots: Number of the parking slot (§20 GaStellV),

Note that Paragraph 7 in §4 Parking slots and driveways is not in our checking scope since fire safety is not the focus in this case.

The GaStS includes six articles, with Article 2 detailing local regulations for the number of parking slots and garages in Ingolstadt. Additionally, Article 3 introduces exemptions for parking slots, such as exemptions for attic apartments.

Overall, the focus is on these selected articles from GaStellV. For calculating the required number of parking slots, the process involves identifying the building functions as defined in GaStS and filling in any gaps for missing functions using GaStellV.

3.1.2 Interpretation and Formalisation of Regulations

Once the relevant regulations for the checking process have been identified, further analysis and breakdown are required. This involves interpreting and standardising the regulations to convert them into a machine-readable format. The approach employed is based on Lee's methodology for establishing the Korean Building Act standards in 2016, which logically rule-based decomposes the regulations (Lee et al., 2016). Despite the diversity in themes and constraints across regulations, a general workflow can be applied to other regulations by identifying commonalities and conclusions from previously interpreted regulations.

Following an initial evaluation of the potential for standard formalisation, making the regulations machine-readable is achieved through several defined processes. These processes define relevant objects, required attribute information, and their logical relationships and constraints. The steps are as follows, shown in Figure 3-1:

1. Manually interpret regulations, consulting local experts in case of ambiguities.
2. Break down the regulatory text into objects, attributes, functions, conditions, and constraints.
3. Generate a conceptual model of the legal requirements.

4. Transform regulations into calculable form based on the conditions and constraints determined in Step 2.

The fourth step will be implemented during the development of the regulation-checking workflow in Section 5.3.

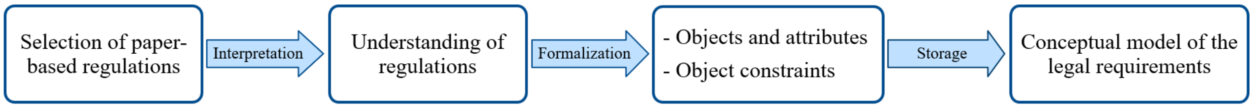


Figure 3-1 Workflow of the interpretation and formalisation of regulations

Figure 3-2 outlines a schematic representation for the formalisation of the selected articles, including the objects, attributes, and constraints. The whole table is shown in the Appendix B.1. Additionally, it introduces two crucial definitions: A **parking slot** is designated for parking motor vehicles within a garage. The **usable area** of a parking lot is the sum of all areas of the parking slots and the circulation areas. Furthermore, the requirements extend beyond defining the parking slot to include restrictions on ancillary elements, such as driveways, footpaths, and entrances.

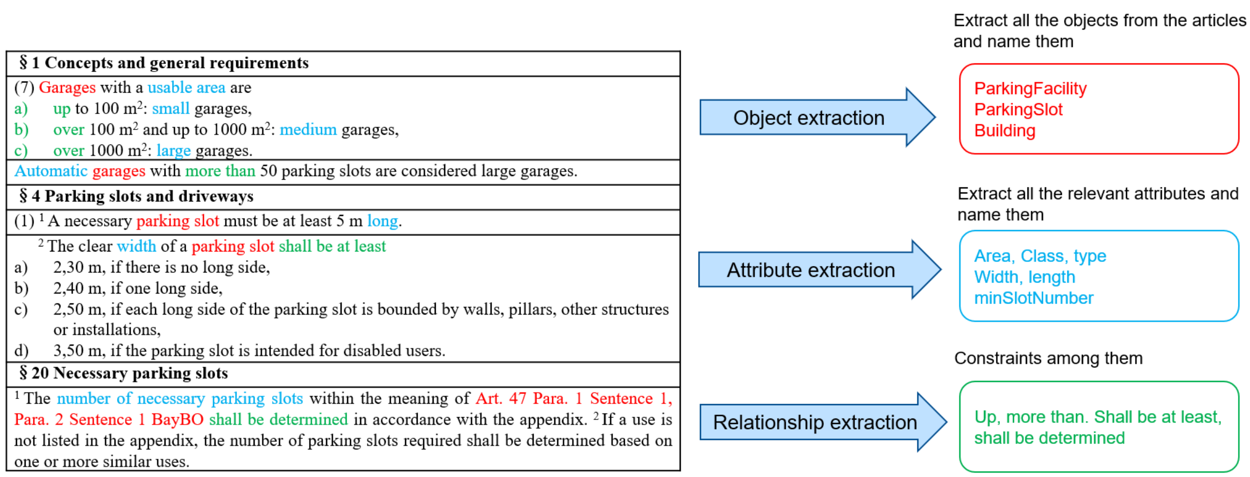


Figure 3-2 Schematic explanation of the regulation formalisation: red represents objects, blue represents the attribute, and green represents constraints

3.1.3 The Conceptual Model of the Legal Requirement

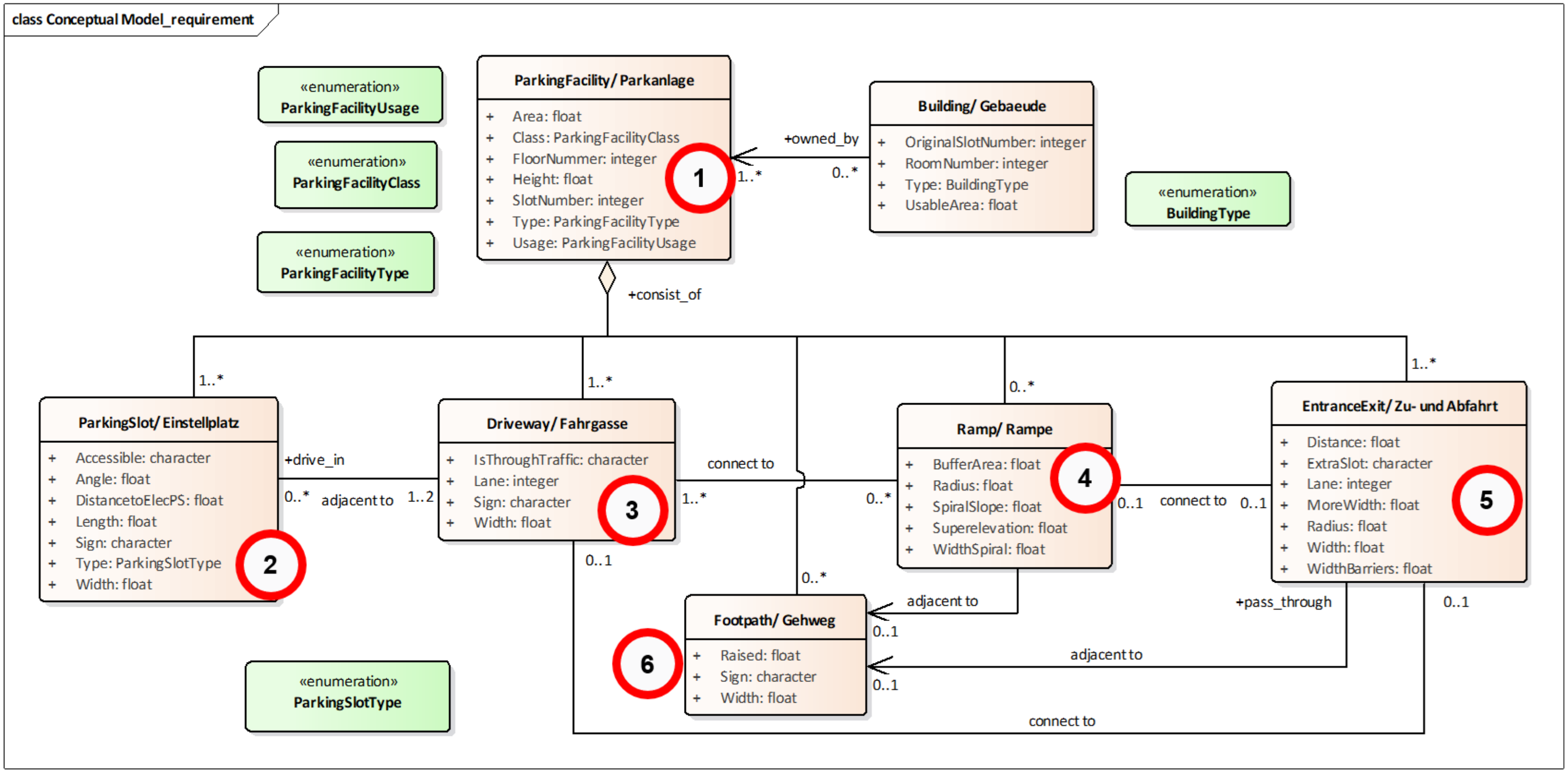
Based on the formalisation results from the previous sub-section, a UML diagram can be created. Objects and corresponding attributes related to the regulations are stored, forming a conceptual model, as shown in Figure 3-3. A total of seven objects are defined. As a formal, software-independent description, this conceptual model serves as an intermediate stage before embedding into CityGML 3.0, aimed at making the mapping of regulations to the data model more efficient and logical. All the objects defined in this model are only intended for general use.

Parking facility in this context refers to comprehensive structures or areas containing multiple parking slots. At this stage, its exact location is not defined or specified. It is composed of five elements (aggregation relation "consist_of"). The purpose here is not to strictly define the precise components of the parking facility but to identify and name the elements based on the subjects of the selected articles. Therefore, *ParkingSlot*, *Driveway*, *Ramp*, *EntranceExit*, and *Footpath* are created. Driveways, ramps, and Entrance/Exit are defined separately here, although they are essentially all driving surfaces (illustrated in Figure 3-3[b]). Thus, their relationships are interconnected (association relation “connect_to”).

Parking slots are the core part of the assessment, including necessary semantic and geometric attributes. They are always adjacent to the driveways, allowing vehicles to drive in and out (association relation “adjacent to” with role “drive_in”). The *Footpath* class represents all traffic surfaces designated for pedestrian use. Based on actual conditions and regulatory requirements, it is adjacent to certain driveway objects (i.e., *Ramp* and *EntranceExit*). In this conceptual model, ground markings are not modelled separately but are embedded as attributes in related objects. In other words, it is assumed that different traffic surfaces are directly adjacent to each other.

Building object is also introduced to store the number of parking slots required for specific buildings based on regulations. To calculate this number, certain building attributes need to be clarified, such as the building’s function and usable area. Finally, some parking facilities may belong to a specific building, so their relationships need to be established (association relation "owned_by").

[a]



[b]

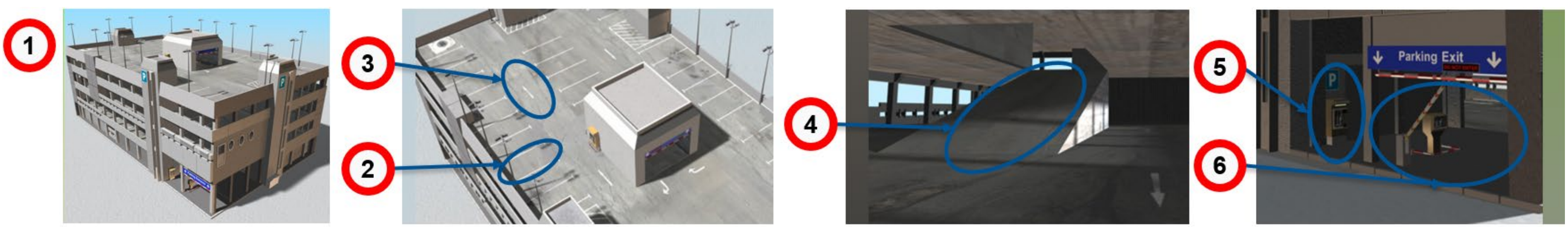


*Figure 3-3 [a] The conceptual model developed from the selected regulation; [b] The explanation of the parking objects**

*The description representation is from: https://www.turbosquid.com/3d-models/3ds-max-parking-garage-architecture-building/526181

3.2 Technical Requirement

The presentation of parking slots and related regulations also requires specific technical specifications. This chapter outlines the technical requirements for this automated parking facility assessment use case and explains how CityGML 3.0 meets all these requirements.

3D models are crucial tools for assessing the compliance of buildings and infrastructure with specific regulations at the city level. In this context, automated regulatory checking for parking facilities requires an integrated model of regulations and entities, where physical objects are assigned specific attributes to complete the checks. More details, for regulatory checks, 3D models must have the following key features:

- Geometric representation: The model must accurately depict the geometry and dimensions of parking slots and their ancillaries.
- Semantic representation: This use case requires semantic definition and classification of information according to regulatory requirements. The model must include semantic information relevant to the regulations, such as the type and size of parking slots. This involves mapping regulatory information and necessary model extensions. Additionally, it should include the function of code list definition.
- Spatial scope: In this study, the use case defines the necessary spatial scope to support its workflow. Regulatory checks need to be performed at different scales. The data model must support detailed representation, from the macro city level to the micro individual level. It is crucial to aggregate and disaggregate the model across various spatial scales.
- Topological relations: The model should represent the topological relationships between parking slots and other city elements. This includes the relationships not only among the objects of study themselves but also between different study objects and other city objects.
    - Correct topological relationships among parking slots and surrounding ancillary objects need to be established to properly assess whether all regulation-related objects meet the legal requirements.
    - The regulations require linking the parking facilities and other city objects, such as street spaces and buildings.
- Visualisation: Visualisation plays a role in the most critical features of 3D city models. Accurately depicting the real world and facilitating interactive visualisation are essential for information extraction and comprehension. By visualising 3D models, potential issues can be identified more intuitively, enhancing the accuracy and efficiency of decision-making. This investigation also includes the forms of expression and the level of detail of the data model.
- Dynamic management: Regulatory checks may involve dynamic information, such as historical or future scenarios, the regulation versions, and real-time sensors. While this study mainly focuses on regulatory assessment, it should have the potential to link with this dynamic information to extend this application further. On the other hand, that means it should incorporate the ability to connect with external sensors. This enhances visualisation with additional hotspot data, offering greater strategic insights.

CityGML 3.0, as a mature semantic 3D city model, has been reviewed in Section 2.1. Its detailed hierarchical structure, rich semantic information, strong extensibility, and new Versioning and Dynamizer modules provide forceful technical support for regulatory checks. The Core module of CityGML 3.0 define all geometric representations and four Levels of Detail (LOD 0-3) (Kolbe et al., 2021). It addresses the generic requirements for representing legal and physical data in parking areas, such as geometric representations and coordinate systems. Further subset feature types will automatically inherit the representations and LODs. For this use case, the focus is on the Transportation module, which provides semantic and geometric information about relevant objects in parking facilities.

The detailed semantic model built into CityGML is also the core to implementing regulatory checks. Each class has detailed and accurate definitions, and its attributes can describe the details and properties of the city objects. The multiple extension mechanisms of CityGML 3.0 allow the creation of new objects and attributes that are not present in the CityGML Conceptual Model. This makes it possible to add specific regulatory information. In CityGML 3.0, some new extension modules have been added, such as Versioning and Dynamizer modules, allowing for dynamic management of the applications. Therefore, this data model can provide generic technical requirements for the parking facility assessment.

4. CityGML3.0 Extension Development

After determining the legal requirements, the regulatory conceptual model needs to be mapped into CityGML 3.0, as mentioned in Section 1.3.2. For any missing components, the data model needs to be extended. Ultimately, all objects from the regulation conceptual model are stored in the extended data model exchanged in CityGML 3.0, including their attributes and geometry. This chapter represents the second part of the Methodology, focusing on mapping and extending into CityGML 3.0.

4.1 Description of the Mapping and Extending

The objective here is to achieve the minimum extension without losing information during the mapping and extension process, as the ultimate aim is to perform regulatory checks. The mapped data model serves as a further intermediate step, forming the basis for creating a workflow. To achieve this goal, I followed the mapping and extension principle based on the master thesis of Bachart (Bachert, 2023).

General mapping principle:
For this use case, it is important to integrate as much as possible when mapping the regulation conceptual model to CityGML 3.0. This means that the first round of mapping of objects and attributes should utilize the CityGML Conceptual Model. Subsequently, the objects and attributes that do not have a corresponding mapping are extended based on the Generics module. The advantage of this strategy is that it minimizes the extensions to CityGML 3.0 and avoids redundancy and the need to create a new schema compared to the ADE mechanism.

When it comes to each object and attribute, the choice of the mapping class and attribute in CityGML 3.0 needs to be evaluated from multiple perspectives. The first consideration is semantic fit. According to the definition of the regulation and CityGML 3.0, a regulatory object may fit several CityGML classes. In this context, it is essential to determine which CityGML class can provide more information and flexibility. Additionally, whether a single attribute can provide information for multiple regulatory objects should be considered as well. For instance, a series of attributes related to regulatory check results can be shared by different features within the same class.

The following sub-sections detail the mapping processes and explain the rationale behind the selection for each object.

4.1.1 Parking Facility Mapping

Parking facilities in this study refer to the comprehensive structures or areas composed of multiple parking slots. They serve as urban infrastructure designed to provide temporary parking slots for vehicles. These facilities can be open-air, underground, or multi-story structures. Unlike ordinary collections of parking slots, these facilities generally include designated entrances and exits, as well as organized layouts and management. The mapping process for *ParkingFacility* involves the reallocation of classes and attributes. The mapping details of this object can be found in Table 4-1, which extends the existing modules through *Generic* attributes.

For the *ParkingFacility* class, there are multiple mapping possibilities. It can be mapped to *AbstractTransportationSpace* or its subclasses. However, as an abstract class, it cannot directly carry the specific attributes of parking facilities. It cannot provide insufficient information either. To avoid creating new objects and redundancy, it is advisable to explore more specific classes.

CityGML 3.0 has created several new classes to store different functional traffic objects onto the semantic concepts of spaces and space boundaries. According to this logic, *ParkingFacility* can be stored in *TrafficArea*, *TrafficSpace, or Square*, the class descriptions are shown in Table 4-2. From the definition, while *Square* can represent large parking lots, it mainly describes open-air parking lots. As a «TopLevelFeatureType», Mapping *ParkingFacility* to *Square* can cause ambiguity and additional modelling work. For example, it requires separate considering the physical location of the parking facility, whether indoors or outdoors. *TrafficSpace* and *TrafficArea* can more flexibly represent various forms of parking facilities without being constrained by their location.

TrafficSpace and *TrafficArea* have an association relationship, with *TrfficArea* bounding *Trafficspace* at the bottom. Mapping parking slots to TrafficArea is a better choice in the sub-section 4.1.3. If parking facilities are also mapped to *TrafficArea*, new "relatedTo" relationship connections need to be introduced to link parking slots and facilities, potentially causing redundancy and confusion in overlapping areas. Mapping *ParkingFacility* to *TrafficSpace* is a more logical option, as it not only shows the traffic spaces of the parking facilities but also has an optional clearance space to represent the clear height. Additionally, the aggregation hierarchy between *TrafficSpace* and *TrafficArea* clarifies the hierarchical relationship between parking slots, their ancillaries, and parking facilities.

Table 4-1 The mapping details from the "ParkingFacility" to CityGML 3.0. The bold text is the extended part.

Parking regulations CM	CityGML CM and extending	
Attribute	Class	Attribute
ParkingFacility		
Class	*TrafficSpace*	***classOfPF***
Form	*TrafficSpace*	***formOfPF***
Usage	*TrafficSpace*	***usageOfPF***
Type	*TrafficSpace*	***typeOfPF***
Area	*TrafficSpace*	***usableAreaOfPF***
FloorNumber	*AbstractBuilding*	*StoreyAboveGround* *StoreyBelowGround*
originalNumberOfPS	*Building*	***originalNumberOfPS***
Height	*ClearanceSpace*	***clearHeight***
	AbstractBuilding	*storeyHeightsAboveGround* *storeyHeightsBelowGround*

Table 4-2 The descriptions of considered classes (Kolbe et al., 2021)

Class	Description
Square «TopLevelFeatureType»	A *Square* is a transportation space for unrestricted movement for vehicles, bicycles and/or pedestrians. This includes plazas as well as large sealed surfaces such as parking lots.
TrafficArea «FeatureType»	A *TrafficArea* is the ground surface of a *TrafficSpace*. Traffic areas are the surfaces upon which traffic actually takes place.
TrafficSpace «FeatureType»	A *TrafficSpace* is a space in which traffic takes place. Traffic includes the movement of entities such as trains, vehicles, pedestrians, ships, or other transportation types.

For attribute mapping, since no new classes are created, the new attributes from *ParkingFacility* can be directly extended using the *Generics* module. It is important to emphasize that *SlotNumber* is mapped to *originalNumberOfPS* in the *Building* class. This attribute needs to be compared with the minimum number of parking slots required within an area or building, which is defined according to Article 20. It can be directly placed in the properties of the *Building*, separated from the parking facilities, since not all parking slots are located within parking facilities.

Here, based on the regulation conceptual model, a total of seven new attributes are added in CityGML 3.0. The meaning of each attribute is described in Table 4-3, the lists of enumeration is shown in Appendix B.4.

Table 4-3 The meaning of extended PackingFaciity attributes

Attribute	Meaning	Data type
TrafficSpace		
classOfPF	The category of size to which it belongs ranges from small to medium to large.	enumeration
formOfPF	The category of form to which it belongs can be open-air, garage, or multi-story.	enumeration
usageOfPF	The category of usage to which it belongs can be either public or private.	enumeration
typeOfPF	The category of type to which it belongs can be either automatic or traditional.	enumeration
usableAreaOfPF	The net usable area of the parking facility comprises all the parking slots and circulation areas.	real
Building		
originalNumberOfPS	The actual number of parking slots in a parking facility or a certain area	integer
ClearanceSpace		
clearHeight	The clear height in the traffic area. The unobstructed vertical height is available in the traffic area.	real

4.1.2 Building Mapping

During the assessment process, buildings also need to provide basic information, such as the usable area or the number of rooms, to determine the necessary parking slots. In the *Building* class, new *Generic* attributes are created. Specific mapping details can be found in Table 4-4, and the meanings of the extended attributes are described in Table 4-5.

It is important to note the mapping of the building function from GaStS and GaStellV to CityGML 3.0. Both of GaStS and GaStellV provide tables of building functions and their corresponding required parking slots (refer to Appendix A. 2 and A. 3 for details). Although the code list of attribute *Function* in *AbstractBuilding* in CityGML 3.0 covers some building functions, it does not fully include the building types specified in GeStS and GastellV.

After a thorough manual inspection and mapping, it was found that there are 69 building functions in GaStS and GaStellV, of which 49 can be mapped onto *Function* in *AbstractBuilding*, covering more than half. Therefore, a partial extension for the *Function* can be implemented to identify the remaining 20 building types. The complete table can be found in the Appendix B.2 and B.3.

Table 4-4 The mapping details from the "Building" to CityGML 3.0. The bold text is the extended part.

Parking regulations CM	CityGML CM and extending	
Attribute	Class	Attribute
Building		
Type	*AbstractBuilding*	*function*
RoomNumber	*Building*	***roomNumber***
SlotNumber	*Building*	***requiredNumberOfPS***
UsableArea	*Building*	***usableArea***

Table 4-5 The meaning of extended Building attributes

Attribute	Meaning	Data type
Building		
roomNumber	The total number of rooms within a building.	integer
requiredNumberOfPS	Minimum required number of parking slots	integer
usableArea	The usable area of the building as defined by DIN 277 Part 2 (Nutzfläche).	real

4.1.3 Parking Slot Mapping

As the core object of the regulation checking in this study, parking slots should be completely mapped to the Transportation Module in CityGML 3.0. A parking slot is defined as a specific space reserved for a single vehicle. Similar to *ParkingFacility*, *ParkingSlot* can be mapped to *AbstractTransportationSpace* or its subclass, such as a specific transportation object, *Square*. To avoid generating a new *Generic* object, the ideal mapping for parking slots is *TrafficArea*.

By definition, the *Square* refers more to spaces containing multiple traffic areas. As a whole object and top-level feature type, *Square* does not typically involve fine-grained descriptions of traffic areas. Using *Square* as a carrier for parking slots may lead to semantic obfuscation and loss.

TraffiArea, being the surface where transportation activities actually occur, is a better choice for serving as the carrier for *ParkingSlot*. The code list for the *TrafficArea* attribute *function* already includes a code "7" for parking slots. This allows for filtering parking slots and subsequently adding

Generic attributes. A total of six parking-related attributes are added to the *TrafficArea*. The mapping details and the meaning of each attribute are shown in Table 4-6 and Table 4-7.

Table 4-6 The mapping details from the "ParkingSlot" to CityGML 3.0. The bold text is the extended part.

Parking regulations CM	CityGML CM and extending	
Attribute	Class	Attribute
ParkingSlot		
Accessible	*TrafficArea*	***isAccessible***
Angle	*TrafficArea*	***angleOfParking***
DistancetoElecPS	*TrafficArea*	***isInFrontOfElecPlattform***
Sign	*Marking*	*class*
Width	*TrafficArea*	***widthOfParkingSlot***
Length	*TrafficArea*	***lengthOfParkingSlot***
Type	*TrafficArea*	***typeOfParkingSlot***

Table 4-7 The meaning of extended TrafficArea attributes based on parking slots

Attribute	Meaning	Data type
TrafficArea		
isAccessible	Indicates whether the parking slot is accessible for disabled people.	string
angleOfParking	The angle between the parking slots and driveways	real
isInFrontOfElecPlattform	Denotes whether the parking slots are located in front of power-operated lifting platforms.	string
widthOfParkingSlot	The parking slot width	real
lengthOfParkingSlot	The parking slot length	real
typeOfParkingSlot	The type category of the parking slot, which can include classic, horizontally movable platforms, sloping power-operated lifting platforms, or standard power-operated lifting platforms	enumeration

4.1.4 Ancillary Objects Mapping

The ancillary objects refer to the city objects in the circulation areas and the supporting facilities around the parking slots. In this study, the regulation conceptual model includes four ancillary objects, namely *Ramp*, *Footpath*, *EntranceExit,* and *Driveway*.

Similar to the mapping of *ParkingSlot*, it also has non-unique mapping possibilities. There are some straightforward literal mappings. *Footpath* could be mapped to *Track*. *Ramp*, *EntranceExit,* and *Driveway* are essentially driveways in different locations. They could be merged and mapped to *Road*. However, a closer examination of their definitions reveals semantic mismatches. *Road* and *Track* as top-level feature types are primarily used to describe and define specific macro-objects, such as urban road networks or park paths. They serve more as parent features with the ability to be further subdivided and decomposed.

When describing necessary vehicle passage spaces and footpaths around parking slots, *TrafficArea* offers more flexibility, allowing for finer definitions. For each transportation object, traffic areas can be provided. Traffic areas are the important elements for traffic usage. It is possible to model and assign attributes for different types of traffic areas. These four objects are finally mapped to the *TrafficArea*. Similar to parking slots, different types of traffic surfaces can be distinguished using attribute *function*, such as driving_lane with “1” and footpath with “2”.

For the mapping and storing attributes, some complex attributes in the regulatory conceptual model are decomposed into multiple attributes in CityGML 3.0, making them more universally applicable. For example, regulations specify the width of ramps: "The width of the driveway on these ramps must be at least 2.75 m and at least 3.50 m in the case of spiral ramp sections." The attribute *WidthSpiral* is created in the regulatory conceptual model. However, in practice, this involves three pieces of information: slope, width, and driveway radius. By separating these attributes, they can be used in any combination of *TrafficArea* features with the function with "1" rather than being limited to ramp sections, avoiding redundancy of the attributes in the data model.

On the other hand, some attributes require multiple classes to complete the mapping. For example, the attribute *BarrierWidth* in the regulatory conceptual model represents the lane width in entrance areas, requiring not only *TrafficArea* attribute *widthOfDriveway* but also the identification of barrier areas in *CityFurniture*.

In the end, seven driveway-related attributes are added to *TrafficArea*, and the mapping details and the meaning of each attribute are shown in Table 4-8 and Table 4-9. For the footpath, two attributes are extended (see Table 4-10 and Table 4-11).

Table 4-8 The mapping details from the “Driveway”, “Ramp”, and “EntranceExit” to CityGML 3.0. The bold text is the extended part.

Parking regulations CM	CityGML CM and extending	
Attribute	*Class*	*Attribute*
Driveway		
IsThroughTraffic	TrafficArea	**isThroughTraffic**
Width	TrafficArea	**widthOfDriveway**
Lane	TrafficArea	**numberOfLanes**
Sign	Marking	class
Ramp		
BufferArea	TrafficArea	**lengthOfDriveway** **slopeOfDriveway**
WidthSpiral	TrafficArea	**widthOfDriveway** **radiusOfDriveway** **slopeOfDriveway**
Superelevation	TrafficArea	**superelevation**
SprialSlope	TrafficArea	**slopeOfDriveway** **radiusOfDriveway**
Radius	TrafficArea	**slopeOfDriveway**
Entrance/Exit		
Distance	TrafficArea	**lengthOfDriveway**
Radius	TrafficArea	**slopeOfDriveway**
Lane	TrafficArea	**numberOfLanes**
Width	TrafficArea	**widthOfDriveway**
WidthBarriers	CityFurniture TrafficArea	Function **widthOfDriveway**
ExtraSlot	TrafficArea	**widthOfParkingSlot**

Parking regulations CM	CityGML CM and extending	
Attribute	*Class*	*Attribute*
MoreWidth	TrafficArea	**lengthOfParkingSlot** **widthOfDriveway** **radiusOfDriveway**

Table 4-9 The meaning of extended TrafficArea attributes based on driveways

Attribute	Meaning	Data type
TrafficArea		
isThroughTraffic	Denotes whether the parking slots are located in front of power-operated lifting platforms.	string
lengthOfDriveway	The driveway length	real
widthOfDriveway	The driveway width	real
slopeOfDriveway	The driveway slope (Vertical Rise/Horizontal Distance)(%)	real
radiusOfDriveway	The radius of the driveway's curvature.	real
numberOfLanes	The number of lanes in the driveway	integer
superelevation	The inward tilt of the road surface on a spiral ramp.	real

Table 4-10 The mapping details from the "Footpath" to CityGML 3.0. The bold text is the extended part.

Parking regulations CM	CityGML CM and extending	
Attribute	Class	Attribute
Footpath		
Raised	*TrafficArea*	***footPathHeightsAboveGround***
Width	*TrafficArea*	***widthOfFootpath***
Sign	*Marking*	*class*
	CityFurniture	*function*

Table 4-11 The meaning of extended TrafficArea attributes based on footpath

Attribute	Meaning	Data type
TrafficArea		
footPathHeightsAboveGround	The footpath height above the roadway level.	real
widthOfFootpath	The footpath width	real

4.1.5 Regulation Extension

The GaStellV was selected as the parking standard regulations applicable for Bavaria. This study conducts regulatory checks based on these standards. These inspection results demonstrate the parking status in these areas and are one of the significant outcomes of this study. The data model should also record these results for reference and review. In other words, the articles, parameters, and inspection results are stored as *Generic* attributes.

Four fundamental *Generic* attributes are defined. The description of each attribute can be found in Table 4-12. Since *TrafficArea* is the most primary class in this study, most of the inspection results can be stored within it, allowing for a more targeted interpretation of each feature. These extensions show good potential applicability, including other articles within this regulation as well as other regulations. However, each regulation is governed by its own set of rules and constraints. Addressing these specific requirements may necessitate modifying the location of these attributes and revising the enumeration list for attribute *failureReason*. The enumeration designed for the attribute *failureReason* can be found in Appendix B.4, Table B- 6.

Table 4-12 The meaning of extended TrafficArea attributes based on regulation checking results

Attribute	Meaning	Data type
TrafficArea		
administrationArea	The applied location	string
articleNumber	Document the checking article number	string
assessmentResult	Document whether the result is in compliance.	string
failureReason	Document the uncompiled reasons; it is an enumeration	enumeration

4.2 The Extending CityGML 3.0

The conceptual model for the CityGML transportation extension has been formed. It is applicable for parking status assessment in the city of Ingolstadt, Bavaria, based on GaStellV. The UML diagram shows the framework of the extended data model in this study, as illustrated in Figure 4-1. Overall, the CityGML 3.0 Conceptual Model needs to be extended to meet the legal requirements for regulatory checks. Since all regulation objects can be mapped to CityGML 3.0 city objects, based on the "general mapping principle," we can extend the data only via the *Generic* attributes.

Attribute extensions were completed for four city objects: *TrafficArea*, *TrafficSpace*, *Building*, and *ClearanceSpace*. In the extended data model, all city objects must include the correct attributes, geometry types, and inheritance/aggregation structures. Specifically, every object and attribute involved in the mapping and extension process needs to comply with the standards set by the city model and follow the generic extension mechanism of CityGML 3.0. This ensures data interoperability and validity.

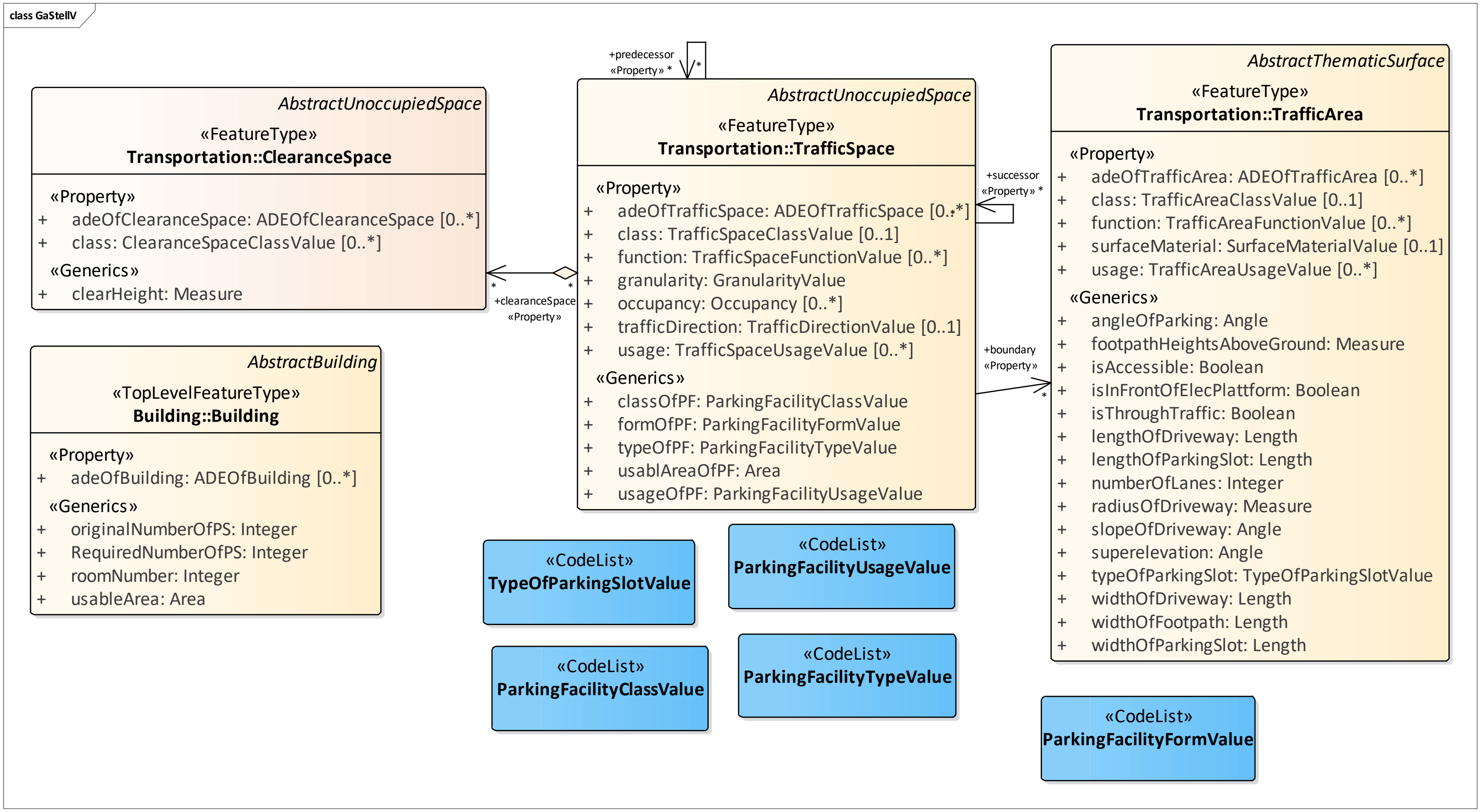


Figure 4-1 UML diagram of the CityGML extension

5. Implementation and Results

This chapter demonstrates the automation and feasibility of regulatory assessment through a use case study, transitioning from the conceptual model to implementation. As outlined in Section 1.3, this study is divided into three main parts. The theoretical frameworks for the first and second steps are detailed in Chapters 3 and 4 respectively. This chapter proceeds with the remaining implementation phase.

The implementation is achieved using workspaces in the FME Workbench. The case study area and datasets are introduced, followed by the conversion of source data to CityGML 3.0 with extensions. The development of the regulatory check workflow is then described. Finally, multiple outputs are provided. Figure 5-1 shows these various parts.

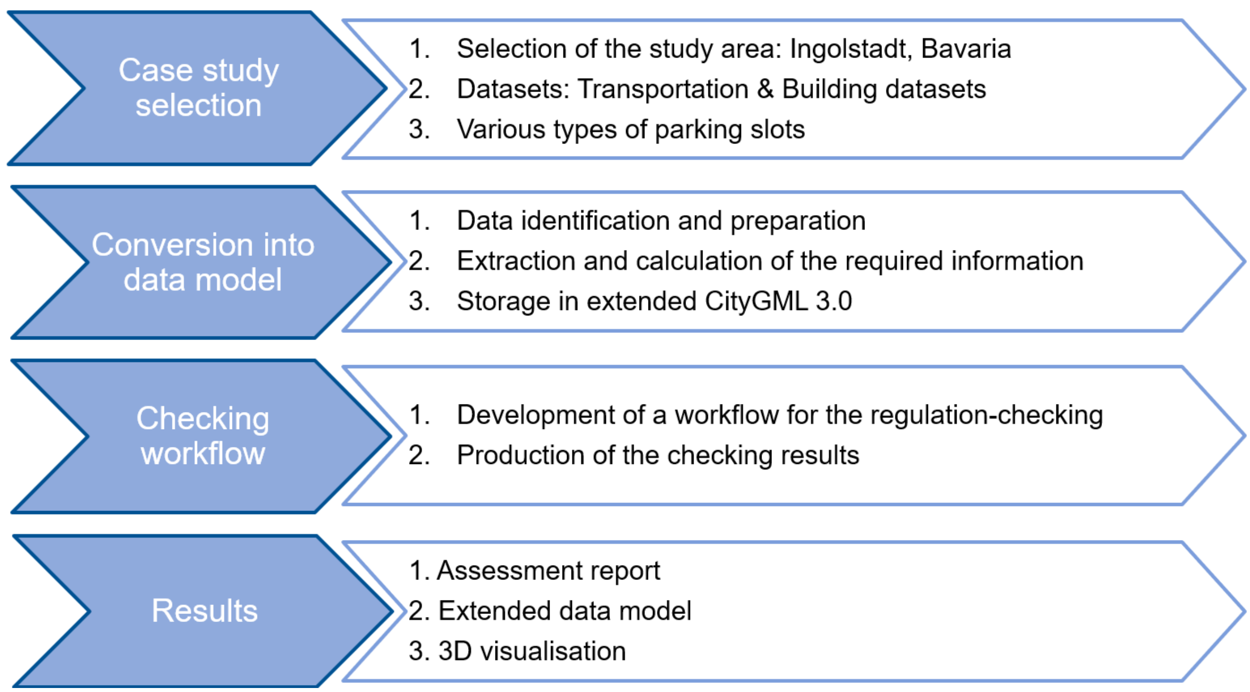


Figure 5-1 Steps of the implementation and results

5.1 Case Study

5.1.1 Study Area

This study selects the city of Ingolstadt in Bavaria as the case study location based on the geographical scope of the regulations and the availability of current data.

The study area includes the Ingolstadt section of Highway A9, parts of Federal Highway B13 within the city, the traffic circle in the northwest, and inner-city traffic, as shown in Figure 5-2. These areas contain various types of parking zones and their ancillary traffic surfaces, such as parking lots, private parking slots, and roadside parking, as illustrated in Figure 5-3.

Additionally, specific building areas have been included in this study, selected based on their proximity to the parking facilities. According to GaStellV regulations, these buildings are utilised to determine the necessary parking slots within the area.

Figure 5-2 Study area in Ingolstadt

5.1.2 Data

Two categories of datasets are used: transportation datasets from OpenDrive and building datasets from OpenData.

Transportation Data

The transportation datasets are sourced from the Funktions- und Verkehrs-**S**icherheit für **A**utomatisierte und **Ve**rnetzte Mobilität – **N**utzen für die Gesellschaft und **o**ekologische **W**irkung (SAVeNoW) project, using Ingolstadt as an example. These datasets are initially exchanged in OpenDrive format. OpenDrive is a data format using the Extensible Markup Language (XML) syntax with the file extension .xodr to describe the road network (ASAM OpenDRIVE®, n.d.). It primarily details the geometry of roads, lanes, and corresponding objects, as well as related features along the road. With the support of r:trån, it can be converted into a semantic 3D city model exchanged in CityGML 3.0. R:trån is an extensible toolkit designed for handling road space models (r:trån, 2020).

For this study, we directly use the converted CityGML 3.0 dataset, which includes the features of the Transportation module required for further analysis. Specifically, in class *TrafficArea*, different traffic surface types are recorded through *function* attribute, including parking slots (*function* = 7), drive lanes (*function* = 1), and footpaths (*function* = 2), as shown in Figure 5-3. However, the attributes required for the parking regulations are not yet included. Some data cleaning and conversion processes are necessary. There are some features missing the values in the *function* (see the <Null> in Figure 5-3). Since addressing these missing values is beyond the scope of this study, these features have been excluded from further analysis.

Figure 5-3 Types of parking zones and ancillary traffic surfaces in transportation dataset

Building Data

Building datasets are sourced from OpenData, a free geographic data provided by the Bavarian State Office for Surveying and Mapping (OpenData, 2024). These datasets are exchanged in CityGML 1.0, which includes building information at LoD2. LoD2 data provides detailed building geometries, including roof shapes and additional semantic information, which comply with ALKIS (Amtliches Liegenschaftskataster-Informationssystem, Official Land Registry Information System) standards. These datasets record basic information about buildings in Bavaria, including attributes such as building function, height, number of storeys, and floor area. For this study, only buildings near parking facilities are selected, as shown in Figure 5-4. The building data includes buildings with various functions.

For detailed regulatory checks on parking, these building attributes are essential for determining the required number of parking slots. However, some buildings in the dataset lack information on the number of storeys. To meet the requirements of this use case, the number of floors is approximated using the known building height, thereby continuing the assessment process.

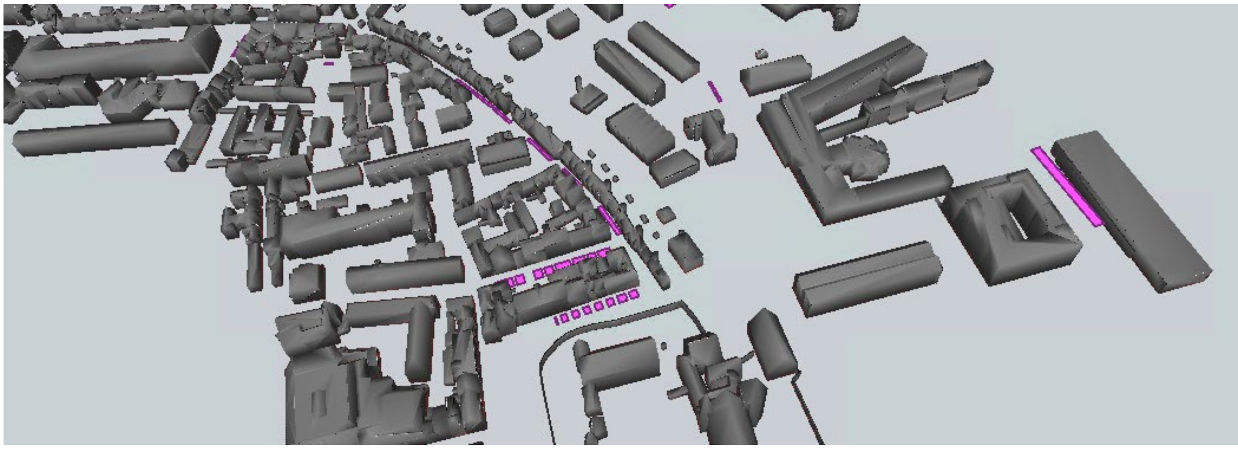


Figure 5-4 Partial view of selected building datasets

5.2 Conversion Process of Source Datasets into CityGML

The selected use case in this study aims to conduct the regulatory assessment of existing parking facilities and parking slots. Therefore, the source datasets need to be converted to the extended data model, including all the attributes required for the checks. In this section, the conversion of the transportation and building datasets to extended CityGML 3.0 is introduced separately. The entire data conversion and attribute supplementation are generated through the FME Workbench.

5.2.1 Transportation Dataset

The transportation datasets are exchanged in CityGML 3.0 converted from OpenDrive using r:trån. Some necessary attributes for regulatory assessments are initially absent and must be added. These missing attributes are supplemented using FME Workbench. The simplified schematic overview is shown in Figure 5-5, comprising six main steps.

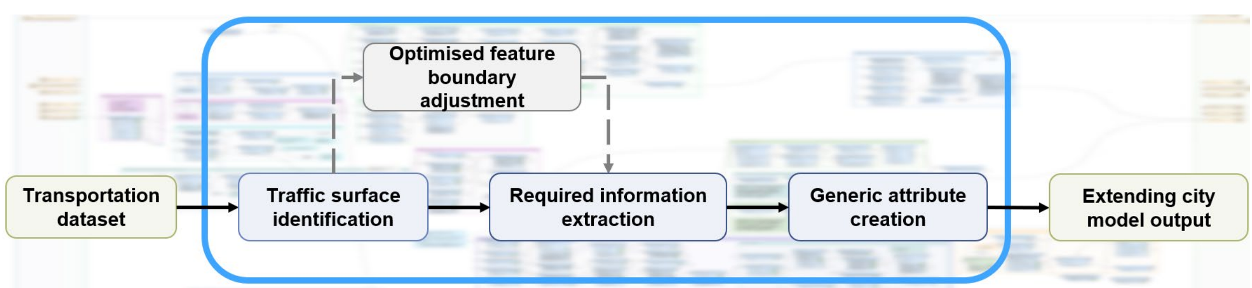


Figure 5-5 Schematic overview of the conversion process of transportation dataset in FME workbench, the green is the in-/output, the steps within the blue boxes are the main process, and the grey is an optional operation

Transformers in FME are utilised to convert and enrich the case dataset, i.e., the process depicted within the blue box in Figure 5-5. In this study, there are three main handling scenarios regarding processing the input data, as illustrated in Figure 5-6:

1. **Direct connection to the Writer**: Feature types that contain the complete attributes do not require processing and can be directly connected to the Writer, such as the class *Marking*.

2. **Attribute enrichment through transformations**: Some features are missing specific attributes, which can be obtained through a series of transformers. These transformers are inserted to process specific attributes before outputting the features to the Writer. The added attributes depend on the values of the *function* attribute. An AttributeFilter transformer separates features with different functions before further calculating and assigning the attributes. In this case, most transformers optimise and enrich geometric parameters and then extract their properties. A series of geometric transformers can be applied for initial conversion, followed by calculations using the ExpressionEvaluator transformer. Finally, an AttributeCreator and/or an AttributeManager transformer can be used to manipulate and store these new attributes. Sometimes, geometric optimisation requires other feature types for assistance (see complementary Feature X in Figure 5-6). This can be achieved using a SpatialFilter transformer to filter relevant candidate features, followed by optimisation using other transformers such as the AreaOnAreaOverlayer transformer. This process facilitates further attribute processing and assignment. Since *Generic* attributes are the list attributes, the AttributeCreator is preferable for creating these attributes with properties or relations with a 0..* multiplicity.

3. **Supplementation from other feature types**: In some cases, it is necessary to use one feature type to supplement the information of another (see complementary Feature Y in Figure 5-6). This process begins with SpatialFilter to identify relevant candidate features. Following this, additional transformers are applied for further attribute processing.

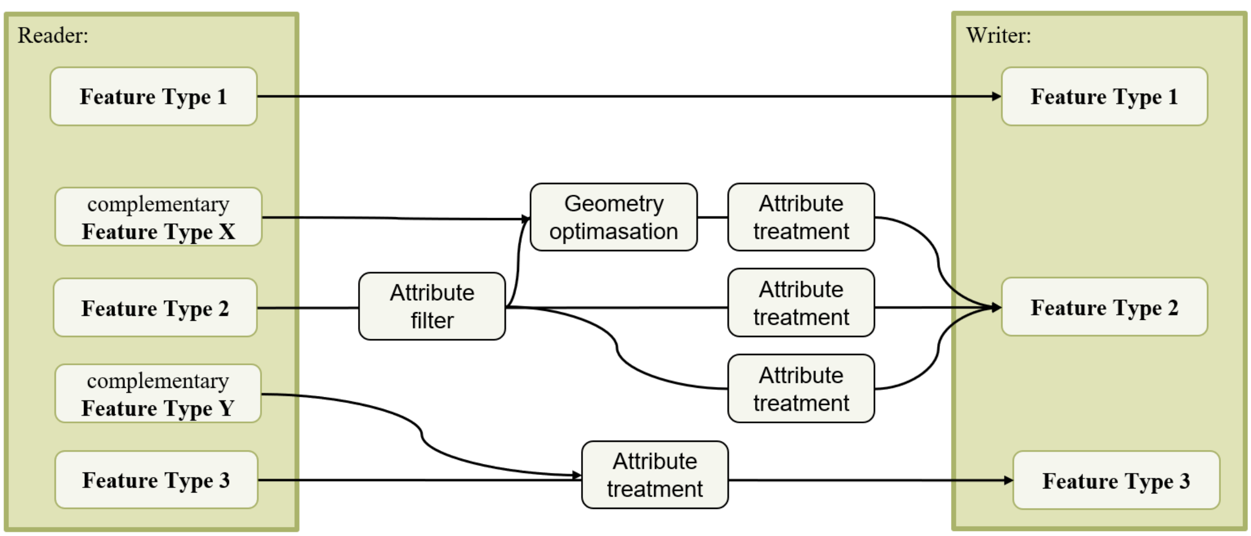


Figure 5-6 An illustrative diagram of the workflow for converting the use case dataset in the FME Workbench. Features 1, 2, and 3 represent different data processing scenarios

Data input and output

A new workspace is created in FME Workbench, with a generic GML Reader and Writer inserted. By utilising the corresponding CityGML 3.0 XSD schema file, the dataset can be defined according to the CityGML 3.0 standard, since the current CityGML Reader and Writer do not support CityGML 3.0. These components are used to handle the dataset subsequently, serving as interfaces for importing and exporting the data. This study primarily targets the *TrafficArea* class. Other related feature types are also incorporated to maintain the structural integrity and inheritance hierarchy of the CityGML city model, ensuring the completeness and validity of the data model.

Traffic surface identification

For the conversion of the *TrafficArea* class, we need to create attributes and assign corresponding values based on the mapping and extension details outlined in Chapter 4. The first step is to extract the traffic surfaces within *TrafficArea* sequentially via *function* attribute, such as parking slots with value "7", drive lanes with value "1", and footpaths with value "2" (as shown in Figure 5-7). In this step, the features with missing values of *function* attribute in the source dataset (see Figure 5-3) are omitted. Upon examining the data, it was found that the required attributes for the regulatory checks are not directly available. To derive their geometric dimensions, the datasets need to be cleaned and processed. Because many geometric transformers compress 3D features into 2D or operate based on 2D geometry, resulting in the loss of the Z coordinate, it is necessary to record the Z values of the features using ElevationExtractor transformer before starting. This allows them to be reconverted into 3D features before exporting to the Writer.

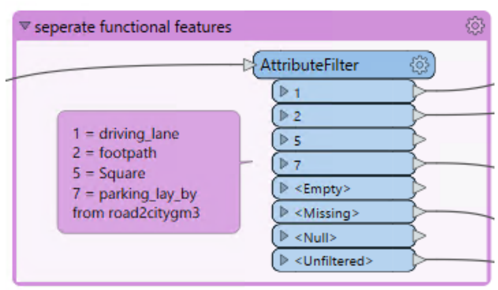


Figure 5-7 Separating different traffic surface of TrafficArea

Optimised feature boundary adjustment

For parking slots, each feature does not correspond one-to-one with a real single parking slot and is irregularly distributed. In other words, some features represent several parking slots, while others represent only a part of a parking slot. Therefore, step-by-step optimisation is needed to obtain the corresponding attribute values, i.e., the grey box in Figure 5-5.

First of all, the features need to be dissolved. Since there is no clear correspondence between the features and the actual parking slots, it is logical to dissolve all adjacent features and then redefine the parking slots. In this step, the Bufferer_2 transformer is used to slightly enlarge each feature by 0.05 m. The purpose of this is to allow adjacent features to overlap. Subsequently, the Dissolver transformer is employed to merge the overlapping features into larger, single features. The progress is shown in Figure 5-8.

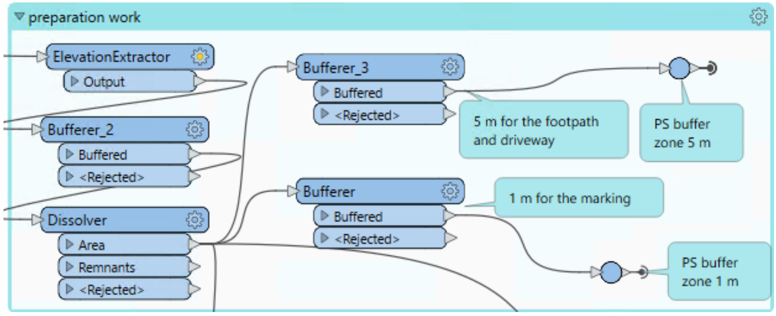
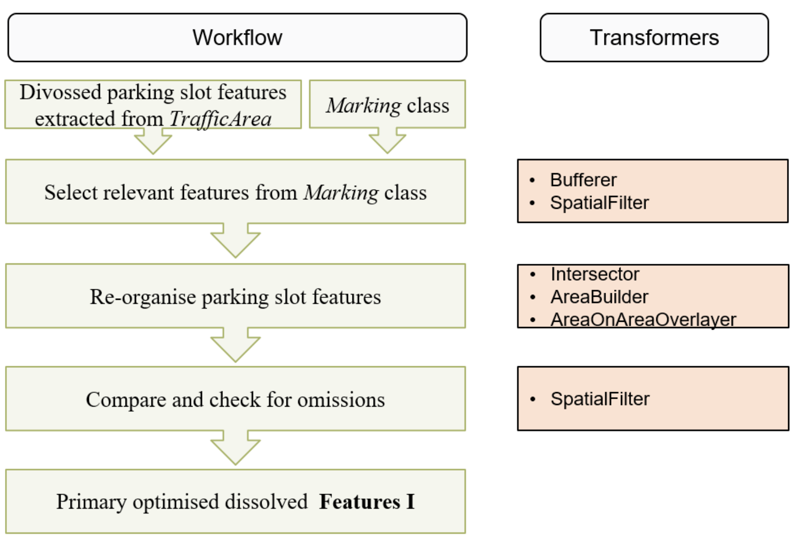


Figure 5-8 Dissolving the parking slot features for further transformation

Before further segmenting features to single parking slots, the *Marking* class, referred to as complementary Feature Type X in Figure 5-6, can be introduced to optimise the features initially. The *Marking* class is represented by lod2MutiSurface geometry to capture the visible road markings within the traffic area. To ensure that the parking slot features intersect with all relevant features in the *Marking* class, a 1-m buffer is applied (see Figure 5-8).

Next, the SpatialFilter is used to capture all these *Marking* features within the 1-m buffer zone. Through the Intersector and AreaBuilder transformers, the dissolved features from the previous step are split at the intersections of the two classes, creating topologically correct polygons. Subsequently, the AreaOnAreaOverlayer transformer is applied to handle the overlapping parts between the dissolved features and the Marking features, creating new polygons and removing redundant areas. This process ensures that the road markings effectively split the dissolved features. Finally, the SpatialFilter is employed to compare the results with the dissolved parking slot features, checking for any omissions. Figure 5-9, [a] illustrates this process in FME, while Figure 5-9, [b] provides a step-by-step geometric optimisation diagram.

[a]



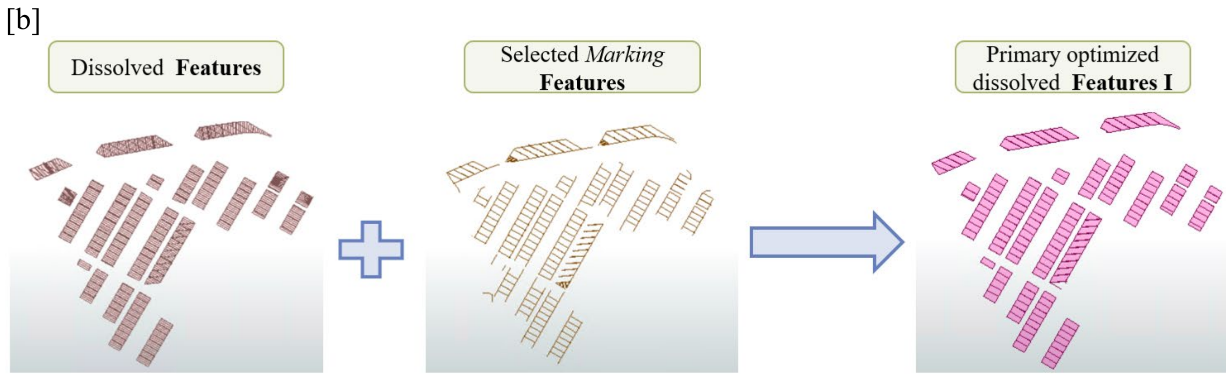


Figure 5-9 Primary optimisation of the parking slot features based on Marking class: [a] shows the workflow in FME, and [b] provides an illustrative example.

However, upon inspecting the spatial display of the dataset, some issues with the geometries optimised by the *Marking* class become apparent. For instance, the *Marking* class does not completely segment every dissolved parking slot feature, leaving some small connecting pathways, as shown in Figure 5-10, [b]. This requires a secondary manual adjustment. Figure 5-10, [a] illustrates a simplified manual adjustment process in the FME Workbench. The main idea is to extend the centrelines of the features in the *Marking* class to fully segment the parking slot features, since the features in the Marking class are elongated and oriented along the direction of the parking slots.

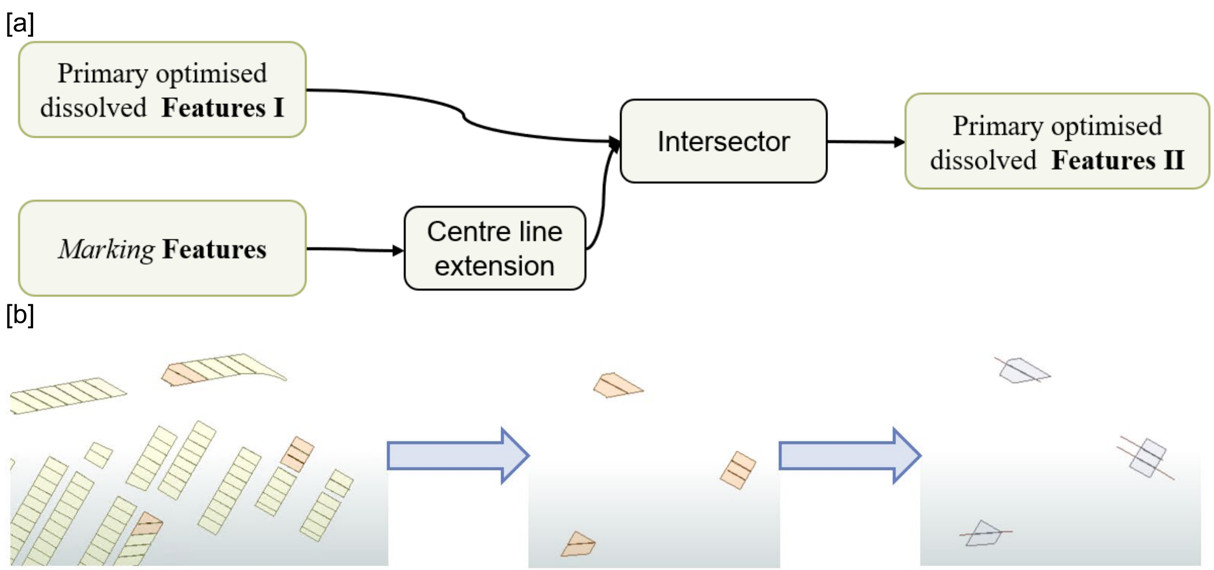


Figure 5-10 Manual adjustment process for Primary optimised dissolved Features I: [a] shows the simplified workflow in FME, and [b] provides an illustrative example. The orange polygons are the features that are not segmented correctly, while the grey polygons represent the optimised features.

After the primary optimisation by the *Marking* class, the parking slot features include some overly narrow and small polygons, as well as some complex polygons that are neither square nor rectangular. This requires further geometric optimisation of the features. The former, isolated, narrow and small polygons do not meet the objective criteria for parking slot assessment, such as a minimum area of less than 2 m^2 and width of less than 1 m, and should be directly removed. For the complex polygons, case-by-case analysis is required to correctly segment individual parking slot features. To separate them, the

TestFilter transformer is utilised to classify and process these features into 1. single parking slot features, 2. features containing multiple parking slots, and 3. narrow features.

The first part is the secondary optimisation of single parking slot features. The challenge is to identify an approximate rectangle within each feature to extract its geometric parameters, specifically the length and width of the parking slot. Because they are not all perfect rectangles, some have narrow protrusions or are approximate parallelograms, as shown in Figure 5-11. The parking slot features exchanged in CityGML 3.0 often contain numerous vertices and spikes, which complicates the process in FME Workbench due to its requirement for higher operational precision. In FME Workbench, there is no straightforward method to directly identify the largest internal rectangle within a polygon (Ideas in FME Community, 2019). In this context, only obtaining an approximate result is both sufficient and efficient. For example, in approximately parallelogram-shaped parking slot features, the two triangular sections can be removed to approximate a rectangle.

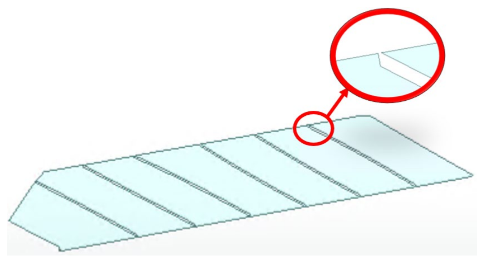
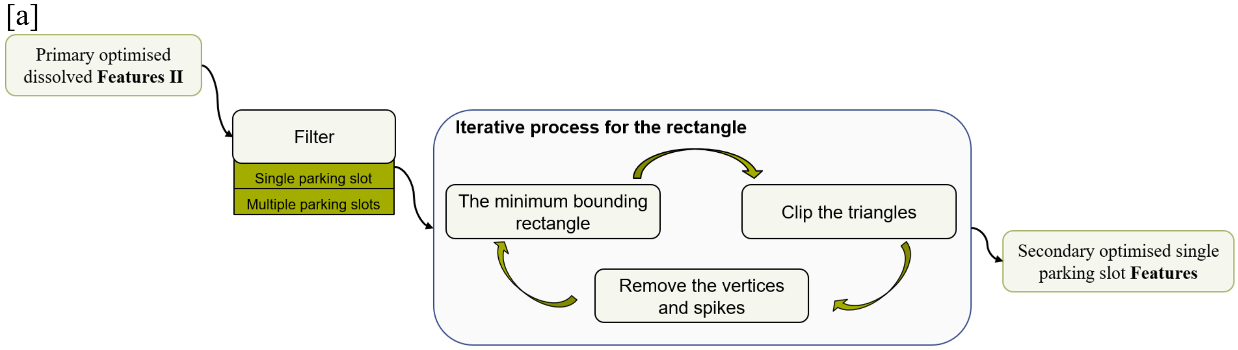


Figure 5-11 An example of the geometry representation for the parking slot feature

To address this, a general approach is provided. The BoundingBoxReplacer transformer is used to enclose the minimum bounding rectangle. Then, the bounding rectangle clips with the original shape via the Clipper transformer. This step produces two separate triangles. These triangles are then de-aggregated through the Deaggragator transformer, and a separate bounding box is created for each through BoundingBoxReplacer. The original features are clipped using these triangular bounding boxes to obtain the targeted approximate rectangle. However, this process can produce some narrow polygons since these shapes are imperfect parallelograms and may have non-parallel opposite sides and excessive vertices. Therefore, it is necessary to clean up the triangles by removing vertices and spikes using Generalizer and SpikeRemover transformers to achieve the desired result. Manual inspections are required at times. This iterative process ensures all single parking slots are converted to an approximate rectangle. Figure 5-12 shows the simplified process and an illustrative example.

[a]



[b]

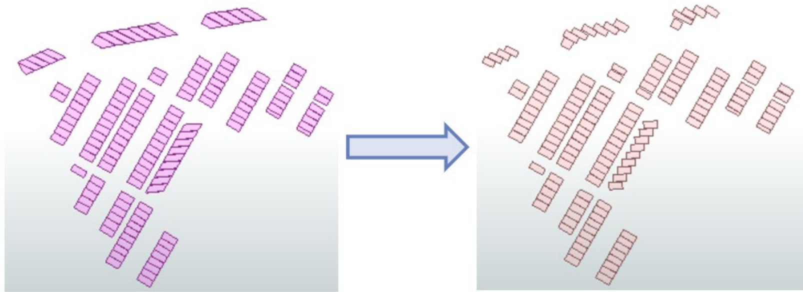


Figure 5-12 Secondary optimisation process for single parking slot features: [a] shows the simplified workflow in FME, and [b] provides an illustrative example.

In the second part, the features of multiple parking slots, which are relatively regular in shape, are addressed. The challenge lies in distinguishing how the parking slots are connected—whether by the long or short side—and handling other curved features.

First, the BoundingBoxReplacer determines their length and width, and then the feature aspect ratio is further calculated using the ExpressionEvaluator. Curved features will have a larger width than usual. For parking slots connected by the long side, the characteristics include a relatively large aspect ratio, and the feature width corresponds to the length of the parking slot, which is typically not less than 4.5 m. The 5-m standard width is not used, considering some parking slots do not meet the regulations. For parking slots connected by the short side, the characteristics include a relatively small aspect ratio, and the feature width corresponds to the width of the parking slot, which is typically not greater than 3 m. This filtering mechanism allows for their separation through the TestFilter.

Manual observation reveals that all roadside parking slots are connected by the short side for curved parking features. In this study, the centreline of these features can be extracted, and the Chopper transformer can segment the centreline into approximately 5-m sections. Then, the centre points of each segment are extracted using the CenterPointExtractor transformer. The VoronoiDiagrammer and Clipper divide the segments into individual parking slots. For rectangular parking features, the number of parking slots that can fit within each feature can be determined using a function in ExpressionEvaluator. The Tiler transformers evenly divide each feature according to the number of parking slots. The simplified schematic explanation is shown in Figure 5-13.

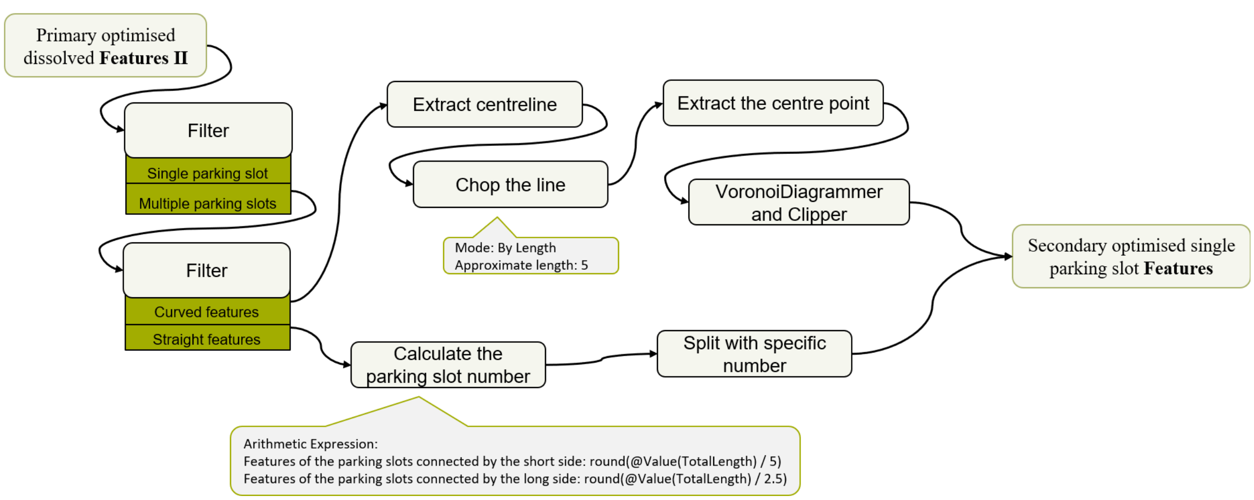


Figure 5-13 The simplified workflow for identifying the single parking slots from the features contained the multiple parking slots in FME

Required information extraction and Generic attribute creation

For the driveways and footpaths, extracting their width is crucial. Here, one approach is to directly maintain the features of the source dataset. The area of each feature is extracted using AreaCalculator transformer, followed by calculating the perimeter of the feature. The length and width of these features are calculated approximately according to the following equations with the ExpressionEvaluator.

$$length = \frac{perimeter}{4} + \sqrt{\left(\frac{perimeter}{4}\right)^2 - area}$$

$$width = \frac{perimeter}{4} - \sqrt{\left(\frac{perimeter}{4}\right)^2 - area}$$

This method is more effective for larger and narrower features. The feature "width" is the width of the driveways and footpaths. However, for smaller features, the feature "width" may capture the distance along the lane direction rather than perpendicular, which means it measures the road section "length". The feature "length" is the actual road section "Width" which is needed. In such cases, it is convenient to combine the TestFilter and AttributeManager to identify these features and transform the feature "length" into the "width" of the driveways and footpath. The specific workflow is presented in Figure 5-14.

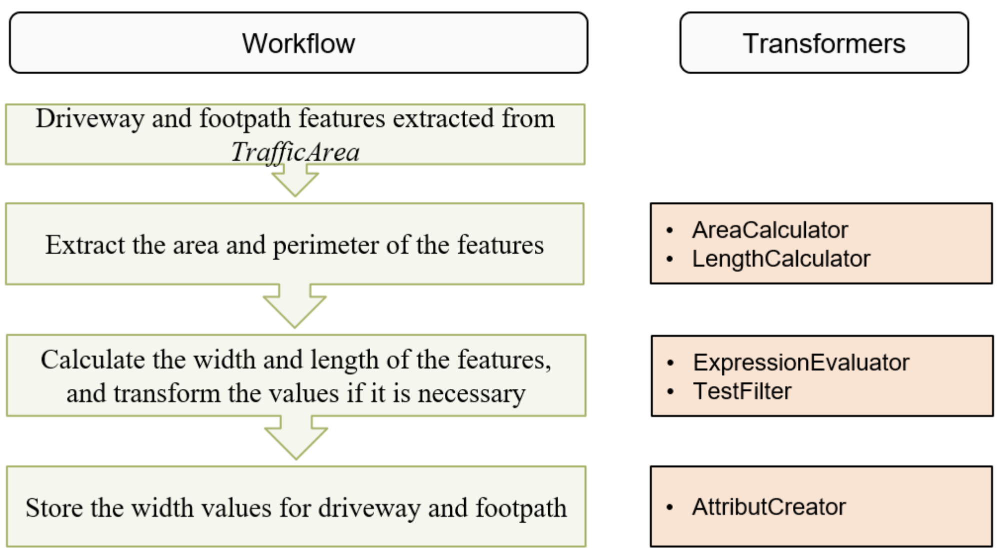


Figure 5-14 The simplified workflow for identifying the width of the driveways and footpath features in FME

After the optimised adjustment of the parking slot features, each feature represents a single parking slot. Their length and width are extracted using the BoundingBoxReplacer transformer. Besides, since some parking slots are within the parking lot, a parking angle is also required. The simplified process is illustrated in Figure 5-15. The BoundingBoxReplacer transformer is used to capture the rotation angle of the long side for the parking slot features. Next, the centrelines of the driveway features are extracted using CenterlineReplacer transformer. The NeighborFinder transformer identifies the driveway centreline features closest to each parking slot. One of the attributes exported by this transformer is *_candidate_angle*, which represents the driveway rotation angle in this case. Finally, the parking angle is calculated as the absolute difference between the parking slot's rotation angle and the driveway's rotation angle, using the ExpressionEvaluator transformer.

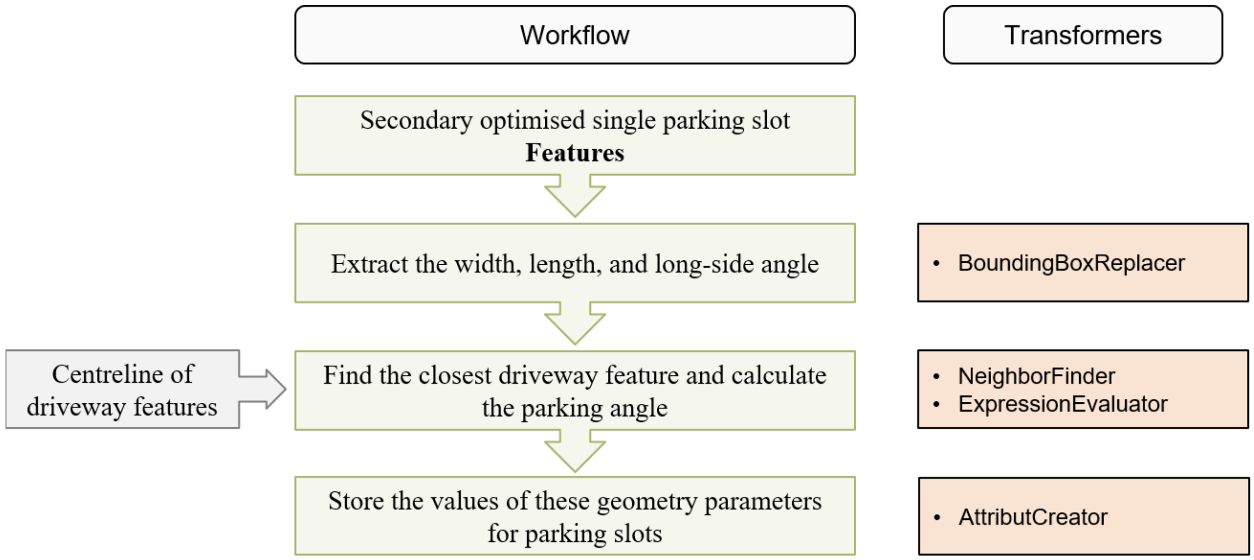


Figure 5-15 The simplified workflow for identifying the parking angle in FME

At this point, all the relevant information that can be extracted from this transportation dataset has been obtained. Finally, this information needs to be correctly written into CityGML 3.0. This is accomplished by using AttributeCreator to create *Generic* Attributes and store the corresponding values, as shown in the last step in Figure 5-14 and Figure 5-15.

5.2.2 Building Dataset

The dataset used in this study is from OpenData, accessible at https://geodaten.bayern.de/opengeodata/. It includes comprehensive 3D building models in Bavaria, with the building footprints and building attributes. The general workflow is illustrated in Figure 5-16.

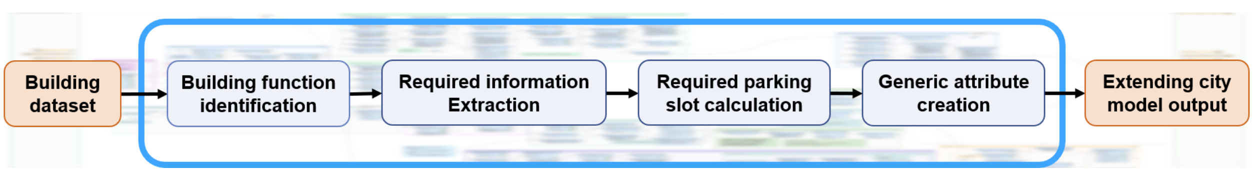


Figure 5-16 Schematic overview of the conversion process of building dataset in FME workbench, the orange is the in-/output, the steps within the blue boxes are the main process

First of all, the buildings adjacent to the parking slot areas are identified using the SpatialFilter. All buildings within a 50-m radius of the parking slots are captured. Then, the different functional buildings are extracted via the attribute *function* using the TestFilter. According to regulatory requirements (refer to the Appendix B.2 and B.3 for details), the required number of parking slots for each building can be calculate. Before the calculation, some parameters that are required during the calculation need to be extracted from the dataset. This involves determining the usable area and the number of rooms in the building, which are not directly provided in the dataset.

To estimate these parameters using existing attributes, the attribute *Flaeche* (*FootprintArea*) from the class *GroundSurface* is used. The usable area is calculated using the formula:

$$\text{UsableArea} = \text{Coefficient1} \times \text{Footprint Area} \times \text{Number of storeys}$$

Here, "Coefficient1" refers to the ratio of usable area per storey to the building footprint. It varies depending on the function of the building. The number of rooms is approximately derived using the formula:

$$\text{Number of Rooms} = \frac{\text{Usable Area}}{\text{Average room area}}$$

The "Average room area" varies depends on the building function.

In cases where the number of storeys is missing as well, the building's height can be used to estimate it by dividing the total height by the average floor height. The height also depends on the function's function, for example, 3 m with the residual buildings and 3.5 m with office buildings. Once these parameters are determined, the required number of parking slots can be calculated following the table in Appendix B.2 and B.3. Finally, the calculated number of required parking slots is stored back into the city model, extending the model with the new *Generic* attribute.

5.2.3 Geometries

This study only involved the extension of the *Generic* attributes without creating new city objects, so there is no need to redefine the geometries. However, due to a series of geometric transformations, the geometric properties in the source dataset may be changed. Therefore, it is crucial to maintain the geometry names and traits of the city objects before export. In the source datasets, the geometry name for *TrafficArea* is *lod2MultiSurface*, and for *TrafficSpace* is *lod2MultiCurve*.

5.3 Workflow Development

In this study, automated regulatory checking is based on the converted extended CityGML 3.0 data models (see section 5.2). To ensure the reproducibility of this process, a new workspace can be created for parking facility assessment against regulations, or work can continue in the original workspace to detect and correct minor issues or inconsistencies from previous steps efficiently. Here, they are conducted continuously in the same workspace in the FME workbench.

The entire checking process primarily involves examining the new attributes created in the previous stage, i.e., the regulation-related attributes extracted from the source dataset. The TestFilter transformer is mainly used to check these attributes. For the selected datasets, a total of six attributes are examined, as shown in Table 5-1.

Table 5-1 The examined attributes for the selected datasets

	Function Filter	Checked Attribute
Transportation dataset	Parking slot	*WidthOfParkingSlot*
		lengthOfParkingSlot
		angleOfParking
	Driveways	*widthOfDriveway*
	Footpath	*widthOfFootpath*
Building dataset	Various building functions	*requiredNumberOfPS*

For parking slots, the attributes are checked using the TestFilter to confirm that the length is more than 5 m and the width is more than 2.3 m. According to Article §4 (2), the width requirements for parking slots vary depending on their location. For instance, if one long side is obstructed by walls or other installations, the clear width must be at least 2.4 m. For disabled parking slots, the clear width must be at least 3.5 m. However, due to the limitations of this dataset, all parking slots are outdoors, and there is no information on disabled parking slots either. Therefore, this detail is not considered. All parking slots are assumed to be at least 2.3 m wide.

For driveways, the main checked attribute is the width. According to Article 4 (2), driveways that provide direct access to the parking slots have width constraints depending on the angle of the parking slot. For example, a driveway providing access to parking slots at a 90-degree angle must be at least 4.5 m wide. However, these width requirements do not apply to roadside parking slots (i.e., parallel to the driveway). Therefore, in this part, the driveway features near non-parallel parking slots are identified using SpatialFilter. Their widths are then checked based on the partial table of different parking angles, shown in Table 5-2 (see Article 4 (2) in Appendix B.1 for the whole table)

Table 5-2 The incomplete table of the required driveway width for different parking angles

Arrangement of the parking slots to the driveway	Required width of driveway (in m) for a parking slot width of: 2,30 m
90°	6,50
60°	4,50
45°	3,50

Regulations only apply to footpaths in large parking lots. Due to the lack of data, this dataset does not contain information on large parking lots. However, to test the usability of this checking process, footpaths near the parking slots are checked against the regulations for footpaths in large parking lots (Articles 2 (5) and 3 (3)).

For the building datasets, the required number of parking slots for each building has already been calculated in the previous sub-section 5.2.2. In this part, the original number of the parking slot needs to be extracted from the transportation dataset. However, similar to the transportation dataset, there is also an issue of data incompleteness — specifically, the lack of necessary property information to determine which parking slots are associated with which buildings. To test the workflow and feasibility of the checking process, the parking slots within a 50-m radius of each building are identified and assigned as the original parking slot number for that building (i.e., attribute *originalNumberOfPS*). This method is similar to the approach used for footpaths. Due to the lack of specific data, an alternative solution is implemented. This allows for comparing the calculated required number and the original actual number of parking slots.

All the primary checking results can first be directly stored through the AttributeCreator, with specific attributes detailed in Section 4.1.5. These results are then be utilised to generate the output in the next sub-section. The schematic diagram is illustrated in Figure 5-17.

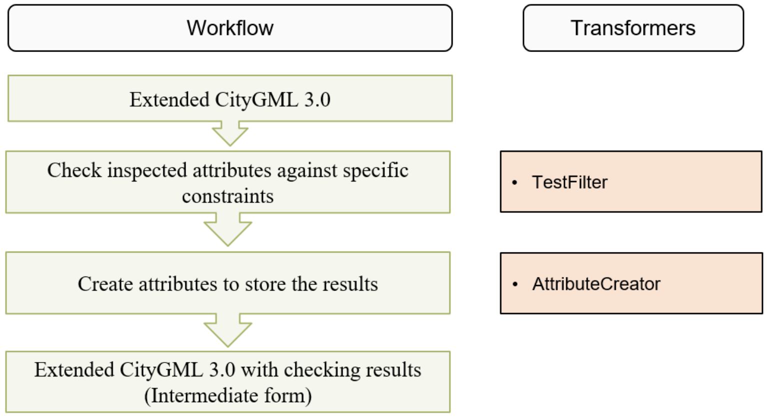


Figure 5-17 The simplified workflow for the checking process in FME workbench

5.4 Results and Outputs

5.4.1 Assessment Report

The assessment reports are presented as HTML reports, divided into two parts: an interactive 2D web map and a textual report. The HTML Writer is utilised in the workspace to generate the output. The overall workflow is illustrated in Figure 5-18. This process involves multiple transformers. Each HTMLReportGenerater transformer is responsible for a specific section of the report.

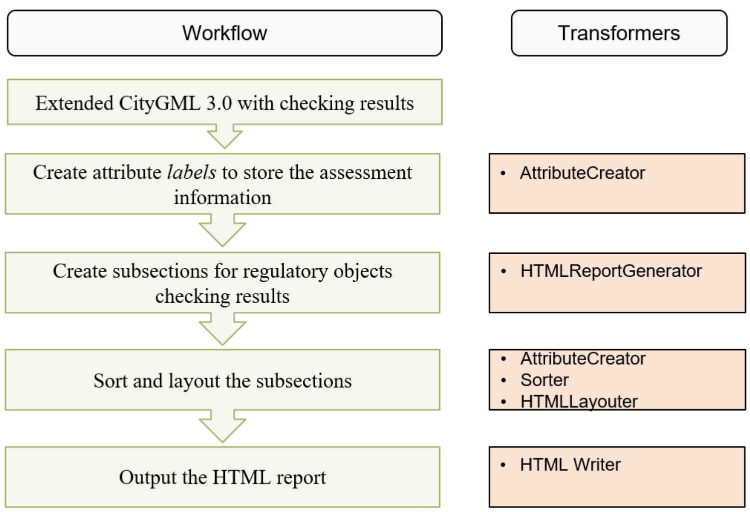


Figure 5-18 The simplified workflow for generating assessment reports in the FME workbench

For the simple 2D visualisation, a lightweight Leaflet map is utilised, providing basic display functionalities. This 2D web map is flexible in terms of dataset format and version, allowing the simultaneous export of the transportation and building datasets. The HTMLReportGenerator is able to create the web map directly, with one attribute selectable as the label attribute for display purposes on the map.

A new attribute label is created using the AttributeCreator to provide comprehensive information. This attribute contains multiple pieces of information, including the feature ID, specific article (e.g.,

§4-1), inspected attributes (e.g., the parking slot length and width), assessment result, and reason for non-compliance. The settings for this attribute are shown in Figure 5-19. Inspected attributes, like width and length, are rounded to three decimal places for better readability.

The base map employed here is the digital orthophotos in WMTS Geobasisdaten Bayern, available via OpenData. The feature layer can be accessed through the following URL: https://geoservices.bayern.de/od/wmts/geobasis/v1/1.0.0/WMTSCapabilities.xml (OpenData-DOP40, 2024). The final Leaflet map in the HTML report is shown in Figure 5-20. It supports some basic interactions, such as zooming and selecting features. When a specific feature is selected, its label attribute is displayed.

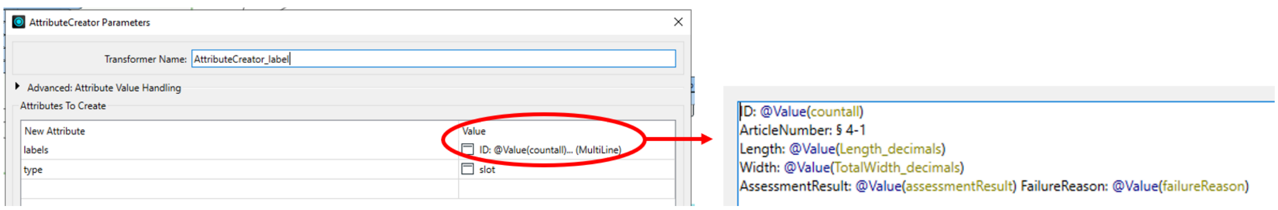


Figure 5-19 The setting of the attribute labels

Figure 5-20 The leaflet map output in HTML report.

The textual report divides the content into six sections based on the selected articles and inspected attributes. Each section contains an overview and a table displaying the inspected attributes and assessment results. A total of six HTMLReportGenerator are required. Since the implementation process is similar across sections and this case primarily involves Article 4 (Parking slots and driveways), the parking slots are mainly used as an example. Other sections are not discussed separately.

The HTMLReportGenerator allows the selection and ordering of content for display through the Page Contents parameter, followed by editing through the Content Settings parameter. Here, a combination of basic contents and custom HTML is used to arrange the layout of the section title, overview, and table. Firstly, the section title and overview are edited in the Header of the Page Contents, with the Header Level set to "H2" and "None", respectively. Next, an accordion table is created, along with a query interface where users can manually enter the feature ID of interest. This step requires some JavaScript. Three custom HTML elements and a table are created in the Page Content, defining the CSS style for the accordion button and the layout of the accordion section.

To simplify and shorten the table, it only displays features that did not comply with the regulations. Each of the six HTMLReportGenerators is configured as described above for their respective sections. An additional HTMLReportGenerator is used to set the title and introduction for the entire report. After arranging the order of the articles using the Sorter and HTMLLayout transformers, the final textual report is exported through HTMLReport Writer. This process automates the generation of the HTML report, as illustrated in Figure 5-18. The initial interface is shown in Figure 5-21, and the accordion part can be opened for viewing and querying as needed, as shown in Figure 5-22.

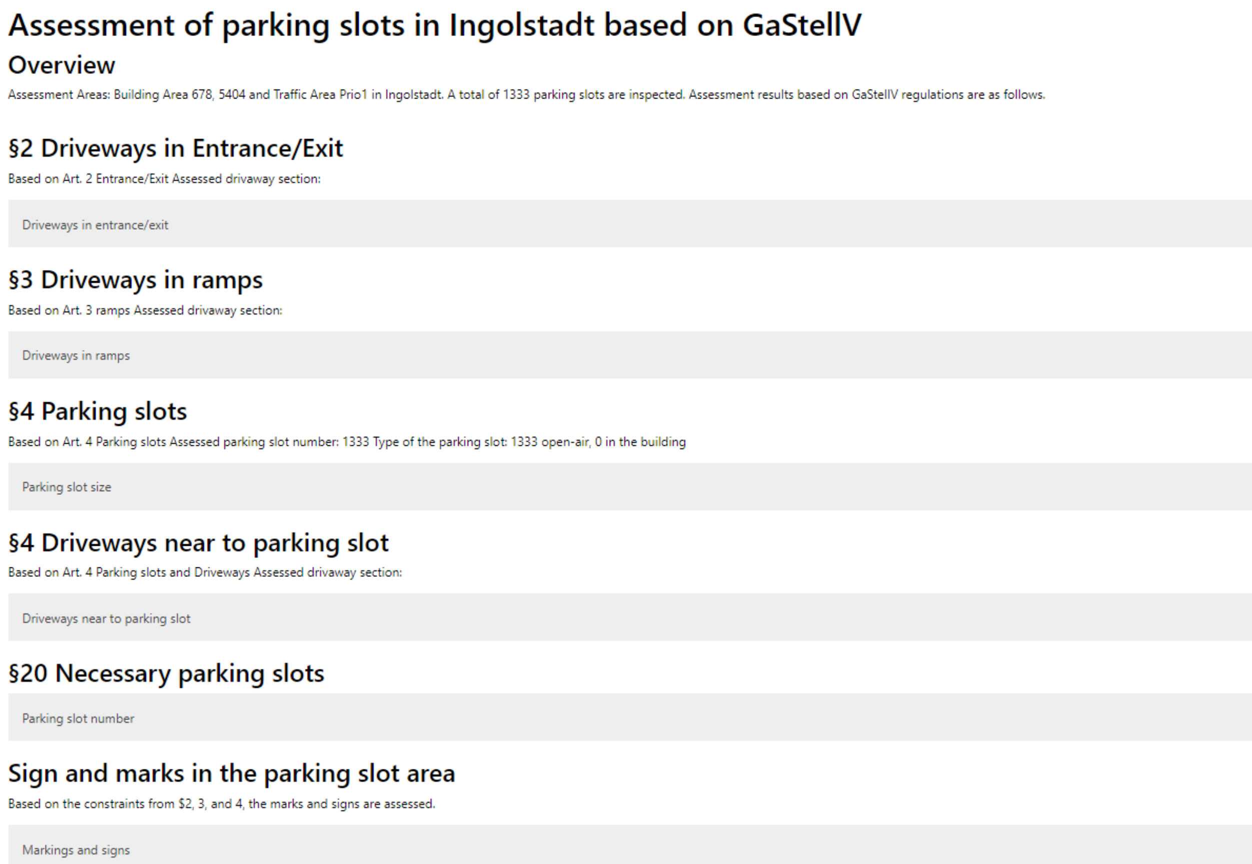


Figure 5-21 The initial interface of the assessment report.

§4 Parking slots

Based on Art. 4 Parking slots. Assessed parking slot number:1332. Type of the parking slot: 1332 open-air, 0 in the building.

Parking slot size

Sort by description | Search names

Nr. Non-compliant parking slots	Location	Explanations
5	Ingolstadt	According to §4-1, the width of the parking slot (1.983) does not comply with the specified requirement (2.3 m).
6	Ingolstadt	According to §4-1, the width of the parking slot (1.983) does not comply with the specified requirement (2.3 m).
7	Ingolstadt	According to §4-1, the width of the parking slot (1.983) does not comply with the specified requirement (2.3 m).
8	Ingolstadt	According to §4-1, the width of the parking slot (2.093) does not comply with the specified requirement (2.3 m).
9	Ingolstadt	According to §4-1, the width of the parking slot (2.093) does not comply with the specified requirement (2.3 m).

Figure 5-22 An example of an accordion table in the assessment report

5.4.2 Extending Data Model

The results from the previous step (see section 5.3) can also be stored in the extended CityGML 3.0 data model. According to the regulation extension content in section 4.1.5 (see Table 4-12), four *Generic* attributes are created to store the assessment results. These attributes include *administrationArea*, *articleNumber*, *assessmentResult*, *failureReason*. The values for *assessmentResult* and *failureReason* are determined by the results of the TestFilter from the previous checking process, while the values for *administrationArea* and *articleNumber* are manually input. Then, the attributes are created, and corresponding values are stored using AttributeCreator. For instance, *assessmentResult* can be "T" or "F", while *failureReason* provides a brief description of these results. Different checked objects align with specific articles and paragraphs of the regulations.

The final data model is exported to the generic GML Writer with the XSD schema file. The simplified workflow is illustrated in Figure 5-23. This GML file exchanged in CityGML 3.0 format can be imported and viewed in the FME Inspector. Figure 5-24 shows the checked attributes and assessment results for a specific parking slot. The results indicate that this parking slot does not comply with Article 4.1 of the regulation. The reason is that its length is shorter than required.

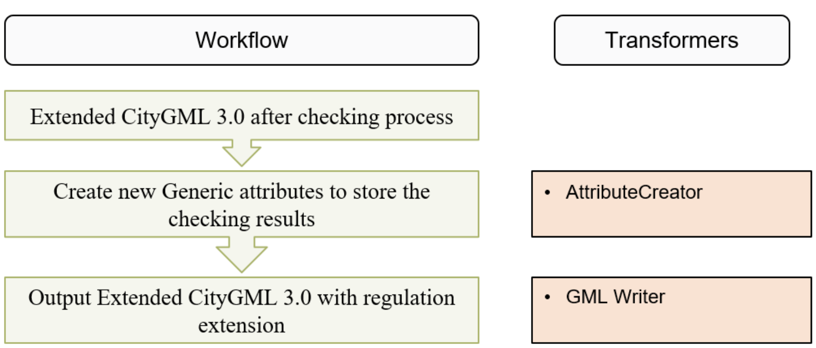


Figure 5-23 The simplified workflow for the output of the extended CityGML 3.0 with the regulation extension in the FME workbench

Figure 5-24 Screenshot of the Generic attribute list for a parking slot in the final CityGML data model

5.4.3 Visualisation

In this study, the visualisation of the inspection results is achieved using ArcGIS Pro. The schematic representation is illustrated in Figure 5-25. The datasets are exported using the Esri Geodatabase (File Geodb) Writer in FME and then viewed in ArcGIS Pro. Before exporting, a new attribute *difference* is created in Building datasets, recording the discrepancy between the actual number and the required number of parking slots. This attribute is used for the gradient colouring of buildings to enhance visualisation.

The datasets are opened in ArcGIS Pro and converted to a 3D scene using the Convert in the View tab. Since the datasets already contain 3D information, additional height extrusion or elevation settings are not needed. The overview of this 3D scene is shown in Figure 5-26. By clicking on any features, the checked parameters and inspection results are displayed in a popup, as shown in Figure 5-27.

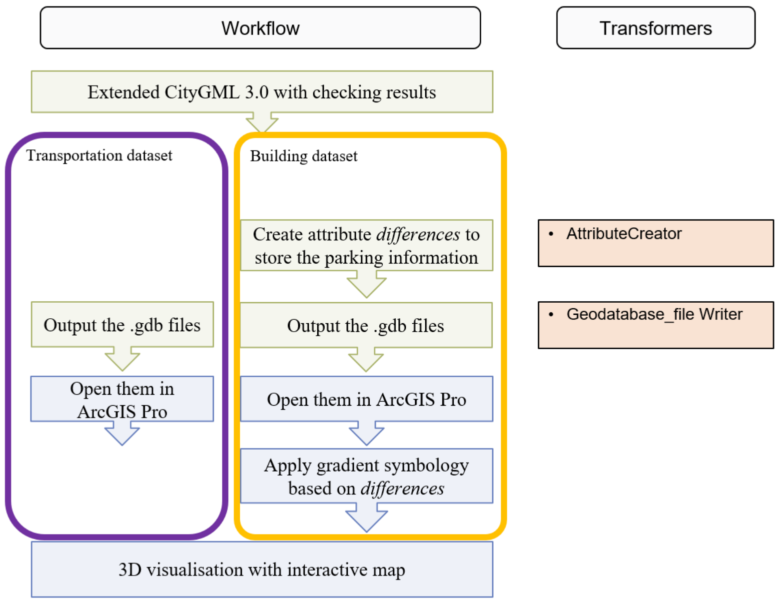


Figure 5-25 The simplified workflow for the output of the 3D visualisation. The green boxes show the steps in the FME Workbench, the blue boxes show the steps in ArcGIS Pro

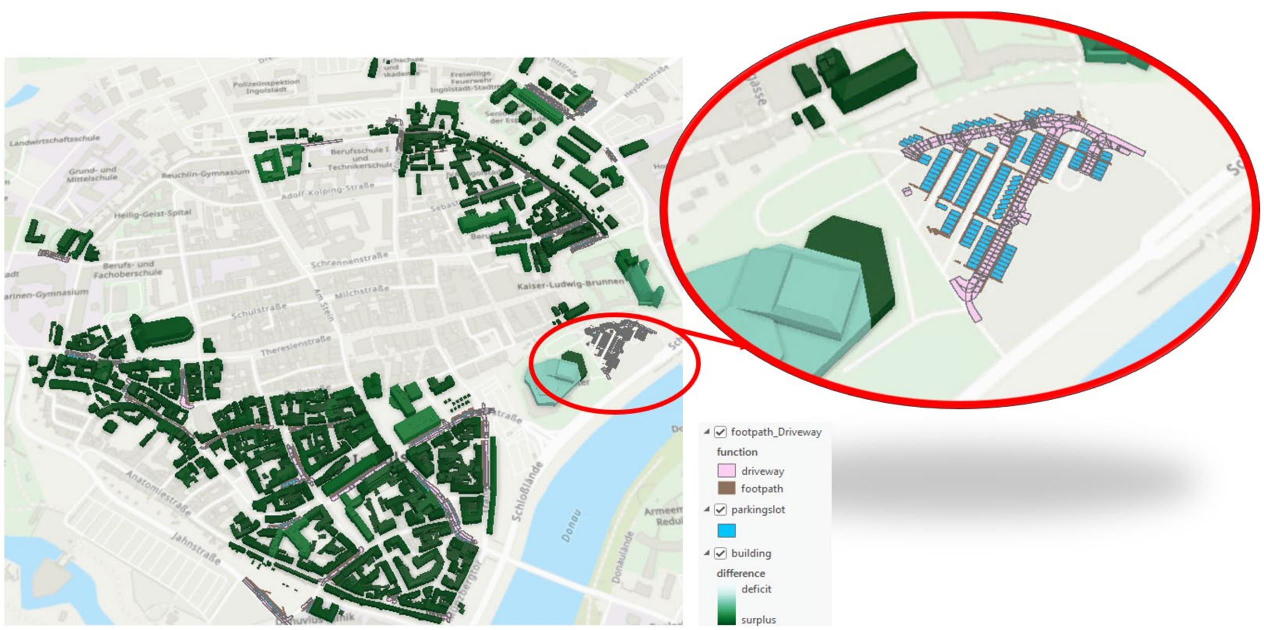


Figure 5-26 Screenshot of the overview of the 3D visualisation in ArcGIS Pro

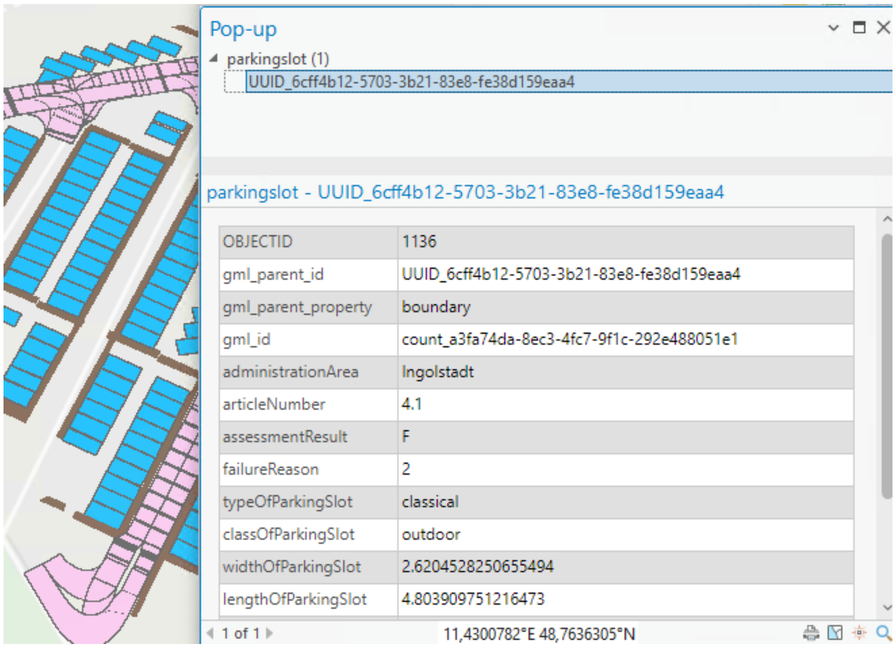


Figure 5-27 Screenshot of the attribute details for a parking slot in ArcGIS Pro

6. Discussion

This thesis proposes a workflow for the automated assessment of the parking facilities using CityGML 3.0, specifically through the derived geometric and semantic information. It involves two main parts: the conceptual framework and the case study. It achieves its research objectives and offers a new application case for the revised Transportation model in CityGML 3.0. In this chapter, a further discussion of the outcomes, potentials, and limitations is presented.

6.1 Legal Requirements and Extension Options

The parking facilities are always assessed independently in this study. Only when determining whether the actual provision of parking slots meets the building's needs are these facilities associated with buildings. However, in practice, parking slots are not only constrained by parking regulations but are also subject to various limitations imposed by building law, city land use planning, and development plans. This study primarily focuses on the constraints imposed by parking regulations on parking slots and does not investigate these interrelated and interdependent laws and regulations.

Mapping the regulatory conceptual model to CityGML 3.0 demonstrated the capability of semantic 3D city models for regulatory assessment. As mentioned at the beginning of Chapter 4, the approach to solving this problem is not unique. The implemented solution follows the general mapping principle, i.e., efficiency and minimal extension. Another possibility is based on a formal extension mechanism, i.e., ADE, in which objects based on parking regulations can be defined in an extra conceptual schema with its own specific namespace. Regulatory objects, such as checking results, can be stored in new city objects instead of being mapped to *TrafficArea*. This approach requires adjusting the locations of the corresponding regulatory attributes and considering their parent classes, i.e., where they derive from.

Compared to the application in this thesis, creating a regulatory ADE can more systematically describe the entire regulatory framework and corresponding details using the structured conceptual schema provided in UML diagrams. On the other hand, creating an ADE is also a complex and meticulous process. When only a few articles from the regulation are selected and used for automated assessment, an ADE with one or two new feature classes may be overly elaborate. In such cases, extending using *Generic* attributes can adequately meet the needs as well. However, in future applications, creating an ADE will still become necessary. When too many regulatory objects and attributes are introduced, *Generic* objects and attributes may no longer suffice. In conclusion, only testing and application through the use cases for these different extension strategies can determine which is the most practical.

6.2 Implementation Difficulties

One of the major challenges of this case study is preprocessing the source data, which requires distinct steps for different functional surfaces. The source dataset is exchanged in CityGML 3.0, converted from a 3D street space model using high-accurate OpenDrive data via r:trån. This original data was collected using a mobile mapping system. For certain reasons, some semantic information was lost or not recorded during the collection and conversion process. For example, some features originally associated with driveways adjacent to parking slots were not defined in the attribute *function*, resulting in the loss of some useful features.

In the datasets, some feature geometries did not align with actual plots, necessitating the identification of parking slots and the determination of their geometric parameters. While all parking slots are theoretically rectangular, they are allowed to be angled with the driveway, meaning non-perpendicular arrangements are permitted. Parking slots might take the shape of parallelograms or trapezoids. Although there are many methods to find the largest rectangle within a parallelogram or trapezoid, they require intensive mathematical calculations in general. In these datasets, the high-resolution geometry generated numerous vertices and spikes, complicating the extraction of geometric parameters using FME Workbench. Identifying the largest rectangles within these complex polygons is more challenging due to potential misjudgement of polygon shapes and incorrect clipping. Currently, FME lacks a transformer capable of this task. Therefore, finding an appropriate solution is both difficult and time-consuming.

Considering the actual conditions of the parking slots and parking logic, the problem can be converted to finding the approximate largest rectangle within an imperfect parallelogram, allowing for some boundary overlap or extension. Ultimately, by combining various transformers and utilising their geometric relationships, polygon transformations are performed to achieve the final goal, avoiding complex mathematical calculations.

6.3 Limitations

Although this thesis has generally achieved its research objectives, several limitations remain. These limitations mainly stem from regulation combination, datasets, and data processing.

As mentioned in Section 6.1, this study focuses on relatively isolated parking slots and their attributes, without integrating them with other relevant laws, regulations, and development plans. On one hand, during the calculation of the required parking slot number for a building, some required assessment attributes are missing, which could potentially be introduced through development plans and land use planning (e.g., usable area and number of rooms within a building). Estimating these missing attributes through virtual coefficients based on the footprint may introduce errors in the calculation of required parking slots. On the other hand, certain parameters are not considered in GaStellV and GaStS, such as greenery requirements within parking lots and restrictions on ground materials. Including more parameters would make the assessment more comprehensive and increase the overall credibility of the results.

The limitations of the datasets also affect the study. The source data, as a street space data model, contains only some outdoor parking slot data and lacks information on in-building parking facilities. As a result, not all attributes defined in Chapters 3 and 4 could be tested. Furthermore, during data processing, some parking slot buffer zones in the source dataset are also identified as parking slots (*TrafficArea*, *function* with “7”). Due to the lack of clear boundaries and marks for parking slot identification, these zones are uniformly processed according to the general method described in Section 5.2, without individual handling, potentially leading to certain errors.

Additionally, the entire implementation was developed in the FME workbench, which also has some workspace limitations. The workspace is relatively large, leading to slower speeds during the later stages of work and debugging. This is due to the large size of the test dataset and the extensive geometric processing required. Some geometric processing steps cannot be fully automated and require manual rechecking and reprocessing.

7. Conclusion

This thesis provides an automated workflow for assessing parking facilities at the city scale by extending the CityGML 3.0 city model. A use-case approach is adopted to conduct this assessment workflow in accordance with the parking standards in Ingolstadt, Bavaria. The relevant regulations within the study area are interpreted and formalised to extract relevant regulatory objects and attributes, forming a conceptual model of the regulations. This conceptual model is subsequently mapped into CityGML 3.0 with Generic attribute extensions. The tested datasets, including transportation and building datasets, are converted in the FME workbench to derive the required attributes for the regulatory assessment. Eventually, a checking process is developed to generate outputs that present the assessment results.

7.1 Conclusion of the Research Objectives

The core research question of this thesis is systematically addressed through a series of sub-questions. In this section, the sub-questions are answered step by step firstly, building up to the final answer to the main research question.

- *What are the relevant regulations regarding parking facilities in the city?*

  The relevant regulations regarding parking facilities vary by region and city. In Germany, these regulations follow a hierarchical structure, from federal traffic and building laws down to specific local municipal regulations, creating an interconnected framework. Within this framework, parking facilities are primarily constrained by the local parking regulations specific to each city. In the case study area of this research, located in Ingolstadt, Bavaria, the relevant parking standards are GaStS and GaStellV. These standards regulate the dimensions, quantity, and ancillary infrastructure of parking slots. Furthermore, GaStellV includes construction constraints for buildings when the parking facilities are located indoors. However, this study focuses only on the articles that are easily quantifiable and directly related to the parking facilities themselves, excluding those interconnected with broader building requirements and development plans.

- *How can the objects and their attributes extracted from the regulations be mapped into CityGML 3.0?*

  This question can be divided into two parts to answer: 1. How can the relevant objects and their attributes be extracted from the regulations? And 2. How can they be mapped into CityGML 3.0 further?

  By interpreting and formalising the regulations, the objects and related attributes involved in the regulations can be extracted, which represent the information required for assessment. In this study, these objects include parking slots and various ancillary objects. The buildings to which the parking slots belong are also examined to determine the required parking slot number. The corresponding attributes include geometric information, such as the length and width of parking slots and the width and slope of driveways, as well as semantic information, such as the type of parking slots and more. These objects and attributes are initially stored in the regulation conceptual model.

To store the data and present the data exchange process, the regulation conceptual model can be mapped to CityGML 3.0 city model and extended if necessary, following general mapping principles, namely, efficient integration and extension minimisation (see Chapter 4). Through semantic analysis and manual mapping, all regulatory objects in this study can be directly mapped to the city objects defined in the CityGML Conceptual Model. However, the new attributes need to be defined to store the corresponding regulatory attributes, as the existing ones do not fulfil all the criteria. Based on this extension requirement, the *Generic* attribute extension mechanism in CityGML can be employed.

- *How can the assessment be realised and automated against the regulations?*

A workflow is developed to assess parking facilities using FME Workbench. Based on the answers to the above two questions, a case study approach is used to test the feasibility of the workflow, as detailed in Chapter 5. First, the source datasets are pre-processed, i.e., converted and calculated to obtain the required attributes. Then, several transformers are applied to conduct regulatory checking on the dataset, verifying that all relevant features meet the specified standards. Finally, the workflow generates an HTML report with a Leaflet map, CityGML 3.0 files containing the assessment results, and 3D visualisation in ArcGIS Pro.

Based on these sub-questions, the final answer to the main research question is as follows:

How can parking facilities be assessed against regulations at the city scale using CityGML 3.0?

This study utilises CityGML 3.0 and its extensions to perform regulatory assessments of parking facilities by developing a corresponding workflow to automate this process. The process begins with interpreting the regulations to generate a conceptual model, which is then mapped into an extended CityGML 3.0 data model. For use-case datasets, inspection information is extracted from the data model and stored in new attributes. Calculations and assessments are made based on specific regulatory constraints. Finally, the outputs are generated to present the inspection results in multiple aspects.

7.2 Future Work

To improve and enhance the automation process of the assessment against parking regulations and extend it to more regulatory domains and regions, further efforts can be continued in the following areas in the future.

This study utilised a specific case study to investigate the parking regulations for the city of Ingolstadt, covering only certain articles of these regulations. In the future, this use case can be expanded to cover the entire set of parking regulations and even extend to other regulations fields and cities. Additionally, since parking regulations are closely related to development planning and building permits, integrating them with the XPlanung data standard—used for urban and regional planning in Germany—could further enhance the efficiency and feasibility of parking inspections. This integration could help restrict the legality of parking lot locations. Moreover, more ancillary objects and detailed attributes of parking slots could be included, such as greenery near the parking facilities.

When extending to all articles in GaStellV, the regulatory conceptual model will become more complex. Mapping to CityGML 3.0 is inherently non-unique. In such cases, creating an ADE could be a better solution. This approach requires a more meticulous mapping process, examining and initial mapping of existing ADEs.

This study assessed existing parking facilities. In future work, the results can serve as a macro control of urban construction and a reference for future development plans. The workflow can also be applied to building permit approval, automating parts of the building permit inspection process, such as the construction of parking facilities. Other building permit-checking processes, such as determining the required parking slots for a specific building, can also utilise this workflow. Through this, government officials and experts can verify that parking facilities meet regulatory requirements in real-world scenarios. This automated process can reduce workload, improve efficiency, and accelerate the overall process.

References

Agugiaro, G., Benner, J., Cipriano, P., Nouvel, R., 2018. The Energy Application Domain Extension for CityGML: enhancing interoperability for urban energy simulations. Open geospatial data, softw. stand. 3, 2. https://doi.org/10.1186/s40965-018-0042-y

Akahoshi, K., Ishimaru, N., Kurokawa, C., Tanaka, Y., Oishi, T., Kutzner, T., Kolbe, T.H., 2020. I-Urban revitalization: Conceptual modeling, implementation, and visualization towards sustainable urban planning using CityGML. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 4, 179–186.

Altıntaş, Y.D., Ilal, M.E., 2021. Loose coupling of GIS and BIM data models for automated compliance checking against zoning codes. Automation in Construction 128, 103743.

Amor, R., Dimyadi, J., 2021. The promise of automated compliance checking. Developments in the built environment 5, 100039.

ASAM OpenDRIVE® [Website], n.d. . ASAM. URL https://www.asam.net/standards/detail/opendrive/ (accessed 6.18.24).

Bachert, C., 2023. Mapping the Energy ADE to CityGML 3.0.

Beach, T.H., Rezgui, Y., Li, H., Kasim, T., 2015. A rule-based semantic approach for automated regulatory compliance in the construction sector. Expert Systems with Applications 42, 5219–5231.

Becker, T., Nagel, C., Kolbe, T.H., 2013. Semantic 3D Modeling of Multi-Utility Networks in Cities for Analysis and 3D Visualization, in: Pouliot, J., Daniel, S., Hubert, F., Zamyadi, A. (Eds.), Progress and New Trends in 3D Geoinformation Sciences, Lecture Notes in Geoinformation and Cartography. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 41–62. https://doi.org/10.1007/978-3-642-29793-9_3

Beil, C., Kendir, M., Ruhdorfer, R., Kolbe, T.H., 2022. Dynamic and web-based 4D visualization of streetspace activities derived from traffic simulations and semantic 3D city models. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 10, 29–36.

Beil, C., Kolbe, T.H., 2017. CityGML and the streets of New York-A proposal for detailed street space modelling, in: Proceedings of the 12th International 3D GeoInfo Conference 2017. pp. 9–16.

Beil, C., Ruhdorfer, R., Coduro, T., Kolbe, T.H., 2020. Detailed streetspace modelling for multiple applications: Discussions on the proposed CityGML 3.0 transportation model. ISPRS International Journal of Geo-Information 9, 603.

Benner, J., Geiger, A., Häfele, K.-H., 2010. Concept for building licensing based on standardized 3D geo information. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 38, 9–12.

Biljecki, F., Kumar, K., Nagel, C., 2018. CityGML Application Domain Extension (ADE): overview of developments. Open geospatial data, softw. stand. 3, 13. https://doi.org/10.1186/s40965-018-0055-6

Brasebin, M., Perret, J., Mustière, S., Weber, C., 2018. 3D urban data to assess local urban regulation influence. Computers, Environment and Urban Systems 68, 37–52. https://doi.org/10.1016/j.compenvurbsys.2017.10.002

Change toolkit for digital building permit - CHEK – Change toolkit for digital building permit, 2022. URL https://chekdbp.eu/ (accessed 3.23.23).

Christiansen, P., Engebretsen, Ø., Fearnley, N., Hanssen, J.U., 2017. Parking facilities and the built environment: Impacts on travel behaviour. Transportation Research Part A: Policy and Practice 95, 198–206.

Ciribini, A.L.C., Ventura, S.M., Paneroni, M., 2016. Implementation of an interoperable process to optimise design and construction phases of a residential building: A BIM Pilot Project. Automation in construction 71, 62–73.

Copyright (c) Safe Software Inc (2023), FME Software (v2023.0.1) [Computer software].

CORENET X [Website], 2024. URL https://portal.corenet.gov.sg/ (accessed 7.7.24).

Deng, T., Zhang, K., Shen, Z.-J.M., 2021. A systematic review of a digital twin city: A new pattern of urban governance toward smart cities. Journal of Management Science and Engineering 6, 125–134.

Dimyadi, J., Amor, R., 2013. Automated building code compliance checking–where is it at. Proceedings of CIB WBC 6.

Eastman, C., Lee, Jae-min, Jeong, Y., Lee, Jin-kook, 2009. Automatic rule-based checking of building designs. Automation in Construction 18, 1011–1033. https://doi.org/10.1016/j.autcon.2009.07.002

Eirinaki, M., Dhar, S., Mathur, S., Kaley, A., Patel, A., Joshi, A., Shah, D., 2018. A building permit system for smart cities: A cloud-based framework. Computers, Environment and Urban Systems 70, 175–188.

Esri. (2023). ArcGIS Pro (v 3.2.2) [Computer software].

EU4dbp [Website], n.d. . eu4dbp.net. URL https://eu4dbp.net/ (accessed 3.21.23).
European Commission, 2002. eEurope 2005: An information society for all. European Commission, Brussels.
Esri. (2023). ArcGIS Pro (v 3.2.2) [Computer software].
Gröger, G., Plümer, L., 2012. CityGML – Interoperable semantic 3D city models. ISPRS Journal of Photogrammetry and Remote Sensing 71, 12–33. https://doi.org/10.1016/j.isprsjprs.2012.04.004
Guler, D., Yomralioglu, T., 2022. Reviewing the literature on the tripartite cycle containing digital building permit, 3D city modeling, and 3D property ownership. Land Use Policy 121, 106337.
Guler, D., Yomralioglu, T., 2021. A reformative framework for processes from building permit issuing to property ownership in Turkey. Land Use Policy 101, 105115.
Ideas in FME Community [Website], 2019. URL https://community.safe.com/ideas/find-largest-internal-rectangle-from-a-polygon-30960 (accessed 6.20.24).
Interactive Instruments [Website], 2024. . Shapechange. URL https ://shape chang e.net/. (accessed 2.5.24).
Kirschner, F., Lanzendorf, M., 2020. Parking management for promoting sustainable transport in urban neighbourhoods. A review of existing policies and challenges from a German perspective. Transport Reviews 40, 54–75.
Klementschitz, R., Stark, J., Sammer, G., 2007. Integrating mobility management in land development planning with off-street parking regulations. Journal of urban planning and development 133, 107–113.
Kolbe, T.H., 2009. Representing and exchanging 3D city models with CityGML, in: 3D Geo-Information Sciences. Springer, pp. 15–31.
Kolbe, T.H., Kutzner, T., Smyth, C.S., Nagel, C., Roensdorf, C., Heazel, C., 2021. OGC City Geography Markup Language (CityGML) Part 1: Conceptual Model Standard.
Kutzner, T., Chaturvedi, K., Kolbe, T.H., 2020. CityGML 3.0: New functions open up new applications. PFG–Journal of Photogrammetry, Remote Sensing and Geoinformation Science 88, 43–61. https://doi.org/10.1007/s41064-020-00095-z
Kutzner, T., Kolbe, T.H., 2016. Extending semantic 3D city models by supply and disposal networks for analysing the urban supply situation, in: Lösungen Für Eine Welt Im Wandel, Dreiländertagung Der SGPF, DGPF Und OVG, 36. Wissenschaftlich-Technische Jahrestagung Der DGPF. pp. 382–394.
Labetski, A., van Gerwen, S., Tamminga, G., Ledoux, H., Stoter, J., 2018. A proposal for an improved transportation model in CityGML. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 42, 89–96.
Lee, H., Lee, J.-K., Park, S., Kim, I., 2016. Translating building legislation into a computer-executable format for evaluating building permit requirements. Automation in construction 71, 49–61.
Liamis, T., Mimis, A., 2022. Establishing semantic 3D city models by GRextADE: the case of the greece. Journal of Geovisualization and Spatial Analysis 6, 15.
Liebich, T., Wix, J., Forester, J., Qi, Z., 2002. Speeding-up the building plan approval–the Singapore e-plan checking project offers automatic plan checking based on IFC. Presented at the E-Work and E-Business in Architecture, Engineering and Construction, Proc. of 4th European Conference on Product and Process Modelling, Portoroz, Balkema, Rotterdam, pp. 467–471.
Löwner, M.-O., Gröger, G., Benner, J., Biljecki, F., Nagel, C., 2016. Proposal for a new LOD and multi-representation concept for CityGML. ISPRS annals of the photogrammetry, remote sensing and spatial information sciences 4, 3–12.
Malsane, S., Matthews, J., Lockley, S., Love, P.E., Greenwood, D., 2015. Development of an object model for automated compliance checking. Automation in construction 49, 51–58.
Marsden, G., 2006. The evidence base for parking policies—a review. Transport policy 13, 447–457.
Müller, F., 2021. Entwicklung einer CityGML Application Domain Extension: Der Amtliche Lageplan in 3D (matser thesis). Technischen Universität München.
Nega, A., Coors, V., 2022. The use of CITYGML 3.0 IN 3D cadastre system: the case of ADDIS ABABA CITY. The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences 48, 109–116.
Nguyen, T.-H., Kim, J.-L., 2011. Building code compliance checking using BIM technology. Presented at the Proceedings of the 2011 winter simulation conference (wsc), IEEE, pp. 3395–3400.
Noardo, F., Wu, T., Arroyo Ohori, K., Krijnen, T., Tezerdi, H., Stoter, J., 2020. Geobim for digital building permit process: Learning from a case study in Rotterdam. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 6, 151–158.

OGC, 2022a. CityGML [Website]. Open Geospatial Consortium. URL https://www.ogc.org/standards/citygml (accessed 6.1.22).
OGC, 2022b. Urban-Planning-ADE [Website]. URL https://github.com/opengeospatial/CityGML3.0-GML-Encoding/tree/main/resources/examples/ADE-examples/Urban-Planning-ADE (accessed 3.12.24).
Olsson, P.-O., Axelsson, J., Hooper, M., Harrie, L., 2018. Automation of building permission by integration of BIM and geospatial data. ISPRS International Journal of Geo-Information 7, 307.
Open Knowledge Foundation [Website], n.d. URL https://opendatahandbook.org/glossary/en/terms/machine-readable/ (accessed 3.3.24).
OpenData [Website], 2024. URL https://geodaten.bayern.de/opengeodata/OpenDataDetail.html?pn=lod2 (accessed 6.18.24).
OpenData-DOP40 [Website], 2024. URL https://geodaten.bayern.de/opengeodata/OpenDataDetail.html?pn=dop40 (accessed 6.26.24).
Padsala, R., Gebetsroither-Geringer, E., Bao, K., Coors, V., 2021. The Application of CityGML Food Water Energy ADE to Estimate the Biomass Potential for a Land Use Scenario, in: CITIES 20.50–Creating Habitats for the 3rd Millennium: Smart–Sustainable–Climate Neutral. Proceedings of REAL CORP 2021, 26th International Conference on Urban Development, Regional Planning and Information Society. CORP–Competence Center of Urban and Regional Planning, pp. 851–861.
Peng, J., Liu, X., 2023. Automated code compliance checking research based on BIM and knowledge graph. Scientific Reports 13, 7065.
Preidel, C., Borrmann, A., 2018. BIM-Based Code Compliance Checking, in: Borrmann, A., König, M., Koch, C., Beetz, J. (Eds.), Building Information Modeling. Springer International Publishing, Cham, pp. 367–381. https://doi.org/10.1007/978-3-319-92862-3_22
Preidel, C., Borrmann, A., 2016. Towards code compliance checking on the basis of a visual programming language. Journal of Information Technology in Construction (ITcon) 21, 402–421.
r:trån [Website], 2020. URL https://rtron.io/ (accessed 6.18.24).
Saeidian, B., Rajabifard, A., Atazadeh, B., Kalantari, M., 2023. A semantic 3D city model for underground land administration: Development and implementation of an ADE for CityGML 3.0. Tunnelling and underground space technology 140, 105267.
Sinning-Meister, M., Gruen, A., Dan, H., 1996. 3D City models for CAAD-supported analysis and design of urban areas. ISPRS Journal of Photogrammetry and Remote Sensing 51, 196–208.
Solihin, W., Eastman, C., 2015. Classification of rules for automated BIM rule checking development. Automation in construction 53, 69–82.
Sparx Systems [Website], 2024. Enterprise architect. URL http://www.sparx systems.com/products/ea/index.html. (accessed 2.5.23).
Tamminga, G., van den Brink, L., Van Lint, H., Stoter, J., Hoogendoorn, S., 2013. Toward GIS-Compliant Data Structures for Traffic and Transportation Models.
Uggla, M., Olsson, P., Abdi, B., Axelsson, B., Calvert, M., Christensen, U., Gardevärn, D., Hirsch, G., Jeansson, E., Kadric, Z., 2023. Future Swedish 3D city models—Specifications, test data, and evaluation. ISPRS International Journal of Geo-Information 12, 47.
Van Berlo, L., Dijkmans, T., Stoter, J., 2013. Experiment for integrating Dutch 3D spatial planning and BIM for checking building permits. ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences 2, 279–284.
Van Den Brink, L., Stoter, J., Zlatanova, S., 2013. UML -Based Approach to Developing a CityGML Application Domain Extension. Transactions in GIS 17, 920–942. https://doi.org/10.1111/tgis.12026
van den Brink, L., Stoter, J., Zlatanova, S., 2013. Establishing a national standard for 3D topographic data compliant to CityGML. International Journal of Geographical Information Science 27, 92–113.
Van Oosterom, P., 2013. Research and development in 3D cadastres. Computers, environment and urban systems 40, 1–6.
Widl, E., Agugiaro, G., Peters-Anders, J., 2021. Linking semantic 3D city models with domain-specific simulation tools for the planning and validation of energy applications at district level. Sustainability 13, 8782.
Wu, J., 2021. Automatic building permits checks by means of 3D city models (matser thesis). the Delft University of Technology.
Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., Kolbe, T.H., 2018. 3DCityDB-a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based

on CityGML. Open Geospatial Data, Software and Standards 3, 1–26. https://doi.org/10.1186/s40965-018-0046-7
Yurtsever, E., Lambert, J., Carballo, A., Takeda, K., 2020. A survey of autonomous driving: Common practices and emerging technologies. IEEE access 8, 58443–58469.
Zhang, J., El-Gohary, N.M., 2016. Semantic NLP-Based Information Extraction from Construction Regulatory Documents for Automated Compliance Checking. J. Comput. Civ. Eng. 30, 04015014. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000346
Zhang, M.Y., 2019. Automated building compliance checking: emphasis on building placement and parking availability. Student thesis series INES.
Zhang, Z., Ma, L., Broyd, T., 2023. Rule capture of automated compliance checking of building requirements: a review. Proceedings of the Institution of Civil Engineers - Smart Infrastructure and Construction 176, 224–238. https://doi.org/10.1680/jsmic.23.00005

Appendix

Appendix A. Regulation texts

A. 1 The relevant regulations from GaStellV (original text and English translation).

The original regulation (German)	**The original regulation**
Teil I Allgemeine Vorschriften	Part I General Provision
§ 1 Begriffe und allgemeine Anforderungen	**§ 1 Concept and general requirements**
(7) Garagen sind mit einer Nutzfläche a) bis 100 m^2 Kleingaragen, b) über 100 m^2 und bis 1000 m^2 Mittelgaragen, c) über 1000 m^2 Großgaragen. Automatische Garagen mit mehr als 50 Einstellplätzen gelten als Großgaragen.	(7) Garages with a usable area are a) up to 100 m2 small garages, b) over 100 m2 and up to 1000 m2 medium garages, c) over 1000 m2 large garages. Automatic garages with more than 50 parking slots are considered large garages.
Teil II Bauvorschriften	Part II building regulations
§ 2 Zu- und Abfahrten	**§ 2 Entrances and exits**
(1) [1]Zwischen Garagen und ögewerbtlichen Verkehrsflächen müssen Zu- und Abfahrten von mindestens 3 m Länge vorhanden sein. [2]Abweichungen können gestattet werden, wenn wegen der Sicht auf die öffentliche Verkehrsfläche keine Bedenken bestehen.	(1) [1]Between garages and public traffic areas, entrances and exits of at least 3 m in length must be provided. [2]Deviations may be permitted if there are no concerns due to the visibility onto the public traffic area.
(2) Vor den die freie Zufahrt zur Garage zeitweilig hindernden Anlagen, wie Schranken oder Tore, ist ein Stauraum für wartende Kraftfahrzeuge vorzusehen, wenn dies wegen der Sicherheit und Leichtigkeit des Verkehrs erforderlich ist.	(2) A storage area for waiting motor vehicles shall be provided in front of installations temporarily obstructing free access to the garage, such as barriers or gates, if necessary for the safety and ease of traffic.
(3) [1]Die Fahrbahnen von Zu- und Abfahrten vor Mittel- und Großgaragen müssen mindestens 2,75 m breit sein; der Halbmesser des inneren Fahrbahnrandes muß mindestens 5 m betragen. [2]Für Fahrbahnen im Bereich von Zu- und Abfahrtssperren genügt eine Breite von 2,30 m. [3]Breitere Fahrbahnen sind in Kurven mit Innenhalbmessern von weniger als 10 m vorzusehen, wenn dies wegen der Verkehrssicherheit erforderlich ist.	(3) [1]The driveways of entrances and exits to medium and large garages must be at least 2.75 m wide; the radius of the inner edge of the driveways must be at least 5 m. [2]A width of 2.30 m is sufficient for driveways in the area of entrance and exit barriers. [3]Wider driveways shall be provided in curves with inner radii of less than 10 m if necessary for road safety.
(4) Großgaragen müssen getrennte Fahrbahnen für Zu- und Abfahrten haben.	(4) Large garages must have separate lanes for entrances and exits.
(5) [1]Vor Großgaragen ist neben den Fahrbahnen der Zu- und Abfahrten ein mindestens 0,80 m breiter Gehweg erforderlich, soweit nicht für Fußgänger besondere Fußwege vorhanden sind. [2]Der Gehweg muß gegenüber der Fahrbahn erhöht oder verkehrssicher abgegrenzt werden.	(5) [1]In front of large garages, a footpath at least 0.80 m wide is required alongside the driveways of the entrances and exits, unless special footpaths are provided for pedestrians. [2]The footpath must be elevated from the roadway or demarcated in accordance with traffic rules.
(6) In den Fällen der Absätze 3 bis 5 sind die Dacheinstellplätze und die dazugehörigen Verkehrsflächen der Nutzfläche zuzurechnen.	(6) In the cases of Sub-sections 3 to 5, the rooftop parking slots, and the associated traffic areas shall be included in the usable area.
§ 3 Rampen	**§ 3 Ramps**
(1) [1]Rampen von Mittel- und Großgaragen dürfen nicht mehr als 15 v.H., bei gewendelten Rampenteilen bezogen auf den inneren Fahrbahnrand, geneigt sein. [2]Die Breite der Fahrbahnen auf diesen Rampen muß mindestens 2,75 m, in gewendelten Rampenbereichen mindestens 3,50 m betragen. [3]Gewendelte Rampenteile müssen eine ausreichende Querneigung haben. [4]Der Halbmesser des inneren Fahrbahnrandes muß mindestens 5 m betragen.	(1) [1]Ramps in medium and large garages cannot be sloped by more than 15 %, in the case of spiral ramp sections in relation to the inner edge of the driveway. [2]The width of the driveway on these ramps must be at least 2.75 m and at least 3.50 m in the case of spiral ramp sections. [3]Spiral ramp sections must have adequate superelevation. [4]The radius of the inner edge of the driveway must be at least 5 m.

<table>
<tr><th>The original regulation (German)</th><th>The original regulation</th></tr>
<tr><td>(2) Zwischen öffentlicher Verkehrsfläche und einer Rampe mit mehr als 10 v.H. Neigung muß eine geringer geneigte Fläche mit weniger als 5 v.H. Neigung und von mindestens 3 m Länge liegen.</td><td>(2) Between the public traffic area and a ramp with a slope of more than 10 %, there must be a less inclined surface with a slope of less than 5 % and a minimum length of 3 meters.</td></tr>
<tr><td>(3) [1]In Großgaragen müssen Rampen, die auch zum Begehen bestimmt sind, einen mindestens 0,80 m breiten Gehweg haben, der gegenüber der Fahrbahn erhöht oder verkehrssicher abgegrenzt ist. [2]An Rampen, die von Personen nicht begangen werden dürfen, ist auf das Verbot hinzuweisen.</td><td>(3) [1]In large garages, ramps intended for pedestrian use must have a footpath at least 0.80 m wide, which is raised above the roadway or is demarcated in a way that is safe for traffic. [2]The prohibition shall be signposted at ramps not to be used by pedestrian.</td></tr>
<tr><td>§ 4 Einstellplätze und Fahrgassen</td><td>§ 4 Parking slots and driveways</td></tr>
<tr><td>(1) [1]Ein notwendiger Einstellplatz muß mindestens 5 m lang sein. [2]Die lichte Breite eines Einstellplatzes muß mindestens betragen
a) 2,30 m, wenn keine Längsseite,
b) 2,40 m, wenn eine Längsseite,
c) 2,50 m, wenn jede Längsseite des Einstellplatzes durch Wände, Stützen, andere Bauteile oder Einrichtungen begrenzt ist,
d) 3,50 m, wenn der Einstellplatz für Behinderte bestimmt ist.</td><td>(1) [1]A required parking slot must be at least 5 m long. [2] The clear width of a parking slot must be at least
a) 2,30 m, if there is no long side,
b) 2,40 m, if there is one long side,
c) 2,50 m, if each long side of the parking slot is delimited by walls, pillars, other components or installations,
d) 3,50 m, if the parking slot is intended for disabled users.</td></tr>
<tr><td>(2) Fahrgassen müssen, soweit sie unmittelbar der Zu- oder Abfahrt von Einstellplätzen dienen, hinsichtlich ihrer Breite mindestens die Anforderungen der folgenden Tabelle erfüllen; Zwischenwerte sind geradlinig einzuschalten:
<table>
<tr><th>Anordnung der Einstellplätze zur Fahrgasse</th><th colspan="3">Erforderliche Fahrgassenbreite (in m) bei einer Einstellplatzbreite von</th></tr>
<tr><th></th><th>2,30 m</th><th>2,40 m</th><th>2,50 m</th></tr>
<tr><td>90°</td><td>6,50</td><td>6,25</td><td>6,00</td></tr>
<tr><td>60°</td><td>4,50</td><td>4,25</td><td>4,00</td></tr>
<tr><td>45°</td><td>3,50</td><td>3,25</td><td>3,00</td></tr>
</table></td><td>(2) Driveways, insofar as they directly serve the access to or from parking slots, must at least meet the requirements of the following table concerning their width; intermediate values are to be interpolated linearly:
<table>
<tr><th>Arrangement of the parking slots to the driveway</th><th colspan="3">Required width of driveway (in m) for a parking slot width of:</th></tr>
<tr><th></th><th>2,30 m</th><th>2,40 m</th><th>2,50 m</th></tr>
<tr><td>90°</td><td>6,50</td><td>6,25</td><td>6,00</td></tr>
<tr><td>60°</td><td>4,50</td><td>4,25</td><td>4,00</td></tr>
<tr><td>45°</td><td>3,50</td><td>3,25</td><td>3,00</td></tr>
</table></td></tr>
<tr><td>(3) Fahrgassen in Mittel- und Großgaragen müssen, soweit sie nicht unmittelbar der Zu- oder Abfahrt von Einstellplätzen dienen, mindestens 3 m, bei Gegenverkehr mindestens 5 m breit sein.</td><td>(3) Driveways in medium and large garages must be at least 3 m wide, if they do not directly serve the access to or from parking slots, and at least 5 m wide if there is two-way traffic.</td></tr>
<tr><td>(4) [1]Einstellplätze auf kraftbetriebenen Hebebühnen brauchen abweichend von Absatz 1 Nr. 1 bis 3 nur 2,30 m breit zu sein; die Fahrgassen müssen mindestens 8 m breit sein, wenn die Hebebühnen Fahrspuren haben oder beim Absenken in die Fahrgasse hineinragen. [2]Einstellplätze auf geneigten kraftbetriebenen Hebebühnen sind in allgemein zugänglichen Garagen nicht zulässig.</td><td>(4) [1]Parking slots on power-operated lifting platforms need only be 2.30 m wide in contrast to Para. 1 Nr a to c; the driveways must be at least 8 m wide if the lifting platforms have driving lanes or extend into the driveway when lowered. [2]Parking slots on sloping power-operated lifting platforms are not permitted in generally usable / publicly accessible garages.</td></tr>
<tr><td>(5) [1]Einstellplätze auf horizontal verschiebbaren Plattformen sind in Fahrgassen zulässig, wenn
a) eine Breite der Fahrgasse von mindestens 2,75 m erhalten bleibt,
b) die Plattformen nicht vor kraftbetriebenen Hebebühnen angeordnet werden und
c) in Fahrgassen mit Gegenverkehr kein Durchgangsverkehr stattfindet.
Absatz 1 Sätze 1 und 2 gelten nicht für diese Plattformen.</td><td>(5) [1]Parking slots on horizontally movable platforms are permissible in driveways if
a) a width of the driveway is at least 2.75 m,
b) the platforms are not placed in front of power-operated lifting platforms, and
c) no through traffic occurs in two-way driveways.
Subsection 1, sentences 1 and 2, shall not apply to these platforms.</td></tr>
<tr><td>(6) [1]Die einzelnen Einstellplätze und die Fahrgassen sind mindestens durch Markierungen am Boden leicht erkennbar und dauerhaft gegeneinander abzugrenzen. [2]Dies gilt nicht für
a) Kleingaragen ohne Fahrgassen,
b) Einstellplätze auf kraftbetriebenen Hebebühnen,
c) Einstellplätze auf horizontal verschiebbaren Plattformen.
[3]Mittel- und Großgaragen müssen in jedem Geschoß leicht erkennbare und dauerhafte Hinweise auf Fahrtrichtungen und Ausfahrten haben</td><td>(6) [1]The individual parking slots and the driveways must be easily identifiable and permanently demarcated from each other at least by markings on the ground. [2]This shall not apply to
a) small garages without driveways,
a) parking slots on power-operated lifting platforms,
b) parking slots on horizontally movable platforms.</td></tr>
</table>

The original regulation (German)	The original regulation
	[3]Medium and large garages must have easily identifiable and permanent signs for driving directions and exits on all storeys.
(8) Die Absätze 1 bis 6 gelten nicht für automatische Garagen.	(8) Para. (1) to (6) do not apply to automatic garages.
§ 5 Lichte Höhe	**§ 5 Clear height**
[1]Garagen müssen in zum Begehen bestimmten Bereichen, auch unter Unterzügen, Lüftungsleitungen und sonstigen Bauteilen eine lichte Höhe von mindestens 2 m haben. [2]Dies gilt nicht für kraftbetriebene Hebebühnen.	[1]Garages must have a clear height of at least 2 m in areas designed for traffic, including under beams, ventilation ducts, and other structural components. [2] This does not apply to power-operated lifting platforms.
Teil V Notwendige Stellplätze	Teil V Necessary parking slots
§ 20 Notwendige Stellplätze	**§ 20 Necessary parking slots**
[1]Die Zahl der notwendigen Stellplätze im Sinn des Art. 47 Abs. 1 Satz 1, Abs. 2 Satz 1 BayBO bemisst sich nach der Anlage. [2]Ist eine Nutzung nicht in der Anlage aufgeführt, ist die Zahl der notwendigen Stellplätze in Anlehnung an eine oder mehrere vergleichbare Nutzungen zu ermitteln.	[1]The number of required parking slots as per Art. 47 Para. 1 Sentence 1, Para. 2 Sentence 1 of the BayBO shall be determined by the appendix. [2]If a use is not listed in the appendix, the number of parking slots required must be determined by referencing one or more comparable uses.

*§4 (8) is deleted.

A. 2 The necessary parking slots according to the traffic source in GaStS (original text).

Nr.	Verkehrsquelle	Stellplatzzahl (St)	Hiervon für Besucher in v. H.
1.0	**Wohngebäude**		
1.1	Ein- und Zweifamilienwohnhäuser (einschl. Reihenhäuser und Doppelhaus-hälften)	2St/WE; ein gefangener Stellplatz möglich, wenn dieser unmittelbar von einer gewidmeten öffentlichen Verkehrsfläche angefahren wird	
1.2	Mehrfamilienwohnhäuser je Wohnung bis 40 m² WF bis 120 m² WF über 120 m² WF	 1,2 St/WE 1,5 St/WE 2 St/WE	10 %
1.3	Wohnungen für Studierende, Auszubildende	0,5 St/Wohnung*	
1.4	Wohnheime für Pflegepersonal, Arbeitnehmer/Innen	1 St/drei Betten, mind. 3 St**	10 %
1.5	Wohnheime für Studierende	1 St/zwei Betten**	
1.6	Öffentlich geförderte Wohnungen	1,0 St/WE	
2.0	**Verkaufsstätten**		
2.1	Grundsätzlich: Läden, Waren- und Geschäftshäuser	1 St/30 m² Verkaufsnutzfläche	75 %
2.2	Einkaufszentren	1 St/15 m² Verkaufsnutzfläche	90 %
2.3	SB-Warenhäuser und -Fachmärkte, Verbrauchermärkte sowie Lebensmitteldiscountmärkte	1 St/15 m² Verkaufsnutzfläche	90 %
2.4	Großflächige Möbelfachmärkte	1 St/60 m² Verkaufs-/Ausstellungsnutzfläche	90 %
2.5	Großflächige Teppichfachmärkte	1 St/40 m² Verkaufs-/Ausstellungsnutzfläche	90 %
3.0	**Gaststätten und Beherbergungsbetriebe**		
3.1	Gaststätten	1 St/10 m² Nettogastraumfläche	90 %
3.2	Gaststätten mit Biergärten bzw. sonstigen Freischankflächen	wie vor, jedoch 1 weiterer St/20 m² Freischankfläche, soweit diese die Nettogastraumfläche übersteigt	90 %
3.3	Biergärten bzw. sonstige Freischankflächen	1 St/20 m² Freischankfläche	95 %
3.4	Hotels, Pensionen und sonstige Beherbergungsbetriebe	1 St/3 Betten für zugehörige, nicht ausschließlich für Hotelgäste genutzte Gasträume, Zuschlag nach Nr. 3.1	

Nr.	Verkehrsquelle	Stellplatzzahl (St)	Hiervon für Besucher in v. H.
3.5	Boardinghouse	1 St/Zimmer	
4.0	**Vergnügungsstätten**		
4.1	Spielhallen und Spielotheken	1 St/10 m² Nettonutzfläche,	90 %
4.2	Diskotheken	1 St/4 m² Nettonutzfläche	90 %
5.0	**Büro-, Verwaltungs-, Geschäfts- und Praxisräumen**		
5.1	Büro- und Verwaltungsräume	1 St/30 m² Hauptnutzfläche, jedoch mindestens 1 St	20 %
5.2	Räume mit erheblichem Besucherverkehr (Schalter-, Abfertigungs- und Beratungs-räume, Banken, Arztpraxen usw.)	1 St/20 m² Hauptnutzfläche, jedoch mindestens 3 St	75 %
5.3	Bahnhöfe	1 St je 3 Pendler im Tagesmittel zusätzlich zu 5.1 und 5.2	90 %
6.0	**Sonstiges**		
6.1	Videotheken - ohne Vorführung - mit Vorführung	 1 St/30 m² Nettonutzfläche 1 St/20 m² Nettonutzfläche	 80 % 90 %
6.2	Fitneßcenter	1 St/20 m² Nettonutzfläche	90 %
6.3	Go-Kart-Bahnen	1 St/50 m² Kartbahn-Nutzfläche	90 %
6.4	Museen	1 St/40 m² Ausstellungsfläche	95 %
6.5	Auto-Gebrauchtwarenmärkte	1 St/150 m² Verkaufs-/Ausstellungsnutzfläche	95 %
6.6	Schulen für Erwachsenenbildung	1 St./3 Schüler älter als 18 Jahre	
6.6.1	wie 6.6, jedoch innerhalb der Altstadt	1 St./5 Schüler älter als 18 Jahre	
6.7	Moscheen und sonstige kirchliche Einrichtungen	1 St./10 Besucher	
7.0	**Vergünstigungen in der Altstadt**		
7.1	Die ermittelte Stellplatzzahl ist innerhalb des Stadtmauerringes um 50% zu reduzieren, das Ergebnis auf ganze Stellplätze abzurunden.		

* Die Wohnungen müssen auf Dauer zur Benutzung durch den Personenkreis bestimmt sein. Eine diesbezügliche rechtliche Sicherung durch Eintragung einer beschränkt persönlichen Dienstbarkeit im Grundbuch zugunsten der Stadt Ingolstadt ist erforderlich.

** Abgrenzungskriterium zu 1.3: keine eigenständigen abgeschlossenen Wohneinheiten (z.B. keine Kochgelegenheit in der Einheit, Gemeinschaftsraum usw.) Zweckbestimmung und Sicherung wie Wohnungen (siehe oben).

A. 3 The necessary parking slots according to the traffic source in GaStellV (original text).

Nr.	Verkehrsquelle	Zahl der Stellplätze	hiervon in Vomhundertsätzen für Besucher
1.	**Wohngebäude**		
1.1	Einfamilienhäuser	1 Stellplatz je Wohnung	–
1.2	Mehrfamilienhäuser und sonstige Gebäude mit Wohnungen	1 Stellplatz je Wohnung	10
1.3	Gebäude mit Altenwohnungen	0,2 Stellplätze je Wohnung	20
1.4	Wochenend- und Ferienhäuser	1 Stellplatz je Wohnung	–
1.5	Kinder-, Schüler- und Jugendwohnheime	1 Stellplatz je 20 Betten, mindestens 2 Stellplätze	75
1.6	Studentenwohnheime	1 Stellplatz je 5 Betten	10
1.7	Schwestern-/ Pflegerwohnheime	1 Stellplatz je 2 Betten, mindestens 3 Stellplätze	10

Nr.	Verkehrsquelle	Zahl der Stellplätze	hiervon in Vomhundertsätzen für Besucher
1.8	Arbeitnehmerwohnheime	1 Stellplatz je 4 Betten, mindestens 3 Stellplätze	20
1.9	Altenwohnheime	1 Stellplatz je 15 Betten, mindestens 3 Stellplätze	50
1.10	Altenheime, Langzeit- und Kurzzeitpflegeheime	1 Stellplatz je 12 Betten bzw. Pflegeplätze, mindestens 3 Stellplätze	50
1.11	Tagespflegeeinrichtungen	1 Stellplatz je 12 Pflegeplätze, mindestens 3 Stellplätze	50
1.12	Obdachlosenheime, Gemeinschaftsunterkünfte für Leistungsberechtigte nach dem Asylbewerberleistungsgesetz	1 Stellplatz je 30 Betten, mindestens 3 Stellplätze	10
2.	**Gebäude mit Büro-, Verwaltungs- und Praxisräumen**		
2.1	Büro- und Verwaltungsräume allgemein	1 Stellplatz je 40 m^2 NF[1)]	20
2.2	Räume mit erheblichem Besucherverkehr (Schalter-, Abfertigungs oder Beratungsräume, Arztpraxen und dergl.)	1 Stellplatz, je 30 m^2 NF[1)], mindestens 3 Stellplätze	75
3.	**Verkaufsstätten**		
3.1	Läden	1 Stellplatz je 40 m^2 NF (V)[2)], mindestens 2 Stellplätze je Laden	75
3.2	Waren- und Geschäftshäuser (einschließlich Einkaufszentren, großflächigen Einzelhandelsbetrieben)	1 Stellplatz je 40 m^2 NF (V)[2)]	75
4.	**Versammlungsstätten (außer Sportstätten), Kirchen**		
4.1	Versammlungsstätten von überörtlicher Bedeutung (z.B. Theater, Konzerthäuser, Mehrzweckhallen)	1 Stellplatz je 5 Sitzplätze	90
4.2	Sonstige Versammlungsstätten (z.B. Lichtspieltheater, Schulaulen, Vortragssäle)	1 Stellplatz je 10 Sitzplätze	90
4.3	Gemeindekirchen	1 Stellplatz je 30 Sitzplätze	90
4.4	Kirchen von überörtlicher Bedeutung	1 Stellplatz je 20 Sitzplätze	90
5.	**Sportstätten**		
5.1	Sportplätze ohne Besucherplätze (z.B. Trainingsplätze)	1 Stellplatz je 300 m^2 Sportfläche	–
5.2	Sportplätze und Sportstadien mit Besucherplätzen	1 Stellplatz je 300 m^2 Sportfläche, zusätzlich 1 Stellplatz je 15 Besucherplätze	–
5.3	Turn- und Sporthallen ohne Besucherplätze	1 Stellplatz je 50 m^2 Hallenflächen	–
5.4	Turn- und Sporthallen mit Besucherplätzen	1 Stellplatz je 50 m^2 Hallenfläche; zusätzlich 1 Stellplatz je 15 Besucherplätze	–
5.5	Freibäder und Freiluftbäder	1 Stellplatz je 300 m^2 Grundstücksfläche	–
5.6	Hallenbäder ohne Besucherplätze	1 Stellplatz je 10 Kleiderablagen	–
5.7	Hallenbäder mit Besucherplätzen	1 Stellplatz je 10 Kleiderablagen, zusätzlich 1 Stellplatz je 15 Besucherplätze	–
5.8	Tennisplätze ohne Besucherplätze	2 Stellplätze je Spielfeld	–
5.9	Tennisplätze mit Besucherplätzen	2 Stellplätze je Spielfeld, zusätzlich 1 Stellplatz je 15 Besucherplätze	–
5.10	Squashanlagen	2 Stellplätze je Court	–
5.11	Minigolfplätze	6 Stellplätze je Minigolfanlage	–
5.12	Kegel-, Bowlingbahnen	4 Stellplätze je Bahn	–
5.13	Bootshäuser und Bootsliegeplätze	1 Stellplatz je 5 Boote	–
5.14	Fitnesscenter	1 Stellplatz je 40 m^2 Sportfläche	–
6.	**Gaststätten und Beherbergungsbetriebe**		
6.1	Gaststätten	1 Stellplatz je 10 m^2 Gastfläche	75
6.2	Spiel- und Automatenhallen, Billard-Salons, sonst. Vergnügungsstätten	1 Stellplatz je 20 m^2 NF[1)], mind. 3 Stellplätze	90
6.3	Hotels, Pensionen, Kurheime und andere Beherbergungsbetriebe	1 Stellplatz je 6 Betten, bei Restaurationsbetrieb Zuschlag nach 6.1 oder 6.2	75
6.4	Jugendherbergen	1 Stellplatz je 15 Betten	75

Nr.	Verkehrsquelle	Zahl der Stellplätze	hiervon in Vomhundertsätzen für Besucher
7.	**Krankenanstalten**		
7.1	Krankenanstalten von überörtlicher Bedeutung	1 Stellplatz je 4 Betten	60
7.2	Krankenanstalten von örtlicher Bedeutung	1 Stellplatz je 6 Betten	60
7.3	Sanatorien, Kuranstalten, Anstalten für langfristig Kranke	1 Stellplatz je 4 Betten	25
7.4	Ambulanzen	1 Stellplatz je 30 m^2 NF[1)], mindestens 3 Stellplätze	75
8.	**Schulen, Einrichtungen der Jugendförderung**		
8.1	Grundschulen, Schulen für Lernbehinderte	1 Stellplatz je Klasse	–
8.2	Hauptschulen, sonstige allgemeinbildende Schulen, Berufsschulen, Berufsfachschulen	1 Stellplatz je Klasse, zusätzlich 1 Stellplatz je 10 Schüler über 18 Jahre	10
8.3	Sonderschulen für Behinderte	1 Stellplatz je 15 Schüler	–
8.4	Hochschulen	1 Stellplatz je 10 Studierende	–
8.5	Tageseinrichtungen für Kinder	1 Stellplatz je 30 Kinder, mindestens 2 Stellplätze	–
8.6	Jugendfreizeitheime und dergl.	1 Stellplatz je 15 Besucherplätze	–
8.7	Berufsbildungswerke, Ausbildungswerkstätten und dergl.	1 Stellplatz je 10 Auszubildende	–
9.	**Gewerbliche Anlagen**		
9.1	Handwerks- und Industriebetriebe	1 Stellplatz je 70 m^2 NF[1)] oder je 3 Beschäftigte	10
9.2	Lagerräume, -plätze, Ausstellungs-, Verkaufsplätze	1 Stellplatz je 100 m^2 NF[1)] oder je 3 Beschäftigte	–
9.3	Kraftfahrzeugwerkstätten	6 Stellplätze je Wartungs- oder Reparaturstand	–
9.4	Tankstellen	Bei Einkaufsmöglichkeit über Tankstellenbedarf hinaus: Zuschlag nach 3.1 (ohne Besucheranteil)	–
9.5	Automatische Kfz-Waschanlagen	5 Stellplätze je Waschanlage[3)]	–
10.	**Verschiedenes**		
10.1	Kleingartenanlagen	1 Stellplatz je 3 Kleingärten	–
10.2	Friedhöfe	1 Stellplatz je 1500 m^2 Grundstücksfläche, jedoch mindestens 10 Stellplätze	–

[1)] **[Amtl. Anm.:]** NF = Nutzfläche nach DIN 277 Teil 2

[2)] **[Amtl. Anm.:]** NF (V) = Verkaufsnutzfläche

[3)] **[Amtl. Anm.:]** Zusätzlich muss ein Stauraum für mindestens 10 Kraftfahrzeuge vorhanden sein.

Appendix B. Tables

B.1 The selected articles, relevant objects, and corresponding attributes.

The original regulation	Relevant object	Attribute
Part I General Provision		
§ 1 Concept and general requirements		
(7) Garages with a usable area are up to 100 m2 small garages, over 100 m2 and up to 1000 m2 medium garages, over 1000 m2 large garages.	ParkingFacility	Area Class
Automatic garages with more than 50 parking slots are considered large garages.	ParkingFacility	Type
Part II building regulations		
§ 2 Entrances and exits		
(1) [1]Between garages and public traffic areas, entrances and exits of at least 3 m in length must be provided. [2]Deviations may be permitted if there are no concerns due to the visibility onto the public traffic area.	Entrance/ Exit	Distance
(2) A storage area for waiting motor vehicles shall be provided in front of installations temporarily obstructing free access to the garage, such as barriers or gates, if necessary for the safety and ease of traffic.	Entrance/ Exit	ExtraSlot
(3) [1]The driveways of entrances and exits to medium and large garages must be at least 2.75 m wide;	Entrance/ Exit	Width
the radius of the inner edge of the driveways must be at least 5 m.	Entrance/ Exit	Radius
[2]A width of 2.30 m is sufficient for driveways in the area of entrance and exit barriers.	Entrance/ Exit	WidthBarriers
[3]Wider driveways shall be provided in curves with inner radii of less than 10 m if necessary for road safety.	Entrance/ Exit	MoreWidth
(4) Large garages must have separate lanes for entrances and exits.	Entrance/ Exit	Lane
(5) [1]In front of large garages, a footpath at least 0.80 m wide is required alongside the driveways of the entrances and exits, unless special footpaths are provided for pedestrians.	Footpath	Width
[2]The footpath must be elevated from the roadway or demarcated from it/ in accordance with traffic rules.	Footpath	IsRaised Sign
(6) In the cases of Sub-sections 3 to 5, the rooftop parking slots, and the associated traffic areas shall be included in the usable area.	ParkingFacility	Area Class
§ 3 Ramps		
(1) [1]Ramps in medium and large garages cannot be sloped by more than 15 %, in the case of spiral ramp sections in relation to the inner edge of the driveway.	Ramp	SprialSlope
[2]The width of the driveway on these ramps must be at least 2.75 m and at least 3.50 m in the case of spiral ramp sections.	Ramp	WidthSpiral
[3]Spiral ramp sections must have adequate superelevation.	Ramp	Superelevation
[4]The radius of the inner edge of the driveway must be at least 5 m.	Ramp	Radius
(2) Between the public traffic area and a ramp with a slope of more than 10 %, there must be a less inclined surface with a slope of less than 5 % and a minimum length of 3 meters.	Ramp	BufferArea
(3) [1]In large garages, ramps intended for pedestrian use must have a footpath at least 0.80 m wide,	Footpath	Width
which is raised above the roadway or is demarcated in a way that is safe for traffic.	Footpath	Sign IsRaised
[2]The prohibition shall be signposted at ramps not intended for pedestrian use.	Footpath	Sign
§ 4 Parking slots and driveways		
(1) [1]A required parking slot must be at least 5 m long.	ParkingSlot	Length
[2] The clear width of a parking slot must be at least a) 2,30 m, if there is no long side, b) 2,40 m, if there is one long side,	ParkingSlot	Width Accessible

The original regulation	Relevant object	Attribute
c) 2,50 m, if each long side of the parking slot is delimited by walls, pillars, other components or installations, d) 3,50 m, if the parking slot is intended for disabled users.		
(2) Driveways, insofar as they directly serve the access to or from parking slots, must at least meet the requirements of the following table concerning their width; intermediate values are to be interpolated linearly: Arrangement of the parking slots to the driveway / Required width of driveway (in m) for a parking slot width of: 2,30 m / 2,40 m / 2,50 m 90° / 6,50 / 6,25 / 6,00 60° / 4,50 / 4,25 / 4,00 45° / 3,50 / 3,25 / 3,00	Driveway ParkingSlot	Width Angle
(3) Driveways in medium and large garages must be at least 3 m wide, if they do not directly serve the access to or from parking slots, and at least 5 m wide if there is two-way traffic.	Driveway	Width Lane
(4) [1]Parking slots on power-operated lifting platforms need only be 2.30 m wide in contrast to Para. 1 Nr a to c;	ParkingSlot	Type Width
the driveways must be at least 8 m wide if the lifting platforms have driving lanes or extend into the driveway when lowered.	Driveway	Width
[2]Parking slots on sloping power-operated lifting platforms are not permitted in generally usable / publicly accessible garages.	ParkingSlot ParkingFacility	Type Usage
(5) [1]Parking slots on horizontally movable platforms are permissible in driveways if	ParkingSlot	Type
a) a width of the driveway is at least 2.75 m,	Driveway	Width
b) the platforms are not placed in front of power-operated lifting platforms, and	ParkingSlot	Type DistancetoElecPS
c) no through traffic occurs in two-way driveways. Subsection 1, sentences 1 and 2, shall not apply to these platforms.	Driveway	IsThroughTraffic
(6) [1]The individual parking slots and the driveways must be easily identifiable and permanently demarcated from each other at least by markings on the ground.	Driveway ParkingSlot	Sign Sign
[2] This shall not apply to a) small garages without driveways,	ParkingFacility	Area Class
b) parking slots on power-operated lifting platforms,	ParkingSlot	Type
c) parking slots on horizontally movable platforms.	ParkingSlot	Type
[3]Medium and large garages must have easily identifiable and permanent signs for driving directions and exits on all storeys.	ParkingFacility Driveway	Area Sign
(8) Para. (1) to (6) do not apply to automatic garages.	ParkingFacility	Type
§ 5 Clear height		
[1]Garages must have a clear height of at least 2 m in areas designed for traffic, including under beams, ventilation ducts, and other structural components. [2]This does not apply to power-operated lifting platforms.	ParkingFacility	Height Type
Teil V Necessary parking slots		
§ 20 Necessary parking slots		
[1]The number of required parking slots as per Art. 47 Para. 1 Sentence 1, Para. 2 Sentence 1 of the BayBO shall be determined by the appendix. [2]If a use is not listed in the appendix, the number of parking slots required must be determined by referencing one or more comparable uses.	Building	minSlotNumber

B.2 Enumeration list of the function in Building mapping and extending, the bold text is extended parts

Regulation's Building List in GaStS (Nr. & Traffic source)	**Code list of the *_AbstractBuilding* attributes *function***
1. Wohngebäude	
1.1 Ein- und Zweifamilienwohnhäuser (einschl. Reihenhäuser und Doppelhaushälften)	1000 residential building
1.2 Mehrfamilienwohnhäuser je Wohnung bis 40 m² WF bis 120 m² WF über 120 m² WF	1010 tenement
1.3 Wohnungen für Studierende, Auszubildende	**9010 residential building for students and trainees**
1.4 Wohnheime für Pflegepersonal, Arbeitnehmer/Innen	**9020 nurses' dormitory** **9030 workers' dormitory**
1.5 Wohnheime für Studierende	**9040 students' domitory**
1.6 Öffentlich geförderte Wohnungen	**9050 publicly subsidized home**
2. Verkaufsstätten	
2.1 Grundsätzlich: Läden, Waren- und Geschäftshäuser	1180 kiosk 1160 department store
2.2 Einkaufszentren	1170 shopping centre
2.3 SB-Warenhäuser und -Fachmärkte, Verbrauchermärkte sowie Lebensmitteldiscountmärkte	**9060 hypermarkets and speciality shops** **9070 discount food store**
2.4 Großflächige Möbelfachmärkte	**9080 large-scale furniture store**
2.5 Großflächige Teppichfachmärkte	**9090 large-scale carpet speciality store**
3. Gaststätten und Beherbergungsbetriebe	
3.1 Gaststätten	1240 restaurant
3.2 Gaststätten mit Biergärten bzw. sonstigen Freischankflächen	1240 restaurant
3.3 Biergärten bzw. sonstige Freischankflächen	-
3.4 Hotels, Pensionen und sonstige Beherbergungsbetriebe	1210 hotel
3.5 Boardinghouse	**9110 bordinghouse**
4. Vergnügungsstätten	
4.1 Spielhallen und Spielotheken	1300 casino
4.2 Diskotheken	**9120 discotheque**
5. Büro-, Verwaltungs-, Geschäfts- und Praxisräumen	
5.1 Büro- und Verwaltungsräume	1120 office building 1150 business building 1970 administration building 1990 guildhall
5.2 Räume mit erheblichem Besucherverkehr (Schalter-, Abfertigungs- und Beratungs- räume, Banken, Arztpraxen usw.)	1330 health centre or outpatients clinic 1190 pharmacy 2030 embassy or consulate 1960 public building
5.3 Bahnhöfe	-
6. Sonstiges	
6.1 Videotheken - ohne Vorführung - mit Vorführung	**9130 video store**
6.2 Fitneßcenter	**9140 fitness centre**
6.3 Go-Kart-Bahnen	-
6.4 Museen	2160 museum
6.5 Auto-Gebrauchtwarenmärkte	**9150 second-hand car market**
6.6 Schulen für Erwachsenenbildung	**9160 adult education school**
6.6.1 wie 6.6, jedoch innerhalb der Altstadt	
6.7 Moscheen und sonstige kirchliche Einrichtungen	2210 religious building 2220 church 2230 synagogue 2240 chapel

	2270 mosque
7. Vergünstigungen in der Altstadt	
7.1 Die ermittelte Stellplatzzahl ist innerhalb des Stadtmauerringes um 50% zu reduzieren, das Ergebnis auf ganze Stellplätze abzurunden	-

B.3 Enumeration list of the function in Building mapping and extending, the bold text is extended parts, GaStS means it is mentioned above in B.2

Regulation's Building List in GaStellV (Nr. & Traffic source)	**Code list of the _*AbstractBuilding* attributes *function***
1. Wohngebäude	
1.1 Einfamilienhäuser	**GaStS**
1.2 Mehrfamilienhäuser und sonstige Gebäude mit Wohnungen	1010 tenement
1.3 Gebäude mit Altenwohnungen	**9170 the elderly apartment**
1.4 Wochenend- und Ferienhäuser	1100 holiday house
1.5 Kinder-, Schüler- und Jugendwohnheime	**9180 kids and youth dormitory**
1.6 Studentenwohnheime	**GaStS**
1.7 Schwestern-/ Pflegerwohnheime	**GaStS**
1.8 Arbeitnehmerwohnheime	**GaStS**
1.9 Altenwohnheime	2360 seniors centre
1.10 Altenheime, Langzeit- und Kurzzeitpflegeheime	2320 healing centre or care home
1.11 Tagespflegeeinrichtungen	**9190 day care centre**
1.12 Obdachlosenheime, Gemeinschaftsunterkünfte für Leistungsberechtigte nach dem Asylbewerberleistungsgesetz	2370 homeless shelter 2390 asylum seekers home
2. Gebäude mit Büro-, Verwaltungs- und Praxisräumen	
2.1 Büro- und Verwaltungsräume allgemein	**GaStS**
2.2 Räume mit erheblichem Besucherverkehr (Schalter-, Abfertigungs oder Beratungsräume, Arztpraxen und dergl.)	**GaStS**
3. Verkaufsstätten	
3.1 Läden	1180 kiosk
3.2 Waren- und Geschäftshäuser (einschließlich Einkaufszentren, großflächigen Einzelhandelsbetrieben)	**GaStS**
4. Versammlungsstätten (außer Sportstätten), Kirchen	
4.1 Versammlungsstätten von überörtlicher Bedeutung (z.B. Theater, Konzerthäuser, Mehrzweckhallen)	2140 theatre or opera 2150 concert building
4.2 Sonstige Versammlungsstätten (z.B. Lichtspieltheater, Schulaulen, Vortragssäle)	1280 cinema 2180 activity building
4.3 Gemeindekirchen	**GaStS**
4.4 Kirchen von überörtlicher Bedeutung	**GaStS**
5. Sportstätten	
5.1 Sportplätze ohne Besucherplätze	2570 building for sports field
5.2 Sportplätze und Sportstadien mit Besucherplätzen	2550 building for sport purposes 2570 building for sports field
5.3 Turn- und Sporthallen ohne Besucherplätze	2560 sports hall
5.4 Turn- und Sporthallen mit Besucherplätzen	2560 sports hall
5.5 Freibäder und Freiluftbäder	2580 swimming baths
5.6 Hallenbäder ohne Besucherplätze	2590 indoor swimming pool
5.7 Hallenbäder mit Besucherplätzen	2590 indoor swimming pool
5.8 Tennisplätze ohne Besucherplätze	2560 sports hall
5.9 Tennisplätze mit Besucherplätzen	2560 sports hall
5.10 Squashanlagen	2570 building for sports field
5.11 Minigolfplätze	**9200 sports fields for mini golf**
5.12 Kegel-, Bowlingbahnen	1290 bowling alley
5.13 Bootshäuser und Bootsliegeplätze	1590 boathouse
5.14 Fitnesscenter	**GaStS**
6. Gaststätten und Beherbergungsbetriebe	
6.1 Gaststätten	**GaStS**
6.2 Spiel- und Automatenhallen, Billard-Salons, sonst. Vergnügungsstätten	

Regulation's Building List in GaStellV (Nr. & Traffic source)	**Code list of the _*AbstractBuilding* attributes *function***
6.3 Hotels, Pensionen, Kurheime und andere Beherbergungsbetriebe	**GaStS**
6.4 Jugendherbergen	1220 youth hostel
7. Krankenanstalten	
7.1 Krankenanstalten von überörtlicher Bedeutung	2310 hospital
7.2 Krankenanstalten von örtlicher Bedeutung	2300 building for health care
7.3 Sanatorien, Kuranstalten, Anstalten für langfristig Kranke	2600 sanatorium
7.4 Ambulanzen	2330 health centre or outpatients clinic
8. Schulen, Einrichtungen der Jugendförderung	
8.1 Grundschulen, Schulen für Lernbehinderte	**9210 special school for the learning disabilities**
8.2 Hauptschulen, sonstige allgemeinbildende Schulen, Berufsschulen, Berufsfachschulen	2080 comprehensive school
8.3 Sonderschulen für Behinderte	**9220 special school**
8.4 Hochschulen	2100 college or university
8.5 Tageseinrichtungen für Kinder	2380 kindergarten or nursery
8.6 Jugendfreizeitheime und dergl.	2350 youth centre
8.7 Berufsbildungswerke, Ausbildungswerkstätten und dergl.	2090 vocational school
9. Gewerbliche Anlagen	
9.1 Handwerks- und Industriebetriebe	1310 industrial building 1320 factory
9.2 Lagerräume, -plätze, Ausstellungs-, Verkaufsplätze	1370 depot
9.3 Kraftfahrzeugwerkstätten	1330 workshop
9.4 Tankstellen	1340 petrol / gas station
9.5 Automatische Kfz-Waschanlagen	**9230 automatic car wash**
10. Verschiedenes	
10.1 Kleingartenanlagen	2620 green house 2630 botanical show house
10.2 Friedhöfe	2450 cemetery building

*The repeat part cpmpared with the enumeration list in GaStS is deleted.

B.4 Enumeration list of the Generic attribute in TrafficSpace and TrafficArea

Table B- 1 Enumeration list of the classOfPF in TrafficSpace

small
medium
large

Table B- 2 Enumeration list of the formOfPF in TrafficSpace

outdoor
indoor

Table B- 3 Enumeration list of the usageOfPF in TrafficSpace

public
privat

Table B- 4 Enumeration list of the typeOfPF in TrafficSpace

automatic
classical

Table B- 5 Enumeration list of the typeOfPS in TrafficArea

classical
power-operated lifting
horizontally-movable

Table B- 6 Enumeration list of the failureReason in TrafficArea

0	Compliant
1	Width too short
2	Length too short
3	Footpath is not raised enough
4	Radius too short
5	Slope too small
6	Insufficient superelevation
7	Incorrect number of lanes
8	Shortage of parking slots
9	Relevant dataset is missing