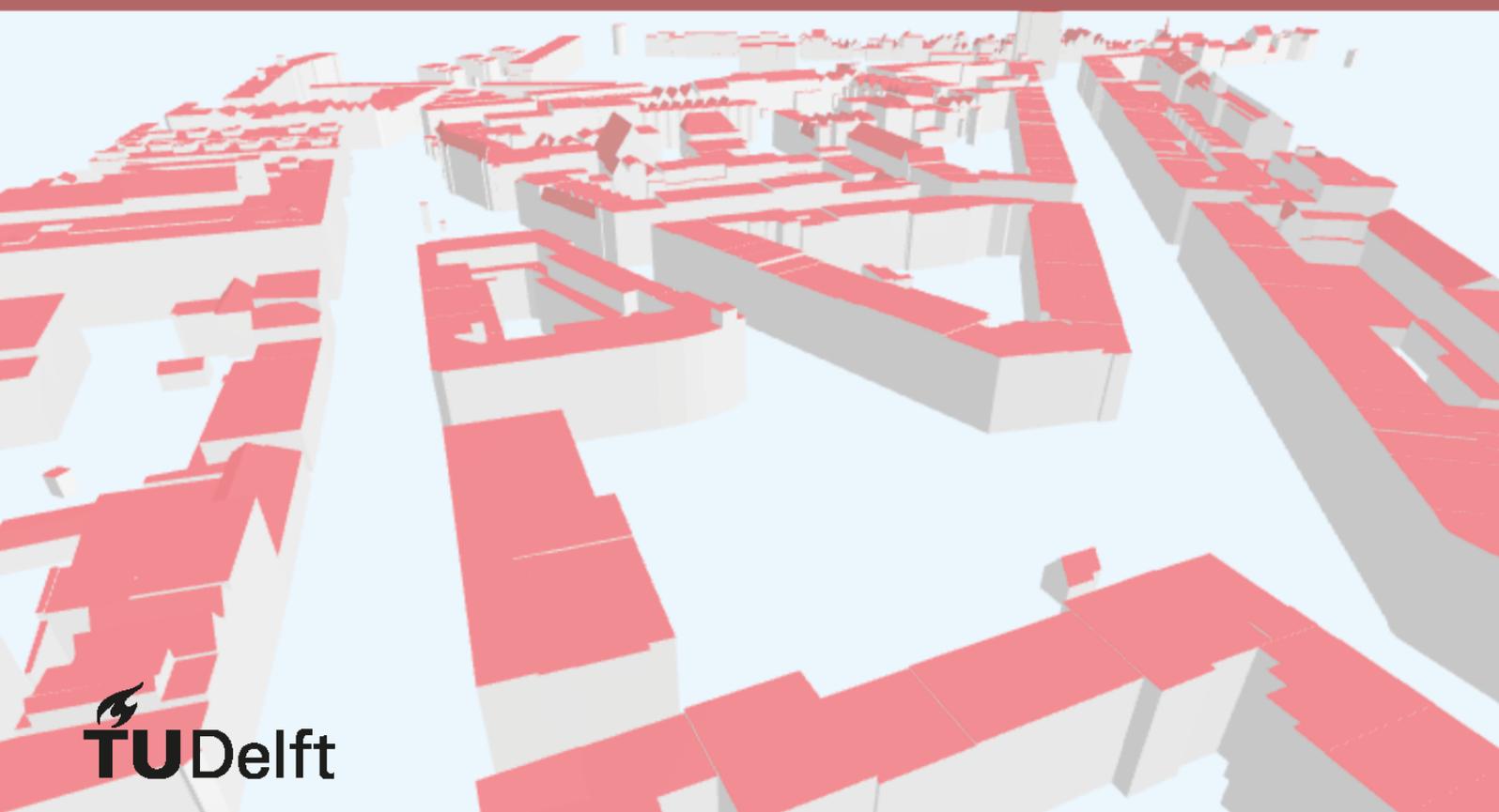




How could BIM support the digital permit process in the Netherlands?

Maarit Prusti
2022



HOW COULD BIM SUPPORT THE DIGITAL BUILDING PERMIT
PROCESS IN THE NETHERLANDS?

A thesis submitted to the Delft University of Technology in partial fulfillment
of the requirements for the degree of

Master of Science in Geomatics for the Built Environment

by

Maarit Prusti

June 2022

Maarit Prusti: *How could BIM support the digital building permit process in the Netherlands?* (2022)

© ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/4.0/>.

ISBN 999-99-9999-999-9

The work in this thesis was made in the:



3D geoinformation group
Department of Urbanism
Faculty of Architecture & the Built Environment
Delft University of Technology

Supervisors: Prof.dr. Jantien Stoter
Mr.dr. Hendrik Ploeger
Co-reader: Dr.ir. Frederika Welle Donker

ABSTRACT

The building permit process in the Netherlands is mostly digitalized, however, there are still some issues. A downside of information, the manual checking from the municipality, and the duration of the process are some of these issues. To overcome these issues information between 3D building design models, so-called Building Information Modeling (BIM)s, and 3D city models must be exchanged. To exchange interoperable information between BIMs and 3D city models is called integration. In this research, automatic rule checking is performed after the BIM encoded in Industry Foundation Classes (IFC) is converted to a 3D city model encoded in CityJSON.

Integration, however, is not as straightforward as it seems. Other researches have been carried out to perform a full integration from 3D city models encoded in CityGML to BIMs encoded in IFC. This is rather complex and so use cases are utilized. In most other researches using the building permit process as a use case, the automatic rule checking is performed in the BIM domain.

In this research, a conversion is performed from a BIM encoded in IFC to a 3D city model encoded in CityJSON. The first step in this research is to analyze land use plans and select the most often used rules. These rules are further analyzed on required information to check the rules. For both IFC and CityJSON, the required information for the rules representing the same information as entities in the standards are selected. In the next step, the input models are analyzed on the presence of the entities from the standards. Before the conversion is performed, it is determined which entity will be converted from which input model and whether or not additional information is needed. Finally, the conversion is performed. The 3D city model can be used for rule checking and satisfies the selected rules. The implications of this research are described for the digital building permit process as well as for the integration of the two domains. Guidelines to model correct BIMs for the digital building permit process and further integration are drafted.

In conclusion, the tool created in this research works successfully. Automatic rule checking on all the rules in land use plans is technically possible. In practice, automatic rule checking will most likely not take over soon, since rules are still written ambiguously, builders work with 2D drawings mostly, and the Environmental and Planning Act is soon to be established.

ACKNOWLEDGEMENTS

I would like to express my gratitude towards my supervisors, Jantien Stoter and Hendrik Ploeger. From the very start, you guided me through this process. In our weekly online meetings, you provided much knowledge, ideas, and experiences from your respective domains which are used in this research. Whenever I got stuck at some point in the process, you guided me to the solution. On top of that, I would like to thank my co-reader Frederika Welles Donker and delegate Lei Qu for their time and effort.

In addition, I would like to thank my roommates, friends of Geomatics, and other friends who continuously listened to my frustrations, supported me, and reassured me. Special thanks to my boyfriend for the joyful moments that kept me going on. Finally, I would like to thank my family, especially my mother who stood by me throughout the process.

CONTENTS

1	INTRODUCTION	1
1.1	Research objectives	2
1.2	Scope	3
1.3	Thesis outline	3
2	BACKGROUND AND RELATED LITERATURE	5
2.1	Background	5
2.1.1	Legal framework	5
2.1.2	BIM	7
2.1.3	Totaal Driedimensionaal	7
2.1.4	Digital building permit checks	8
2.2	Related literature	9
2.2.1	Digital building permit process	9
2.2.2	Full integration	9
2.2.3	Integration (use case: building permit process)	10
3	METHODOLOGY	12
3.1	Mapping the rules	12
3.1.1	Retrieving the land use plan	13
3.1.2	Selecting the rules	13
3.1.3	Determining the information	13
3.1.4	Linking the entities	13
3.2	Inspecting the input	14
3.3	Preparing the conversion	14
3.4	Performing the conversion	15
3.4.1	Data preparation	15
3.4.2	Validation	16
3.4.3	Conversion	16
3.4.4	Validation	18
3.4.5	Rule checking	18
3.5	Implications	19
3.6	Guidelines	19
4	IMPLEMENTATION	21
4.1	Tools	21
4.1.1	AutodeskViewer	21
4.1.2	Revit	21
4.1.3	IfcCheckingTool	21
4.1.4	MeshLab	21
4.1.5	val3dity	21
4.1.6	cjval	22
4.1.7	QGIS	22
4.1.8	ninja	22
4.2	Datasets	22
4.2.1	BIMs	22
4.2.2	SOR	23
4.2.3	3D Basisbestand Volledig	24
4.2.4	Peil	24
4.2.5	AHN4	24
5	EXPERIMENTS AND RESULTS	26
5.1	Mapping the regulations	26
5.1.1	The maximum building height	27
5.1.2	The maximum gutter height	28
5.1.3	The maximum height of other constructions	28

5.1.4	Required information	28
5.1.5	IFC	29
5.1.6	CityJSON	29
5.1.7	Summary	30
5.2	Inspecting the models	31
5.2.1	BIM	31
5.2.2	SOR	31
5.2.3	Summary	32
5.3	Preparing the conversion	32
5.3.1	Peil from 3D Basisvoorziening	32
5.3.2	Peil from dataset Rotterdam	33
5.3.3	Ground height from AHN4	33
5.3.4	Summary	33
5.4	Performing the conversion	33
5.4.1	Data preparation	33
5.4.2	Validation	34
5.4.3	Conversion	35
5.4.4	Validation	44
5.4.5	Rule checking	45
5.4.6	Summary	46
6	IMPLICATIONS	48
6.1	Building permit process	48
6.1.1	Transparency	48
6.1.2	Municipalities	49
6.1.3	Process	49
6.2	Integration	50
7	DISCUSSION	52
7.1	Parsing	52
7.1.1	Relative Placement	52
7.1.2	True North	52
7.2	Rounding	52
7.3	Missing link	53
7.4	Peil	53
7.5	Georeferencing	53
7.6	Reflection	55
7.6.1	Digitalization	55
7.6.2	Domain	55
7.7	Limitations of this research	56
7.7.1	Straight walls	56
7.7.2	Overlapping roof	56
7.7.3	Other building elements	56
7.7.4	More complex footprint	57
8	GUIDELINES	60
8.1	Current guidelines	60
8.2	Guidelines integration	60
8.2.1	Adding features in IFC	60
8.2.2	Georeferencing	61
8.3	Guidelines digital building permit process	61
8.3.1	IFC schema	61
8.3.2	Georeferencing	61
8.3.3	Join building elements	62
9	CONCLUSION AND FUTURE WORK	63
9.1	Overview of research	63
9.2	Conclusion	63
9.3	Future work	65

A	APPENDIX	71
A.1	Figures and tables	71
A.2	Summary of transcriptions	72
A.2.1	Municipality of Rotterdam (Rolf Jonker)	72
A.2.2	Municipality of Delft (Marianne Schoenmakers en Henry Rienstra)	72

LIST OF FIGURES

Figure 2.1	The procedure of the building permit [SAB, 2022].	6
Figure 3.1	General workflow	12
Figure 3.2	Workflow mapping the rules	13
Figure 3.3	Selected land use plan in online viewer	13
Figure 3.4	Simple.ifc	14
Figure 3.5	Workflow inspecting the models	14
Figure 3.6	Proposed workflow conversion	16
Figure 4.1	Simple.ifc	22
Figure 4.2	BIMs part one	23
Figure 4.3	BIMs part two	23
Figure 5.1	BIMs part one	26
Figure 5.2	BIMs part two	27
Figure 5.3	The ground height matters in checking the rules in this example. (1) the peil value, (2) ground height of the building, (3) maximum building height stated in the land use plan, (4) height of the building	28
Figure 5.4	Pin for Survey Point located in Revit on the location Schependomlaan	34
Figure 5.5	Building design rotated to True North for Schependomlaan	34
Figure 5.6	Walls before rotation and translation	36
Figure 5.7	Triangular faces after rotation and translation	37
Figure 5.8	Selecting the outer surfaces: (1) The surfaces of the BIM, (2) Removing horizontally oriented surfaces, (3) Find the vertex closest to bottom left corner and calculate the bounding box of the surface it belongs to, (4) Repeat for each outer surfaces, until all are found.	38
Figure 5.9	Exploded view of Simple.ifc	38
Figure 5.10	Upper row: if the horizontally oriented surfaces are included, this will lead to internal surfaces being added, bottom row: without the horizontally oriented surfaces, only the outer surfaces are added.	39
Figure 5.11	Orientation triangular faces: (1) Create viewpoint outside of bounding box , (2) Determine orientation triangular face nearest by the viewpoint, (3) Set closest triangular face to counterclockwise orientation and set orientation of neighboring faces to the opposite orientation.	39
Figure 5.12	Creating b-rep surfaces: (1) One of the outer surfaces, (2) Starting triangular face with counterclockwise orientation, (3) Determine the position of the vertex that is not already included in the b-rep surface, (4) Vertex is added between vertices of the b-rep surface.	40
Figure 5.13	OBJ visualization of Simple.ifc	41
Figure 5.14	Merging the roof and wall surfaces: (1) select downwards oriented roof surfaces, (2) create a bounding box of this surface, (3) select the wall vertices that lie within the bounding box, (4) add the points in counterclockwise order to surface, (5) apply to all downwards oriented faces, (6) result.	42
Figure 5.15	3D city model visualized in ninja	44
Figure 7.1	Pin located on Survey Point in Revit on the location Schependomlaan	54

Figure 7.2	Coordinates pin checked in viewer.	54
Figure 7.3	The simple BIM with a flat roof.	57
Figure 7.4	Other building elements added to the simple BIM.	57
Figure 7.5	Selecting the outer surfaces when the footprint is not rectangular	58
Figure A.1	From maximum building height to architectural elements to entities in IFC and CityJSON	71
Figure A.2	From maximum gutter height to architectural elements to entities in IFC and CityJSON	71
Figure A.3	From maximum height other constructions to architectural elements to entities in IFC and CityJSON	71

LIST OF TABLES

Table 3.1	Determining which information will be converted.	15
Table 5.1	Required information summarized per regulation	28
Table 5.2	Linking the required information to entities	30
Table 5.3	The amount of appearances of entities in Simple.ifc	31
Table 5.4	The presence of entities in the SOR	31
Table 5.5	Determining which information will be converted.	32
Table 5.6	Latitude longitude (DMS) of Survey Point	42
Table 5.7	Rotation to True North	42
Table 5.8	Ground height closest to the Survey Point from AHN4	43
Table 7.1	Deviation in X and Y dimension for each location of the Survey Point from Revit and the viewer.	54
Table A.1	Measurements for each rule: (a) Schependomlaan, (b) Haviklaan, (c) Rabarberstraat, (d) Witte de Withstraat, (e) Wijndaelersingel, (f) Jufferstraat	72
Table A.2	Errors 3D BGT detected by val3dity: (a) Schependomlaan, (b) Jufferstraat, (c) Den Haag	72

ACRONYMS

BIM Building Information Modeling	ii
IFC Industry Foundation Classes	ii
DTM Digital Terrain Model	7
AHN Actueel Hoogtebestand Nederland	8
GML Geographic Markup Language	15
CRS Coordinate Reference System	16
SOR Samenhangende Objectregistratie	1
AHN Actueel Hoogtebestand Nederland	8
T3D Totaal Driedimensionaal	5
VNG Vereniging van Nederlandse Gemeenten	7
AEC Architecture, Engineering and Construction	1

1 | INTRODUCTION

Building regulations in the Netherlands ensure the high quality of buildings and city development [CHEK, 2021] and are stated by the government, the provinces, and municipalities. For most construction work, these regulations have to be followed, or a permit has to be requested and granted [Wu, 2021]. The building permit process includes checking building designs against these regulations and granting permits. In the Netherlands, the process is mostly digitalized, however, there are still some issues. Whenever the application of a building design is submitted, the information is downsized [Dijkmans and van Berlo, 2013]. Architects create a 3D model of the building design, but have to deliver 2D drawings for the building permit process. Besides, checking the building design against the regulations is done manually by the municipality. The current duration between the request and the grant of the permit can be maximum 8 weeks, but can be extended with 6 weeks. It is clear that the process takes time and can cause errors [CHEK, 2021]. To overcome these issues, further digitalization and improvement of the building permit process have been a widely studied topic in EU countries. Many countries calculated their savings in terms of time and money with a digitalized process [Noardo et al., 2020b]. In order to enhance digitalization, information must be exchanged from 3D building design models to 3D city models.

On the one hand, 3D city models stem from the geospatial domain. An example of a 3D city model is the Samenhangende Objectregistratie (SOR), which will consist of mostly 2D and partially 3D information. It will merge several key registers, each with its own origin and purpose. The SOR is not yet in use, but is in the development phase. The key registers that will be included in the SOR are the BGT (Basisregistratie Grootchalige Topografie), BAG (Basisregistratie Adressen en Gebouwen), BRT (Basisregistratie Topografie) and WOZ (Basisregistratie Onroerende Zaken) [Leijten et al., 2021]. Because each of these key registers has its own content, semantics, information modeling, finance, and governance, there is little cohesion. The above-mentioned key registers will be merged into an efficient and complete key register, which will improve the exchange and maintenance of information [Odiijk, 2019]. The SOR, like other 3D city models, covers large areas. As a result of the increasing availability of computing power, more advanced data acquisition techniques, and automated workflows [Ohori et al., 2018b], these models have become so increasingly accurate, single buildings rich in details can be modeled.

On the other hand, designed building models or BIMs stem from the Architecture, Engineering and Construction (AEC) domain [Noardo et al., 2020b]. BIMs are used for the design and construction of building sites. The increase of computing power and the availability of better software allows users to incorporate the surrounding features of their designed buildings into the BIM software. This results in enhanced standards of the AEC domain to support datasets from the geospatial domain [Ohori et al., 2018b].

Since the building permit process is in demand to be further digitalized and the geospatial domain and AEC domain are starting to overlap, an integration between the domains is desired. The BIM could enrich the 3D city model with more details

on the buildings, while the 3D city model could enrich the BIM with surrounding features. Yet, integration is not as straightforward as it might seem. BIMs are most often encoded in IFC, while 3D city models could be encoded by CityJSON. The domains also differ in the focus, range of detail, industry, and use [Diakité, 2018].

This research aims to overcome current issues in the building permit process by performing an automatic rule checking based on a 3D city model. As a result, the current building permit process will be further digitalized by means of integration. In this research, integration implies data interoperability between the AEC domain and the geospatial domain [Zhu et al., 2018]. The integration in this research consists of information from on the one hand, the BIM encoded in IFC and on the other hand, the SOR which is assumed to be encoded in CityJSON, that will be combined into a 3D city model encoded in CityJSON. The newly created 3D city model will be used for automatic rule checking, since it can incorporate information about single buildings and its context. CityJSON is used as it is a less complex and a more user-friendly standard than CityGML [Ledoux et al., 2019].

A substantial part of land use plans include rules that require information from buildings, BIMs, as well as information of the context, 3D city models. The SOR incorporates information about the context and is assumed to be encoded in CityJSON, which means that this information has to be selected and taken into the 3D city model also encoded in CityJSON. On the other hand, the information of the BIM is encoded in IFC, which is a standard from the AEC domain. As pointed out before, retrieving information from IFC and using it in 3D models encoded in CityJSON is not a simple task, since these standards belong to different domains. To have information about the BIM in the 3D city model, information must be converted from IFC to CityJSON. When all required information is combined in one 3D city model, automatic rule checking based on the rules of the land use plan will be much simpler to perform. This is done to overcome the issues in the current building permit process. From this research, implications of this tool are described and guidelines can be drafted that enable BIMs to be modeled properly to support the digital building permit process as well as guidelines that support further integration between the two domains.

To achieve automatic rule checking, several steps need to be taken into account. Before performing the checks, several land use plans are inspected on the rules. The most often used rules are further inspected on required information. The required information is linked to entities in the BIM and the SOR. Since the surrounding context is part of the rules in the land use plan, a 3D city model has to be created in which the rule checking can be performed. The information of the BIM needs to be converted to CityJSON, a standard used for 3D city models. This is done since some rules from the land use plan acquire information from the surrounding context.

1.1 RESEARCH OBJECTIVES

This research aims to overcome the current issues in the building permit process by performing an automatic rule checking based on the rules of land use plans. A 3D city model encoded in CityJSON is used for the automatic rule checking. It is created by extracting information from the SOR encoded in CityJSON and converting information from BIMs encoded in IFC. This leads to the main question of this research:

How could BIM support the digital building permit process in the Netherlands?

This research will be split into the following sub-questions:

1. Which information from the **BIM** and the **SOR** is needed for the digital building permit process?
2. Which required information in the **BIM** and the **SOR** is present? Which required information is absent?
3. Which information from the **BIM** and the **SOR** should be converted regarding the digital building permit process?
4. How to perform a conversion from a **BIM** and the **SOR** supporting the digital building permit process?

1.2 SCOPE

In this research, only the rules dictated by the municipalities in land use plans are analyzed. Regulations, for example, the ones stated in the Building Decree (in Dutch: Bouwbesluit), will not be discussed. This means the checked rules are focused on spatial planning rather than building quality.

The conversion will be performed from a **BIM** and the **SOR** to a 3D city model. The **BIM** is created for the purpose of this research and is encoded in **IFC**, the most common standard in **BIM**. On the other hand, the **SOR** is assumed to be encoded in CityJSON, which is a JSON-based encoding for a subset of the CityGML data model [Ledoux and Dukai, 2022]. Other standards will not be used in this research.

The subterranean network will also not be handled during this research. Finally, only a conversion from **IFC** to CityJSON is performed, not the other way around.

1.3 THESIS OUTLINE

The paper consists of the following chapters:

- Chapter 2 states the legal framework and further background information related to this research. Besides, other research focused on the digital building permit process and integration is given.
- Chapter 3 illustrates the methodology of how to reach the end results. Each step in the research is explained with details.
- Chapter 4 states the implementation of this research as well as the datasets and tools.
- Chapter 5 shows the main results of this research.
- Chapter 6 formulates the implications of automatic rule checking.
- Chapter 7 states the discussion based on the results of the research.
- Chapter 8 draws up guidelines derived from the research.
- Chapter 9 formulates the conclusion of the research and answers all the research questions.

2 | BACKGROUND AND RELATED LITERATURE

In this chapter background information is provided as well as related literature. The background information is needed to understand the research and is information relatively unknown in the geospatial domain. The related literature are papers describing researches with similar topics as this research.

2.1 BACKGROUND

In this section, relatively unknown information for the geospatial domain is stated. This section is split into several topics: the legal framework, **BIMs**, Totaal Driedimensionaal (**T3D**), and current digital building permit checks.

2.1.1 Legal framework

The building permit process in the Netherlands is regulated in several laws: Wet ruimtelijke ordening (Wro), Wet algemene bepalingen omgevingsrecht (Wabo), Besluit ruimtelijke ordening (Bro) and Algemene wet bestuursrecht (Awb). Finally, the future legal framework is introduced.

WRO The Wro arranges the way spatial planning is executed and which governmental instance is responsible. The most important instrument within the Wro is the land use plan. Municipalities are responsible for arranging land use plans for every inch of its municipal area. A single building lot could be part of multiple land use plans (in Dutch: bestemmingsplannen) [Rijkswaterstaat, 2022]. Land use plans consist of three parts: clarification (in Dutch: toelichting), imagery (in Dutch: verbeelding), and rules (in Dutch: regels).

The clarification is a spatial foundation of the land use plan. In this part multiple subjects are explained: the goal of the land use plan, relevant environmental aspects, the feasibility of the land use plan, and the explanation of the rules.

The second part of the land use plan is the imagery. It is a digital map consisting of functions and indications. Living, sports, and agriculture are some categories of functions, which are depicted in colors on the digital map. Indications have to do with the measurements of building lots and measurements within the building lots.

Finally, the rules are stated in the land use plan. These are divided in several chapters: preceding rules, function rules, general rules, and the final chapter consists of the transitioning and final rules. The preceding rules contain the definitions and ways of measuring. The function rules provide the specific fulfillment of the functions on the imagery of the land use plan. It is also stated whether buildings are allowed and if so, which height the building cannot exceed, what the maximum built area may be, etc. The general rules state how deviation is possible in the land use plan with percentages and dimensions. Usually, in this part, it is stated that a deviation of 10% on the land use plan is allowed (on condition).

BRO Municipalities are obliged to produce a new land use plan every 10 years that covers every inch of the municipal area. The procedure of establishing a new land use plan is regulated by the Wro and the Bro [SAB, 2022].

WABO A building design is tested against the rules stated in the land use plan specific to the building lot in which the building design will be realized. If the requirements of the rules are not met, it is still possible to execute the building design under conditions stated by the municipality. This means the building design will deviate from the land use plan. This is possible by requesting a permit at the municipality in which the building design will be realized. There are two categories of deviations. The first is a deviation within the land use plan. The municipality already has stated conditions to deviate from the rules in the land use plan. With a permit, the building design can be deviated within these conditions.

The other category is the deviation outside the land use plan. In this case, the permit must be supported by another document (in Dutch: ruimtelijke onderbouwing) which states whether the building fits its environment, whether there is a necessity of this type of building, whether the building is protected to the noise levels in the environment, and whether the air quality is not exceeding standard [Rijkswaterstaat, 2022]. Other aspects from the environment may be included, but the before-mentioned aspects must be.

Besides deviating from the land use plan, a permit can be requested for other purposes as well. Some examples are demolition of buildings, cutting trees, and fire safety. More purposes of the permit are described in the Wabo [SAB, 2022].

AWB The standard procedure of requesting and granting a permit is stated in the Awb and consists of the following steps: file an application, the publication of the application, and the decision on the application [SAB, 2022]. If one of the parties disagrees with the outcome, an objection can be lodged at the municipality within 6 weeks. Whenever one of the parties disagrees with the outcome of the objection, an appeal can be lodged within 6 weeks. If one of the parties disagrees with the outcome, there is another final opportunity to lodge an appeal. This will be done at the district court and must be done within 6 weeks after the result of the previous appeal. Finally, the case can be appealed at the Administrative Jurisdiction Division of the Council State (in Dutch: Raad van State). A brief overview of the procedure of the permit is given in figure 2.1.

Automatic rule checking can be useful at different stages of the building permit process. It can be done before requesting a permit by the architects, just after the request by the municipality, and when the municipality has to make a decision.

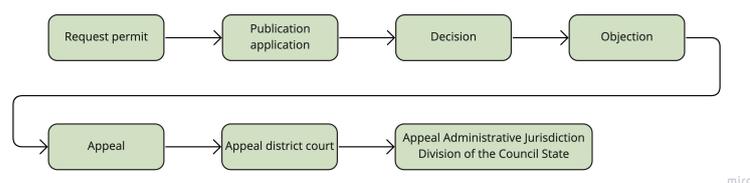


Figure 2.1: The procedure of the building permit [SAB, 2022].

FUTURE 10 years ago the government aimed to simplify and decentralize the laws with regard to among other things spatial planning. The government aims to assemble over 100 regulations into one law, the Environment and Planning Act (in Dutch: Omgevingswet). This new law would come in force on the 1st of January 2022, however, it has been delayed to the 1st of January 2023. One of the causes of this delay has to do with the ICT system which is needed to support the law. This system, Digitaal Stelsel Omgevingswet (DSO), will allow to show all the map layers of different governmental instances as well as it is aiming to automatize the building permit process. The purpose of the ICT system is to combine several maps from different instances. However, there are still errors in crucial elements in the system.

2.1.2 BIM

Currently, the AEC domain is shifting from 2D planning to 3D object-based modeling [Jaud et al., 2020]. Previously, drawings and accompanying documents were used to describe buildings, however, digital models are gaining more popularity. BIM plays an important role in this trend, since it allows the exchange of digital models between different actors throughout the process. BIM complies with the IFC standard, an open data model, developed by buildingSmart [Floros et al., 2018]. Using IFC as a standard in the AEC domain enables seamless flow of design, costs, project, production and maintenance information [Laakso and Kiviniemi, 2012].

GEOREFERENCING Georeferencing is the process of specifying a geolocation, placement, of an asset on the surface of the Earth. Georeferencing is important for data conversion and data linking between different data formats [Jaud et al., 2020]. This is possible in IFC using the entity **IfcSite** that allows the storage of the real-world location using `RefLatitude`, `RefLongitude`, and `RefElevation`. In practice, it is still rather complex, which makes it hard to exchange information [Ohori et al., 2018b].

POSITIONING Each building element in IFC resides within a local coordinate system [Jaud et al., 2020]. This can be the global coordinate system of the model (absolute placement) or another local coordinate system of an element higher in the hierarchical structure (relative placement). There are three possibilities of relative placement which can be combined:

- Local placement: common practice in building design. For example, a wall is placed within a building story.
- Grid placement: elements are placed in a grid-like constellation. For example, Digital Terrain Model (DTM) is a raster-like representation.
- Linear placement: the coordinate system is an element relative to a curve. This is common in infrastructure design.

GEOMETRY There are multiple ways to represent 3D geometry in BIM, which can be combined [Jaud et al., 2020]:

- Boundary representation (b-rep): 3D vertices are linked to edges, which forms the basis of the faces of the solids.
- Tessellated geometry: an array of 3D vertices connected in loops represent polygonal planar surfaces.
- Sweep: a shape is extrapolated along a basis curve.
- Constructive solid geometry (CSG): the principles of point set theory are combined with Boolean operators.

SEMANTICS Building elements are referred to in a single line. In these elements, there are pointers to other properties and descriptive information about the element. Additional information is stored in other entities, for example, **IfcPropertySingleValue**.

2.1.3 Totaal Driedimensionaal

T₃D is a program to ensure better services to civilians and provide more complete information for societal tasks [Gemeente Den Haag, Gemeente Rotterdam, Gemeente Amsterdam, VNG, 2020]. The program is set up in collaboration with the municipality of Amsterdam, The Hague, and Rotterdam with the association Verenging

van Nederlandse Gemeenten (VNG). The central issue of this program is the currently missing 3D information in registrations. The 3D information from BIMs can therefore not be used for applications that require 3D information as well. There is demand for a complete 3D chain of information to provide more complete service to civilians and companies. Besides, 3D information could support societal tasks, for example, more efficient building permit checks, lower unnecessary costs in projects in public spaces, and better maintenance of municipal real estate [Gemeente Den Haag, Gemeente Rotterdam, Gemeente Amsterdam, VNG, 2020].

The municipalities focus each on an aspect of this program. The Hague will focus on integral data acquisition, Rotterdam on the registration of this data, and Amsterdam on the use and provision of the data. The eventual goals of this program is to have an integrated 3D object registration by 2024 [Gemeente Den Haag, Gemeente Rotterdam, Gemeente Amsterdam, VNG, 2020]. One of the first steps in this program is to create the so-called 3D Basisvoorziening.

3D Basisvoorziening

Basisvoorziening 3D is a dataset containing 3D objects. It is based on three datasets, namely the BAG, the BGT, and the Actueel Hoogtebestand Nederland (AHN) [PDOK, 2022]. The latter is a dataset with height values all over the Netherlands. Combining the height values of the AHN, buildings, bridges, roads, and other city objects are assigned with heights. Basisvoorziening 3D has three subtypes:

- 3D Basisbestand Volledig: contains 3D objects, like buildings, bridges, roads, etc.
- 3D Basisbestand Gebouwen: contains 3D buildings in which buildings can have multiple height values.
- 3D Hoogtestatistieken Gebouwen: contains 2D geometries of buildings with corresponding height values.

2.1.4 Digital building permit checks

There are already some researches being done concerning the digital building permit process. These are stated in this section.

Municipality of Amsterdam

The municipality of Amsterdam focuses on the digital building permit process as part of the T_{3D}. The use case used by the municipality, civilians can draw a building plan in 3D or deliver the building plan in 3D [Geo-Informatie Nederland, 2021]. This information can directly be used for the building permit process and 3D object registration, which is part of the T_{3D} project. The information will always be checked by the municipality.

Municipality of Rotterdam

The municipality of Rotterdam created a tool to check several rules from different categories: fire security, aesthetics, land use plans, and noise. This tool is still in the development phase, but is able to check the rules in two areas of the municipality (see section A.2.1).

2.2 RELATED LITERATURE

In this section, researches with similar topics as this research are described to acquire more knowledge about current developments. This is done for the digital building permit process and the integration.

2.2.1 Digital building permit process

Although there are many differences within the building permit processes in Europe [Pedro et al., 2011], most of them take a long time and are resource-intensive. There are multiple reasons causing this. On the one hand, the regulations that building designs must adhere to and the documents describing the building design are often open for interpretation. This means different decisions can be made for similar situations [Noardo et al., 2020a]. Further on, regulations can stem from different levels of the government hierarchy. Regulations stated on a national level overrule the regulations of land use plans in the Netherlands.

On the other hand, current building permit processes rely on PDF or even paper documents submitted for the application of permits. A web service is available in the Netherlands to submit the application [Noardo et al., 2020a]. These PDFs are manually checked by authorities. The complex and paper-based processes increases awareness of further automation and digitalization.

2.2.2 Full integration

Integration means the interoperability of data between the geospatial and AEC domain [Zhu et al., 2018]. Integration can be conducted on several levels: the fundamental level and the application level. The examples discussed in this section belong to the first category. The integration on the fundamental level can be divided into geometry and semantics [Zhu et al., 2018]. The vast majority of research is done on the integration from BIM encoded in IFC to a 3D city model encoded in CityGML, since IFC and CityGML are the most commonly used standards in their respective domain. There are several programs that allow the possibility to convert all the BIM data to a 3D city model. Three examples are: Building Information Modelserver, IfcExplorer, and Safe FME [Ohori et al., 2018b]. Other attempts have been tried as well. de Laat and van Berlo [2011] made an extension to a 3D city model encoded in CityGML in which BIM data encoded in IFC could be stored. In the GeoBIM project by Ohori et al. [2018b] IfcOpenShell was used in combination with CGAL and Boolean set operations to obtain a single Nef polyhedron. All of these attempts have concluded that a full integration of geometry and semantics is complex. This has to do with different challenges [Noardo et al., 2020a]:

- Differences of modeling the world.
- Geometry representations: b-rep in geospatial domain and solids in the AEC domain.
- Scale of representation: geospatial domain covers large areas, while BIM focuses on a single construction project.
- Intended use of data: geospatial information is used for spatial analysis and BIMs are usually used for construction.
- Coordinate systems: in the geospatial domain geodetic reference systems are used and in the AEC domain Cartesian coordinates are used.

2.2.3 Integration (use case: building permit process)

It becomes clear from the previous section that full integration is rather complex. Multiple studies selected a use case, like the digital building permit process. The current building permit process has several issues at the moment: the downsize of information at the submission of the application and the manual checking by the authorities [Dijkmans and van Berlo, 2013]. Therefore, it was proposed by Dijkmans and van Berlo [2013] to develop 3D land use plans and perform rule checking in BIMs based on these 3D land use plans. Technically, this is possible if all stakeholders make the effort and have agreements among each other. Noardo et al. [2020b] developed a tool to check rules of the municipality of Rotterdam in the BIM. The tool is still in the development phase, but is effective in rule checking. Another research has been done to increase the efficiency of the building permit process by using automated checking of regulations from Swedish land use plans [Olsson et al., 2018]. The regulations were classified into three categories, and a separate methodology for each category was developed. The BIM encoded in IFC is transformed to a 3D city model encoded in CityGML. A ray-casting method was performed in combination with FME. It was concluded that the majority of criteria for land use plans are quantitative, which means they could be automated to a large extent. However, a fully automated permit process is not possible since it requires definitions of regulations suitable for BIM.

This chapter reviewed background information that provided knowledge outside the geospatial domain. Mainly knowledge about the legal framework and BIMs are described, which will be useful to understand the research.

Afterwards, the related literature showed the research that already has been done in the topics used in this research. Integration of the both domains as well as using the building permit process as a use case is already investigated in the past. In the next chapter, the methodology of this research is explained.

3 | METHODOLOGY

In the previous chapter, background information and related literature were described. With this information, the methodology of this research can be proposed in this chapter. The methodology points out which steps and procedures should be undertaken in order to answer the research question and sub-questions. Therefore, a general workflow is proposed in figure 3.1.

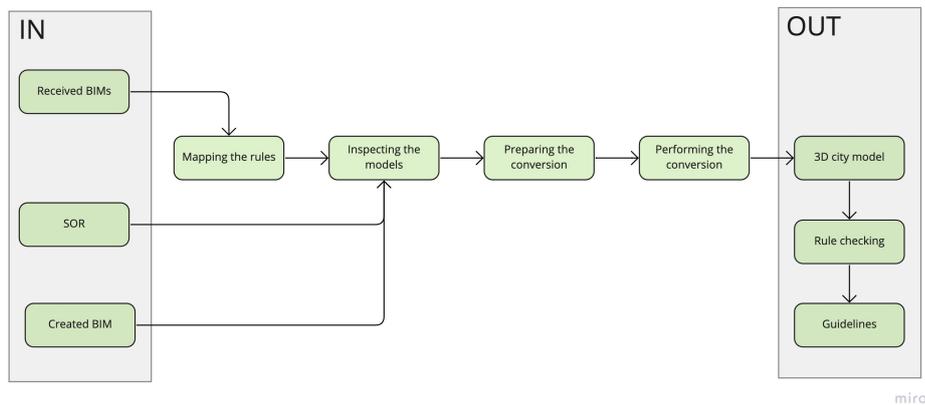


Figure 3.1: General workflow

The methodology contains four stages to answer each research sub-question. Each section will be explained in a separate section. In the first section, the rules stated in the land use plans will be analyzed on its information (see section 3.1). In this research, the rules are stated in the land use plan and do not include regulations like the Wro. Once the information is linked to the corresponding entities in IFC and CityJSON, the BIMs and the SOR will be inspected on the required information (see section 3.2). The entities in the standards and the input models are inspected on its presence. By means of the presence or absence of the required information, it is determined which entities are converted (see section 3.3). Finally, the conversion of the entities is performed from the BIM and the SOR to a new 3D city model encoded in CityJSON.

3.1 MAPPING THE RULES

In this section, the first sub-question is investigated: Which information from the BIM and the SOR is needed for the digital building permit process? To answer this question, several BIMs were received to work with and used to retrieve several land use plans that will be analyzed on its rules. The steps shown in figure 3.2 will be discussed separately.

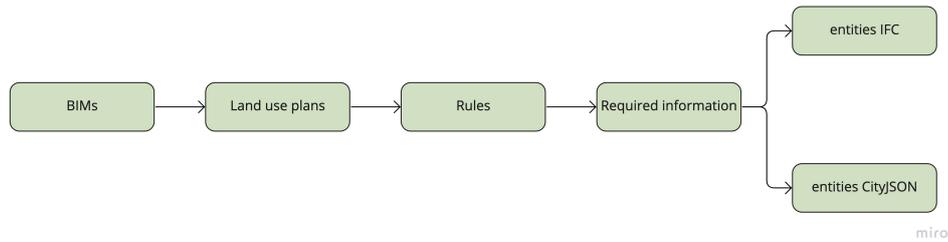


Figure 3.2: Workflow mapping the rules

3.1.1 Retrieving the land use plan

First of all, the rules stated in the land use plan will be retrieved. The postal address of the BIMs are inserted into an online viewer, [Ruimtelijkeplannen.nl](https://www.ruimtelijkeplannen.nl). All the relevant plans are visualized for the area, as well as additional information about the building lot. It is important to select the land use plan that has been irrevocably established (see figure 3.3). The spatially related rules relevant for the BIM in the land use plan are analyzed.



Figure 3.3: Selected land use plan in online viewer

3.1.2 Selecting the rules

For each BIM, a land use plan will be retrieved. The rules from the land use plans are compared on their presence and absence. The rules present in most land use plans will be further analyzed.

3.1.3 Determining the information

This subset of rules chosen in the previous section will be analyzed on its information needed to check the rule. This process is repeated for each rule in the subset.

3.1.4 Linking the entities

Finally, the information will be linked to entities in IFC and CityJSON. Linking the entities in this research means to find entities that represent the same information as the required information for the rules. The information can be a building element, for example a roof. The entities of IFC are analyzed on the information. Entities are objects containing information, that can have attributes to further define the information, and that are used to represent certain objects. If an entity represents

the same object as the required information for the rules, they will be linked. The same approach is repeated for the entities of **CityJSON**.

3.2 INSPECTING THE INPUT

In the previous section, the required information from the land use plans is linked to entities in both standards. In this section, the second sub-question is to be researched: Which required information in the **BIM** and the **SOR** is present? Which required information is absent? The input models will be checked for the entities determined in the previous section on its presence. On the one hand, a simple **BIM** is created in Revit for the purpose of this research (see figure 3.4). This **BIM** will be checked on the presence of entities determined for **IFC** in the previous section.

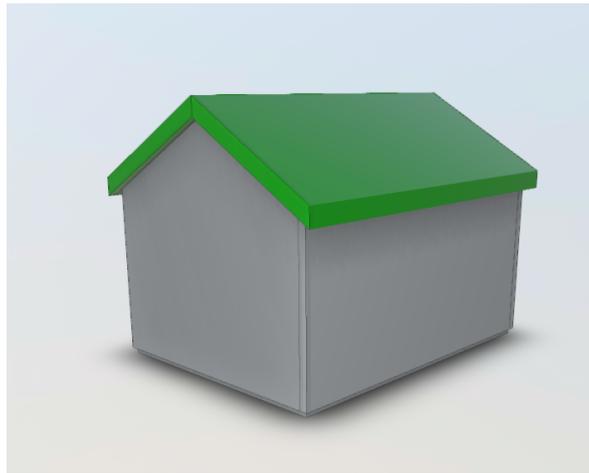


Figure 3.4: Simple.ifc

On the other hand, the key registers that will form the **SOR** will be checked for the presence of the required entities from **CityJSON**. Finally, the presence and absence is summarized for both input models (see figure 3.5).

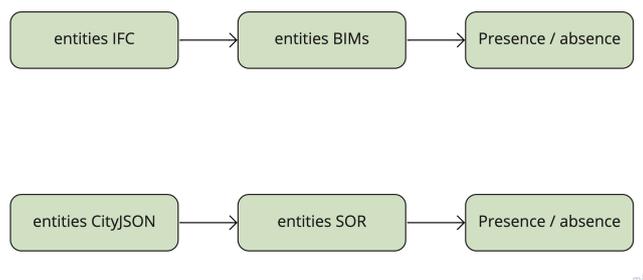


Figure 3.5: Workflow inspecting the models

3.3 PREPARING THE CONVERSION

The aim of this research is to perform automatic rule checking. The information from the **BIM** and the **SOR** will be converted to a 3D city model, since it is less complicated to test a 3D city model on the rules. In the previous sections, the presence and absence of entities in the input models were checked. The third sub-question of

this research is: Which information from the **BIM** and the **SOR** should be converted regarding the digital building permit process? In order to answer this research question, it must be made sure which information will be converted from which input model and whether all the required information can be converted.

First of all, it is determined which information will be converted from which input model. If the required information from the rules is present in both input models, it depends on the sort of information. All information belonging to buildings, will be converted from the **BIM** and not the **SOR**. The building design in the **BIM** will be the future situation that must be checked, while the building in the **SOR** is the current situation. Information about the surroundings will be converted from the **SOR** on the contrary. The surrounding features will likely remain unchanged in the building design. If changes appear in any of the registers, it is obligated to report these changes to [Kadaster, 2022a]. These changes will then be corrected in the future version of the key register. If the information is present in neither input models, this part of the conversion will not be possible.

		BIMs	
		present	absent
SOR	present	conversion SOR/BIM	conversion SOR
	absent	conversion BIM	conversion not possible

Table 3.1: Determining which information will be converted.

Secondly, it will be determined whether all the required information can be converted. If the required information is incomplete, a solution is to be found. The required information could be supported by other datasets or replaced completely by another dataset. This step is to ensure all the required information can be converted.

3.4 PERFORMING THE CONVERSION

After the required information has been categorized and possibly replenished, the conversion can be performed to investigate the final sub-question: How to perform a conversion from a **BIM** and the **SOR** supporting the digital building permit process? The conversion has two starts as can be seen in the proposed workflow (see figure 3.6). The **BIM** created in Revit and exported to an **IFC** file is a start as well as the key registers of the **SOR** encoded in Geographic Markup Language (**GML**). The required information from both input models will be combined and converted to a 3D city model encoded in CityJSON. The 3D city model can incorporate the required information of building designs as well as its surroundings. Since all the required information is combined in one 3D city model, the rule checking will be much simpler to perform.

3.4.1 Data preparation

A new simple **BIM** is created for this research (see figure 3.4). The **BIM** contains four walls, a ground floor, and a gabled roof.

In order to improve the exchange between geospatial and **AEC** applications, the **BIM** needs to cover some desirable aspects [Ohori et al., 2018a]. An important guideline is georeferencing. This means the **IFC** files should contain precise real-world location coordinates [Ohori et al., 2018a]. This guideline is crucial for rule checking. The **BIM** shown in figure 3.4 will be georeferenced to the locations of the **BIMs** used for the land use plans.

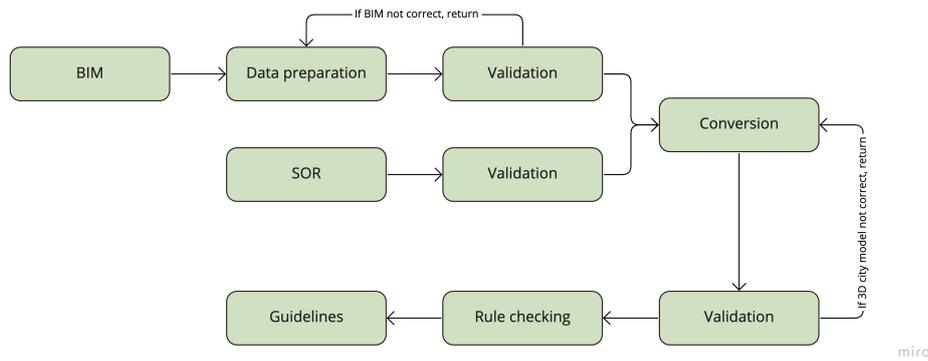


Figure 3.6: Proposed workflow conversion

Other guidelines have been proposed by [BIM Locket \[2022\]](#), which are divided in two categories. On the one hand, the guidelines improve the exchange of [BIMs](#) and on the other hand, guidelines that state the minimum amount of information in a [BIM](#). It is important that [IfcPropertySets](#) are used, since *IsExternal* is a relevant property. Elements of a building bordering the outside are labeled "TRUE" for this property, while internal elements are labeled "FALSE".

3.4.2 Validation

Before the conversion is performed, both input models are validated. This is done to ensure that the input models are correct and do not contain any errors, that could potentially cause issues later on in the process.

BIM The previously mentioned guidelines will be checked upon before performing the conversion. On top of that, the [BIM](#) must be validated using [IfcCheckingTool](#). The tool shows a report of errors of the [BIM](#). If there are any errors, these must be analyzed. Errors that do not exceed the guidelines [[Ohori et al., 2018a](#)] [[BIM Locket, 2022](#)], can be accepted and the conversion can be performed. If these errors exceed the guidelines, the simple [BIM](#) must be improved on these errors. If no errors occur, the conversion can be performed.

SOR The [SOR](#) is not yet in use, therefore, the existing key registers that will eventually be merged into the [SOR](#), are utilized in this research. Just as the [BIM](#), the key registers are validated, before the conversion is being performed. If any errors occur, these must be overcome before continuing to the conversion. If no errors occur, the conversion can be performed.

3.4.3 Conversion

Once the validation of the [BIM](#) and the key registers of the [SOR](#) is done successfully, the conversion can be performed. The required information determined from the rules of the land use plan will be converted from the input models to a new 3D city model encoded in CityJSON. The conversion consists of two parts. Foremost, the conversion of the [BIM](#) encoded in [IFC](#) to a new 3D city model. Secondly, the key registers are currently encoded in [GML](#) and need to be converted to the same new 3D city model encoded in CityJSON. The information that will be converted is divided in three categories: geometry, semantics, and the Coordinate Reference System ([CRS](#)).

The categories of information will be discussed for the input models, the [BIM](#) and the [SOR](#), at first. It will be discussed which information from the categories must be handled to perform the conversion. Afterwards, the categories of information will also be handled for the new 3D city model. It is stated where the above-mentioned information will be placed in the CityJSON file.

BIM

GEOMETRY Only the required information from the [BIM](#) will be converted. The representation of geometries can be: b-rep, tessellated geometry, sweep, and CSG. The representation differs per element in the [BIM](#). The different sorts of geometry must be converted into b-rep surfaces for the CityJSON standard. The geometry is represented by separate entities linked to the building element.

SEMANTICS The semantic information that must be converted is stored in several ways in [IFC](#). The type of the entity will be used to distinguish the different building elements from one another.

Further information of this entity is given in the **IfcPropertySet**. This entity consists of multiple **IfcPropertySingleValues** that could state the *FireRating* or whether an entity is external. It depends on the producer how much information is stated in the **IfcPropertySet**. In this research, the **IfcPropertySingleValue** *IsExternal* will be used to select only the outer elements of the [BIM](#).

CRS Using georeferencing it is possible to assign WGS84 coordinates to a [BIM](#). However, the latitude and longitude are assigned to a single point in the [BIM](#), the Survey Point. [IFC](#) uses local placement of geometries for the rest of the points. This means objects are using a Cartesian coordinate system. All geometries inside **IfcSite** are placed relative to **IfcSite**.

By georeferencing, WGS84 coordinates are assigned to the entity **IfcSite** which has the attributes *RefLatitude*, *RefLongitude*, and *RefElevation*. Besides, the orientation is assigned with **IfcDirection**. It is important to note that the WGS84 coordinates must be transformed. In the Netherlands, RD-coordinates are used for all maps and applications, for example the rule checking of the land use plans. The transformation is only applied to the Survey Point, while the rest of the points are modeled regarding the Survey Point.

SOR

The current key registers are encoded in [GML](#). This standard differs from CityJSON as well, however, not as much as [IFC](#), since both standards are used in the geospatial domain.

GEOMETRY Only the required geometries will be converted from [GML](#) to CityJSON. Contrary to [BIMs](#), the key registers are mostly 2D information.

SEMANTICS Just as in CityJSON, it is possible to set an unlimited amount of attributes in [GML](#). As well as the geometry, the semantic information is stored inside the city object.

CRS Unlike the [BIM](#), the key registers are using RD-coordinates already. In the Netherlands, the RD-coordinates are used for all purposes and thus converted from WGS84 or ETRS89. The X and Y coordinates do not need to be transformed, but the Z coordinate is absent.

3D city model

The 3D city model will be written in CityJSON (v.1.1.). The future geometry, semantics, and coordinates are described per section.

GEOMETRY The 3D city model is newly created and will be encoded in CityJSON. The geometry of a city object is included in the city object. The b-rep surfaces are listed under the attribute *boundaries*. Instead of having the coordinates inside the boundaries, it includes an index referring to the position of a vertex in the member “vertices” [Ledoux and Dukai, 2022].

SEMANTICS The semantic information in CityJSON can be divided into three categories. City objects in CityJSON must be assigned with a unique global ID. In this research, the global id from the produced BIM is used as the id.

On top of that, the building can have an unlimited amount of attributes [Ledoux and Dukai, 2022] that entail further information. For example, the type of roof can be included in the attributes section.

The final category is the so-called Semantic Object in CityJSON which represents the semantics of a geometric primitive [Ledoux and Dukai, 2022]. This means that the different surfaces must be appointed with a type, for example, a **RoofSurface**. These semantics may have attributes, like a parent-child relationship or other attributes.

CRS The coordinates will be stored on several places. The CRS can be defined at the start of a CityJSON by stating the EPSG code. EPSG is a registry that entails the parameters of many CRS. This code enables to switch between different CRSs in GIS software [Idrizi, 2021]. By stating the EPSG code EPSG:28992, the CityJSON is set to the correct CRS in the Netherlands, containing the updated RD-coordinates.

Each city object will have a member “boundaries” within the geometry. The boundaries will store the b-rep surfaces in the form of indices. These indices refer to the place of the vertex in the list vertices.

The **transform** member holds a *translate* and *scale* attribute. These can be set manually to avoid storing all the large coordinates and instead transform the coordinates with the *scale* and *translate*.

3.4.4 Validation

val3dity is used to validate the newly created 3D city model. 3D geometric primitives of the CityJSON file can be validated. Besides val3dity, cjval is used to validate the resulting 3D city model. It checks, unlike val3dity, on JSON syntax. If errors occur during the validation, these are analyzed and the process will return to the conversion. This process is repeated recursively until no errors appear at the validation.

3.4.5 Rule checking

If no errors occur, the 3D city model can be used for rule checking. The rules determined in section 3.1.2 will be checked using the 3D city model. A separate script will be written that takes the newly created 3D city model as input and checks the model on the most frequently appeared rules in the land use plans. There are three different categories in which the outcome can be placed. The first category means that the model passed the rule checking.

The second category holds a threshold. A significant amount of land use plans state that it is possible to deviate with a maximum of 10% of the measurements. Whenever the result exceeds the rules, but within the 10% the model is placed in the second category. Deviation could be possible, however, it is possible that several conditions are set for the deviation. Further investigation of the land use plans is needed.

When the model exceeds more than 10% over the rules, it is placed in the third category. The building design is rejected based on the rules in the land use plan.

3.5 IMPLICATIONS

Once the automatic rule checking has been performed, the implications of the research are described. This research has significant effects for the digital building permit process and the integration of the geospatial and [AEC](#) domain.

3.6 GUIDELINES

When the rule checking has been performed, the [BIM](#) in the digital building permit process can be assessed. From this assessment, several guidelines can be opted for an improved future integration. The guidelines will be drawn on two different levels. First of all, the guidelines will be stated on a generic level on the integration between the [AEC](#) domain and the geospatial domain. Afterwards, guidelines are drawn for [BIMs](#) which will be checked on the rules by the municipalities for the digital building permit process.

In this chapter, the methodology was described. However, further information about the input models, software, and other tools is still unclear. The next chapter states the information about the implementation.

4 | IMPLEMENTATION

In the previous chapter, the methodology of this research was explained. It is shortly mentioned which software and tools are used, however, in this chapter further information on these matters will be stated. The chapter is divided into two sections: tools and datasets.

4.1 TOOLS

This section describes the tools used in this research. All of them are freely available, except for Revit. These tools are necessary to complete tasks described in the methodology (see chapter 3).

4.1.1 AutodeskViewer

The input is visualized in an online desktop viewer developed by Autodesk. The BIMs encoded in IFC can be viewed freely. It includes several tools, for example, the model can be represented in an exploded view.

4.1.2 Revit

Revit is used for the design and documentation of buildings [Autodesk, 2022]. Users can produce drawings and schedules. For this research, it is used to create a simple BIM and to georeference it. Afterwards, the BIM can be exported to an IFC file.

4.1.3 IfcCheckingTool

Before the integration is performed, the simple BIMs used as input will be validated. The IfcCheckingTool is developed by the Karlsruhe Institute of Technology and checks the BIMs on semantic and syntactic errors [Karlsruhe Institute of Technology, 2022]. The tool is only available on Windows.

4.1.4 MeshLab

The intermediate results are written to an OBJ file and visualized in MeshLab. It is oriented to the management and processing of unstructured large meshes.

4.1.5 val3dity

After the integration has been performed, the newly created 3D city model will be validated by val3dity. This is to ensure the CityJSON file meets all specifications. Among other things, it checks on topology, geometry and the orientation of normal vectors. This is done according to the specifications in the international standard ISO19107 [Ledoux, 2018].

4.1.6 cjval

The 3D city model is also validated by `cjval`. Unlike `val3dity`, `cjval` check CityJSON files on its syntax. Several aspects are tested, namely the JSON syntax, CityJSON schemas, parent-children consistency, duplicate vertices, etc. [Ledoux and Feenstra, 2022].

4.1.7 QGIS

QGIS is a free and open source GIS software. The software can be used to view vector (like Shapefile) or raster data. On top of that, data can be edited, managed and exported to other standards. QGIS uses core features and plugins to edit data [QGIS3.22, 2022]. For this research, it is used to validate a shapefile. A core feature Geometry Checker is used.

4.1.8 ninja

The final 3D city model will be visualized in `ninja`. CityJSON files can be dropped in the web viewer and will be visualized.

4.2 DATASETS

In this section, the datasets used as input models for this research are described. Information from these input models are selected, modified, or retrieved for different purposes as described in the methodology (see chapter 3).

4.2.1 BIMs

For this research, a simple BIM is created in Revit.

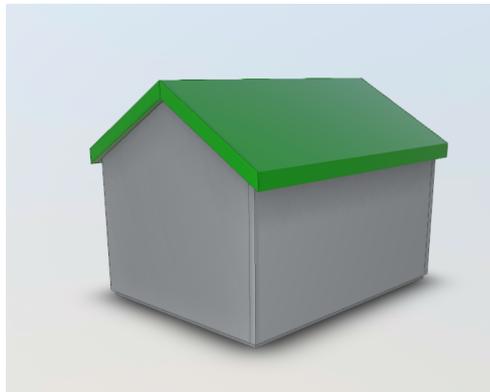


Figure 4.1: Simple.ifc

For the analysis of the land use plans, several BIMs are used which have been provided by the municipality of The Hague (Haviklaan.ifc, Rabarberstraat.ifc, Wit-tedeWithstraat.ifc, Wijndaelersingel.ifc) and Rotterdam (Jufferstraat.ifc). One BIM is publicly available (Schependomlaan.ifc). The BIMs can be viewed in figure 4.2 and 4.3. These BIMs will be solely used to retrieve the land use plans. The conversion will not be performed on these BIMs.

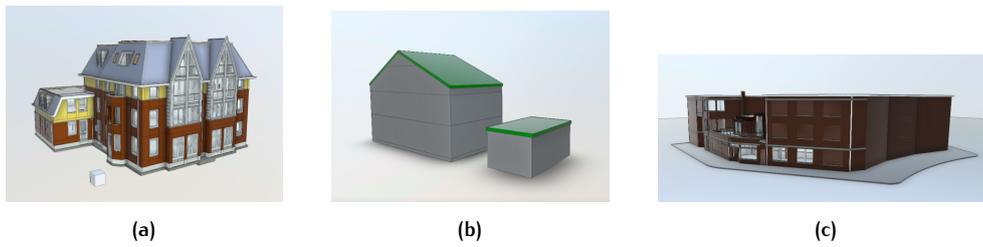


Figure 4.2: BIMs (a) Schependomlaan.ifc (b) Haviklaan.ifc (c) Rabarberstraat.ifc

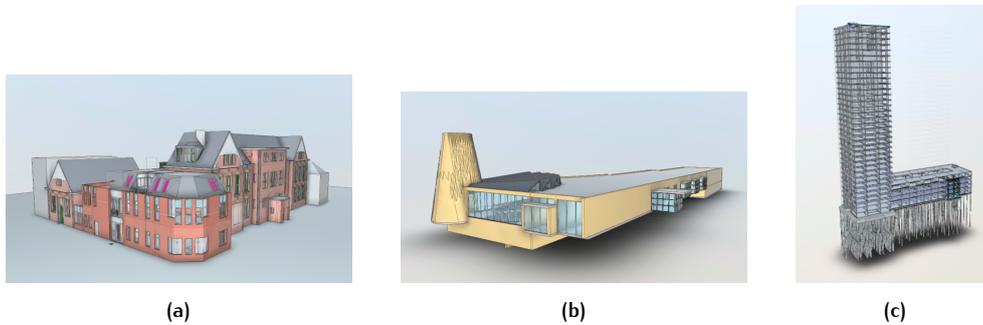


Figure 4.3: BIMs (a) WittedeWithstraat.ifc (b) Wijndaelersingel.ifc (c) Jufferstraat.ifc

4.2.2 SOR

The **SOR** will consist of several key registers: BAG, BGT, WOZ, and BRT. The government has appointed these registers as key registers. The system of key registers ensure the correct information is exchanged. Currently, the Kadaster is in charge of the maintenance of the registers together with the source data holders and the users in commission of the Dutch government. Mistakes in any of the registers must be reported and ought to be updated by the Kadaster.

BAG

The BAG entails information about addresses linked to a public space, number indication and place of residency [Kadaster, 2022b]. Besides addresses, buildings and other object types are defined and provided with a unique ID. Several attributes are added to the object types, for example, the building area and the year of construction.

Currently, a dataset covering the buildings and addresses on a national level can be downloaded. The downloaded file consists of several subfiles that include among other things, buildings, objects that are investigated, and objects that should not be included in the BAG.

BGT

The information in the BGT is about large-scale topographical objects, for example, buildings, roads, railways, etc. Each of these object types are provided with a unique ID.

The dataset can be accessed through a **viewer**. The viewer shows a map of the Netherlands. The postal address can be inserted, and the viewer will focus on the area surrounding the postal address. A tile can be created and downloaded. In the BGT, there are several formats available, CityGML, GMLLight, and StUF-GEO. The downloaded tile consists of several files that include the objects mentioned previously.

BRT

The BRT shows some similarities to the BGT. It contains several topographical elements, for example, roads, buildings, railways, etc. The difference compared to the BGT is that the BRT can be retrieved on different scale levels [Kadaster, 2022c].

The dataset can be downloaded similarly to the BGT. However, the tile will always be downloaded in [GML](#) format.

WOZ

In the WOZ, all buildings are provided with its WOZ value. The WOZ value determines the height of taxes and municipal fines that must be paid by the owner of the building [Kadaster, 2022d].

Unlike the other registers, the WOZ cannot be downloaded freely. There is an online [viewer](#) available that displays the value of the building. This can be done by inserting the postal address into the viewer. The viewer shows information about the building, among other things, the year of construction, WOZ value, and the built area.

4.2.3 3D Basisbestand Volledig

3D Basisbestand Volledig is part of the Basisvoorziening 3D. It can be downloaded online and is freely available. A map of the Netherlands appears and a tile within the country can be downloaded as a CityJSON or GeoPackage. A folder is downloaded with four files, each a quarter of the tile picked earlier on.

4.2.4 Peil

The municipality of Rotterdam created a dataset containing the peil values in the whole municipality. The dataset is written as a Shapefile. This dataset is only used for one of the received [BIMs](#) that resides in Rotterdam.

4.2.5 AHN4

[AHN](#) is a dataset for the whole Netherlands with detailed and accurate height information. The acquisition of information takes years, since it is measured using LiDAR. The [AHN](#) is a multi-annual project and cooperation between the Water Authorities (in Dutch: waterschappen), the provinces, and the Ministry of Infrastructure and Water Management (in Dutch: Rijkswaterstaat).

In this chapter, the tools and datasets needed to execute the methodology are described. In the next chapter, the methodology is executed, which produces the results.

5 | EXPERIMENTS AND RESULTS

In this chapter, the methodology will be executed with the help of the tools and datasets and the results will be presented. For this research, a simply created BIM will be used (see figure 3.4). All the steps discussed in the methodology will be applied to this BIM. This chapter is divided into five sections. The first four sections are dedicated to each sub-question. The final section discusses the effects of this research in the building permit process and the integration.

5.1 MAPPING THE REGULATIONS

In the current building permit process, the municipality checks whether BIMs satisfy the rules of the land use plans. The required information from the BIM needs to be collected using the land use plan that applies to the BIM. Therefore, the first sub-question of this research is as follows: Which information from the BIM and the SOR is needed for the digital building permit process?

The first step answering this sub-question is to select the most often used rules in land use plans. Six BIMs were received and used for selecting land use plans. These land use plans can be retrieved through [Ruimtelijkeplannen.nl](https://www.ruimtelijkeplannen.nl) and can be viewed in figures 5.1 and 5.2. The received BIMs reside on the red markers visible in the figures.

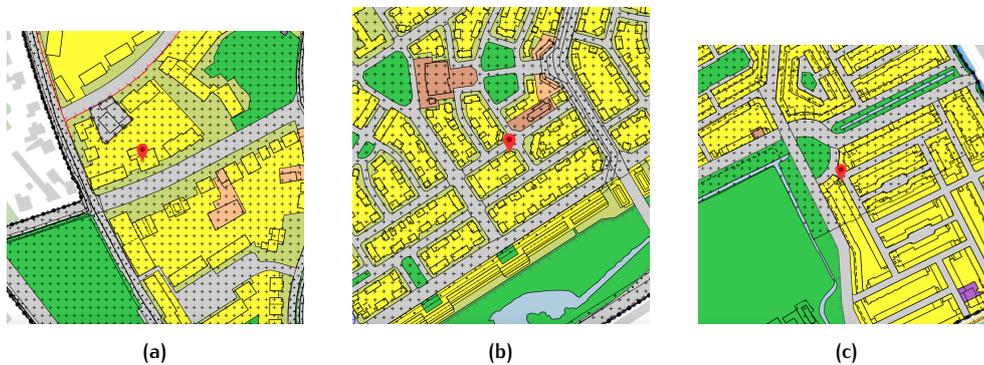


Figure 5.1: Land use plans of (a) Schependomlaan, Nijmegen (b) Haviklaan, The Hague (c) Rabarberstraat, The Hague. Yellow = living.

In the images, the areas are colored in different colors that represent a certain use stated in the development plan. Yellow represents living, brown represents a social function, and pink represents city centre uses. The land use plan entails rules for all uses that appear in the area of the land use plan. In the case of these BIMs, the rules of the land use plan corresponding to the use of the digital map will be analyzed.

All rules from these six development plans are compared on their frequency of appearing. The rules that most often appeared in the land use plans are:

- The maximum building height.
- The maximum gutter height.

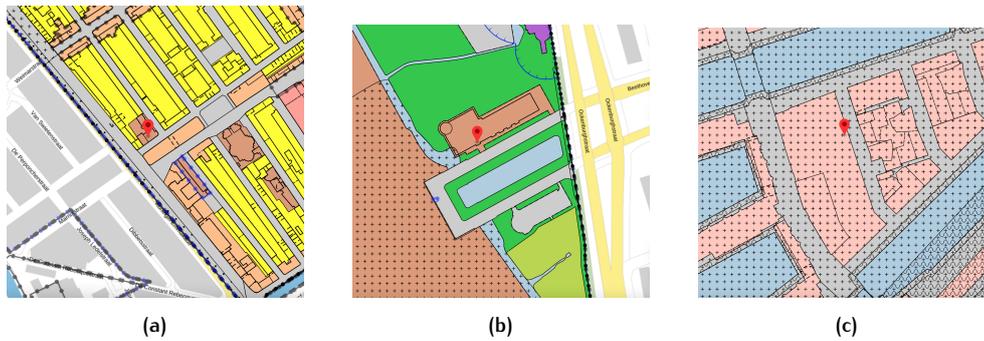


Figure 5.2: Land use plans of (a) Witte de Withstraat, The Hague (b) Wijndaelersingel, The Hague (c) Jufferstraat, Rotterdam. Brown = social function, pink = city centre.

- The maximum height of other constructions, not a building being, on the parcel.

Firstly, each rule will be analyzed on definitions to understand the whole meaning of the rule. Afterwards, the information required to check the rule is summarized based on the definitions.

5.1.1 The maximum building height

The first rule is checking the maximum building height of the building design. In the land use plan, different definitions of this rule are given. Therefore, the most commonly used definition of the rule will be used. The most frequently given definition is [Gemeente Den Haag, 2020] [Gemeente Rotterdam, 2015] [Gemeente Den Haag, 2014]:

“Vanaf het peil tot aan het hoogste punt van een gebouw of van een bouwwerk geen gebouw zijnde met uitzondering van ondergeschikte bouw(onder)delen, zoals schoorstenen, antennes, en naar de aard daarmee gelijk te stellen bouw(onder)delen.”

This definition states the maximum building height is measured from the peil to dominant building elements. Chimneys, antennas and other similar subordinate building elements are excluded when measuring the building height. To determine the point with the highest Z coordinate of a building, the main building elements are taken into account, namely the walls and the roof.

The peil is necessary to measure the height of buildings, but it is not explained in this rule. However, it is explained in the land use plan. The definition differs among the land use plan, therefore the most commonly used definition is given here [Gemeente Den Haag, 2016] [Gemeente Den Haag, 2020] [Gemeente Nijmegen, 2015] [Gemeente Den Haag, 2014]:

“Voor gebouwen, waarvan de hoofdtoegang onmiddellijk aan een weg grenst: de hoogte van de kruin van die weg; in de overige gevallen: de gemiddelde hoogte van het aansluitende afgewerkte maaiveld; in de gevallen waarin de hoogte wordt uitgedrukt in meters NAP: 0 m NAP.”

The entrance of the building determines how the peil can be calculated. The BIMs are further analyzed from figure 5.1 and 5.2. It can be concluded that all the BIMs are bordered immediately to a road. As a result, the roads must be taken into account for the rule checking.

Finally, the ground height of the building is also required to check the rule. The rule is checked from the peil value to the maximum height stated in the land use plan.

The height of the building is measured from the ground height to the highest point of the building. It is important to take the ground height into account. In figure 5.3, the height of the building itself is lower than the maximum building height stated in the land use plan. Since the building is located at an elevation, it surpasses the maximum building height that is measured from the peil value. This means that the ground height of the building must be taken into account in this research.

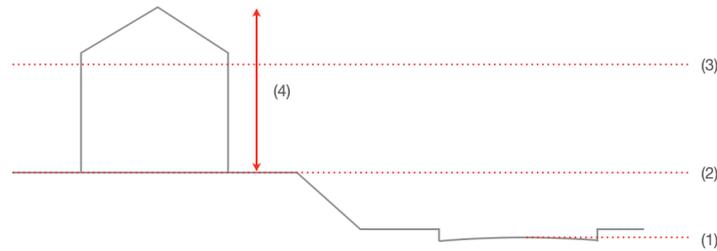


Figure 5.3: The ground height matters in checking the rules in this example. (1) the peil value, (2) ground height of the building, (3) maximum building height stated in the land use plan, (4) height of the building

5.1.2 The maximum gutter height

The most frequently used definition of the gutter height given in the land use plans is [Gemeente Den Haag, 2016] [Gemeente Rotterdam, 2015] [Gemeente Den Haag, 2020] [Gemeente Nijmegen, 2015] [Gemeente Den Haag, 2014]:

“Vanaf het peil tot aan de bovenkant van de goot c.q. de druiplijn, het boeibord of een daarmee gelijk te stellen constructiedeel.”

The gutter height is measured from the peil to the gutter. The gutter is usually attached to the lowest edge of the roof, which is assumed the corresponding element. In this rule, the peil is mentioned again, therefore, the roads are taken into account. On top of that, the ground height must be taken into account to calculate the highest point of the building.

5.1.3 The maximum height of other constructions

The maximum height of other constructions is calculated the same ways as the maximum building height. The highest point of the other construction, the roads for the peil value, and the ground height are needed to check the rule.

5.1.4 Required information

The required information from the previously discussed regulations is summarized in Table 5.1.

Rules	Required information
Maximum building height	Walls, roof, roads, ground height
Maximum gutter height	Roof, roads, ground height
Maximum height of other constructions	Other construction, roads, ground height

Table 5.1: Required information summarized per regulation

5.1.5 IFC

In this section, the previously determined information is linked to corresponding entities in IFC, if that is possible. If the required information from the land use plan can be represented by entities in the standards, they correspond to one another. In this research, this will be called linking. The entity types are listed by [buildingSmart \[2022\]](#).

Wall

Walls are represented by the entity **IfcWall** in IFC. This super-type has two sub-types that can occur in BIMs:

- **IfcWallStandardCase** is used for all occurrences of walls, that have a non-changing thickness along the wall path and where the thickness parameter can be fully described by a material layer set. These are always represented geometrically by an "Axis" and a "SweptSolid".
- **IfcWallElementedCase** is used for occurrences of walls which are aggregated from subordinate elements.

However, **IfcWall** represents all occurrences of walls within a BIM [[buildingSmart, 2022](#)].

Roof

The roof is represented by the entity **IfcRoof**. **IfcRoof** shall either be represented [[buildingSmart, 2022](#)]:

- as a roof assembly that aggregates all parts (slabs, rafters and purlins, or other included roofs, such as dormers) with own shape representation, or
- as a single roof without decomposition, including all shape representations directly at the roof entity.

The roof is a description of a total roof which could consist of slabs or beams. In the newly created BIM, the roof consists of slabs and this is the most standard roof that can be created in Revit. A slab is a component of the construction that may enclose a space vertically. The slab may provide the lower support (floor) or upper construction (roof slab) in any space in a building [[buildingSmart, 2022](#)]. Slabs are represented by **IfcSlab**. This super-type entity has two sub-types:

- **IfcSlabStandardCase** used for all occurrences of slabs that are prismatic and where the thickness parameter can be fully described by the **IfcMaterialLayerSetUsage**.
- **IfcSlabElementedCase** used for slabs aggregated from subordinate elements.

Roads, other construction, and ground height

The walls and roof can be represented using IFC entities. The remaining information, however, could not be linked to corresponding entities.

5.1.6 CityJSON

In this section, the required information will be linked to corresponding entities in CityJSON if possible.

Walls and ground height

Building is one of the 1st-level city objects in CityJSON. The building can be a semantic object which is able to represent semantics and store attributes. The semantic object must have a type, which can be among other things a *WallSurface*.

The ground height is included in this semantic object as the Z coordinate.

Roof

Similar to the *WallSurface*, the roof can be represented in a semantic object with the type *RoofSurface*.

Road

In CityJSON roads are represented by Roads under the category **Transportation**. Several classes can be omitted to the city object, like Section or Intersection [Ledoux and Dukai, 2022]. The geometry is saved in its boundaries which consists of vertices and its X, Y, and Z coordinate. From these vertices, the height of the crest of the road can be determined.

Other construction

One of the 1st-level city objects in CityJSON can represent other constructions, namely the OtherConstruction object. In the section attributes of the city object, a class can be assigned that could specify what sort of construction the city object is [Ledoux and Dukai, 2022].

5.1.7 Summary

Six land use plans were analyzed. Each of the documents contains lots of rules, however, most rules are unique to the neighborhood. Therefore, only three rules occur in almost every land use plan, namely:

- The maximum building height
- The maximum gutter height
- The maximum height of other constructions

For each of these rules, the required information is determined (see table 5.1). After the information has been determined, it can be linked to corresponding entities. This is done for IFC and CityJSON (see table 5.2).

Required information	IFC	CityJSON
Wall	✓	✓
Roof	✓	✓
Road	X	✓
Other construction	X	✓
Ground height	X	✓

Table 5.2: Linking the required information to entities

Since the roof in BIMs can be represented by slabs, **IfcSlab** is taken into account. When creating a BIM, the default roof that can be selected will be of type **IfcSlab** as part of **IfcRoof**. In customized BIMs, this could differ. An overview of the mapping of the rules to entities in IFC and CityJSON can be viewed in figures A.1, A.2, and A.3.

Now that the required information is linked to entities in both standards, it is known which entities must be used in order to check the rules. The next step is to investigate whether these entities are present in the input models.

5.2 INSPECTING THE MODELS

In this section, the second sub-question will be researched: Which required information in the **BIM** and the **SOR** is present? Which required information is absent? In the previous section, the required information was linked to entities in **IFC** and **CityJSON**. In this section, the entities in **IFC** and **CityJSON** will be inspected for presence in the simple **BIM** and the **SOR**.

5.2.1 BIM

Each entity discussed in section 5.1.5 is inspected in the created **BIM**. The presence of the entities are shown in table 5.3.

Required information	Simple.ifc
IfcWall	0
IfcWallStandardCase	4
IfcWallElementedCase	0
IfcRoof	1
IfcSlab	3
IfcSlabStandardCase	0
IfcSlabElementedCase	0

Table 5.3: The amount of appearances of entities in Simple.ifc

IfcWall, **IfcWallElementedCase**, **IfcSlabStandardCase**, and **IfcSlabElementedCase** do not appear in the **BIMs** used for this project. These are still taken into account in case these are used in other **BIMs**. This is done using the supertypes, **IfcWall**, **IfcRoof** for the modelling of the 3D city model. In the **BIM**, there are three **IfcSlab** entities present. One of them represents the ground floor, while the other two represent the roof.

5.2.2 SOR

The **SOR** will be inspected similarly for the presence of the required entities. Since the **SOR** will consist of several registers, these are checked upon the presence of the entities. The results are shown in table 5.4.

Required information	BAG	BGT	BRT	WOZ
Wall	X	X	X	X
Roof	X	X	X	X
Road	X	X	X	X
Other construction	X	X	X	X
Ground height	X	X	X	X

Table 5.4: The presence of entities in the SOR

Even though there are entities that could represent the wall, the ground height, and roof surfaces in **CityJSON**, these are not represented in the **SOR**. The **SOR** consists mostly of 2D information and in the cases of buildings only the footprint is part of the geometry. This means wall and roof surfaces are not modeled.

The **BGT** and the **BRT** include other types of construction than buildings, however, these are objects that can be found in public spaces. The rules of the land use plan relate to other constructions built on the parcel of the corresponding **BIM**. Therefore, it will not be possible to perform rule checking using the **BIMs** and the **SOR**.

Finally, the roads are marked as absent in table 5.4. In the key registers there are actually roads modeled, however, these are in 2D just like the buildings. To perform the rule checking, the height of the crest of the road is needed. This information is not present and therefore marked as absent.

5.2.3 Summary

The required entities from IFC are mostly present in the BIM. To include all necessary entities, the super-types (**IfcWall** and **IfcRoof**) are taken into account for the conversion. The **IfcRoof** does not hold the geometry of the roof. The geometry is in this case handled by **IfcSlab**.

The SOR, however, is lacking all entities due to consisting mostly of 2D data.

At this point, the presence and absence of entities is determined. As can be seen in this section, there is information lacking. In order to perform a conversion, it must be determined which information can be converted and which information must be included in order to enable the rule checking.

5.3 PREPARING THE CONVERSION

This section researched the sub-question: Which information from the BIM and the SOR should be converted regarding the digital building permit process? Previously, the presence and absence of entities in the input models was determined. In this section, it must be determined which information will be converted (see table 3.1). On top of that, it is to be determined whether all required information can be converted. For the first step, the required information is categorized in this table (see table 5.5).

		BIM	
		present	absent
SOR	present	-	-
	absent	walls, roof	road, other construction, and ground height

Table 5.5: Determining which information will be converted.

As it appears, some information is missing, namely the roads and the ground height. This issue has to be overcome before performing the conversion, otherwise the rule checking cannot be performed. The issue lies in the information of the SOR. The roads are modeled in 2D in the key registers, however, 3D data is required to determine the peil. To determine the peil other datasets are used.

5.3.1 Peil from 3D Basisvoorziening

In section 5.1, it was determined that all building lots were bordering the road. This means the entrance of the BIM on these location would border the road. In this case, the peil is determined by calculated the height of the crest of the road. For most locations, the 3D Basisvoorziening can be used. The 3D Basisvoorziening is a register developed and updated yearly by PDOK [PDOK, 2022]. This dataset entails the city objects from the BGT with a height component, which is exactly the information needed to determine the peil. It can be downloaded through a viewer. The dataset can be downloaded in CityJSON format.

5.3.2 Peil from dataset Rotterdam

One of the locations is in Rotterdam. The municipality of Rotterdam created a dataset with all the peil information. This dataset is provided in different formats, one of them is Shapefile. Only for the location in Rotterdam, this dataset will be used.

5.3.3 Ground height from AHN4

Since the ground height could not be linked to any entities in either standards, another dataset is used. The ground height is selected from the [AHN viewer](#). In this viewer, the AHN4 dataset is available amongst other datasets of [AHN](#).

5.3.4 Summary

The walls and the roof will be converted from the [BIM](#), while nothing could be converted from the key registers that will form the [SOR](#). As a result, the peil is retrieved in other ways. For most locations, the roads will be extracted from the 3D Basisvoorziening, which does include the height information. For the location in Rotterdam, the separate peil dataset will be used.

However, no solution is found to represent other constructions, not buildings being. This rule cannot be checked using the current datasets.

The required information from the [BIM](#) can be converted and the peil can be retrieved from other datasets. This means that a rule checking can be performed in the next section.

5.4 PERFORMING THE CONVERSION

In this section, the last sub-question is investigated: How to perform a conversion from a [BIM](#) and the [SOR](#) supporting the digital building permit process? The required information from the created [BIM](#), the peil dataset, and the 3D Basisvoorziening will be converted accordingly from table [5.5](#).

5.4.1 Data preparation

Before the conversion, the data needs to be prepared. In the case of the [BIM](#), guidelines must be applied that state how to model a correct [BIM](#) not only for exchanging the [BIM](#) between different stakeholders, but also for the conversion.

BIM The guidelines proposed by [Ohori et al. \[2018a\]](#) and [BIM Loket \[2022\]](#) are focused on the creating phase of [IFC](#) files. The guidelines can be used while [IFC](#) files are being created which is the case in this research. An important guideline that must be applied in this project is georeferencing, which is crucial for rule checking. Currently, georeferencing is often either not provided by the designer or not supported by the design software [[Diakit , 2018](#)]. Georeferencing is in fact possible in the Revit software and is performed following the actions described by [Diakit  \[2018\]](#):

- The simple [BIM](#) is created in Revit.
- The postal address in (Manage > Location) is adjusted to the correct postal address. The addresses are the locations used for the land use plans in section [5.1](#). The exact location is pinpointed at the same location as the Project Base

Point and the Survey Point of the BIM (see figure 5.4). The Survey Point is always set to the left corner of the building which is bordering the street. The used CRS in Revit is WGS84.



Figure 5.4: Pin for Survey Point located in Revit on the location Schependomlaan

- The Revit file is rotated for the purpose of the design. The actual north (*True North*) is oriented differently. The *True North* always has to be set manually as well. To correct the angle to the *True North* the BRT in the PDOK viewer is used to create a screenshot, which can be imported in Revit. Two lines are drawn and the angle between them is calculated. To rotate the project to its *True North* the correct angle must be inserted to the attribute *Angle to True North*. The angle is calculated as follows: $360^\circ - 20^\circ = 340^\circ$ (see figure 5.5).

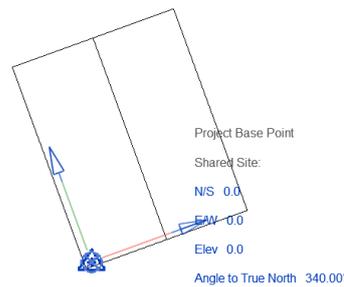


Figure 5.5: Building design rotated to True North for Schependomlaan

- The file is now georeferenced with WGS84 coordinates. Finally, the Revit file is exported as an IFC file.

5.4.2 Validation

Now that the BIM data is prepared, the IFC file can be validated before performing the conversion. The information used for this research from the 3D Basisvoorziening and the peil dataset from Rotterdam are validated as well. This is done to avoid errors during the rest of the research.

BIM

The BIMs are validated using the IfcCheckingTool. The tool checks the files against IFC schemas. This means the IFC file is checked on the correct use of entities. It does not check whether geometries are intersecting. The BIM is checked and it does not show any errors.

3D Basisvoorziening

In order to determine the peil information from the [SOR](#). Two ways to calculate the peil are implemented. For the first one, the height of the crest of the road is needed. This information is retrieved from the 3D Basisvoorziening encoded in CityJSON. Only the tiles in which the locations of the received [BIMs](#) reside, are taken into account. These tiles are firstly checked by `cjval`. All 3D Basisvoorziening files used for this research were validated without any errors. Afterwards, the files were validated with `val3dity`. During this validation, multiple errors occurred in all tiles. The errors mostly have to do with non-manifold cases, duplicate vertices, and the wrong orientation of geometries. For the purpose of this research, only the height value of a single vertex is selected instead of the whole geometry. Therefore, the dataset could still be used for this research. The errors are summarized in table [A.2](#).

Peil dataset

The peil dataset was validated in QGIS using the Geometry Checker. No errors occurred during the validation and so it can be used for the conversion.

AHN4

The AHN viewer shows all the datasets created by AHN. The newest dataset is AHN4. This dataset consists of several subsets. The one used for this research is the AHN4 DSM 0,5m. This dataset shows the buildings in the viewer, which is needed. The ground height closest to the Survey Point is selected and, therefore, the ground height nearest is selected. The DTM datasets do not show any buildings. The DSM 0,5m dataset can be downloaded in TIFF format as a tile through another [viewer](#). No validator worked and so the TIFF file could not be validated. The AHN viewer is working properly and data can be retrieved from the viewer, which will be sufficient for this research as only the height of the point closest to the Survey Point is needed.

5.4.3 Conversion

Once the input models have been validated, the conversion can be performed. This section is divided into four sections. First of all, the conversion of the [BIM](#) is discussed. The peil dataset and the 3D Basisvoorziening files are parsed to retrieve the correct information. The correct ground height is retrieved from AHN4. Finally, it is discussed what the structure and attributes of the new 3D city model are.

BIM

In the previous section, it was determined the walls and the roof should be extracted from the BIM to perform the rule checking. The conversion of the wall and roof geometries will be explained separately step by step. The goal is to create b-rep surfaces in which the vertices are ordered in counterclockwise order, since this is required in the CityJSON specifications [[Ledoux and Dukai, 2022](#)]. Afterwards, the transformation of the coordinates will be discussed.

WALLS

Retrieve external elements In order to parse the [BIM](#) a library, `IfcOpenShell` is used. Through this library, it is possible to select the required information, for example, the walls or the roof. Each building element in the [BIM](#) has `IfcPropertySingleValues` that contains information about the element. One of these is the property `IsExternal`. This can be set to either `True` or `False`. In this research, the external building elements are required, which means all walls with the property `IsExternal` set to

True, will be taken into account.

The vertices of an entity in IFC file contain X, Y, and Z coordinates and are stored as follows: [v1x, v1y, v1z, v2x, v2y, v2z, ...]. To work properly with the information, the flat list is rewritten into this format: [[v1x, v1y, v1z], [v2x, v2y, v2z], ...]. Using this format will clarify which coordinate belongs to which vertex. This makes it easier to work with in this research and it is easier to understand which coordinate belongs to which vertex.

The faces are expressed with indices referring to the position of the corresponding vertices in the list of vertices. The faces are stored as: [fv1, fv2, fv3, f2v1, f2v2, f2v3, ...]. The flat list shows that a face of an entity consists of three vertices. This means the geometry of the entity consists of triangular faces. On top of that, the flat list of the triangular faces contains indices to the vertices. This emphasizes the importance of rewriting the flat list of the vertices into the new format. Therefore, the flat list is rewritten into: [[fv1, fv2, fv3], [f2v1, f2v2, f2v3], ...]. This format is much clearer and easier to work with.

In conclusion, a wall does not consist of surfaces when IfcOpenShell is used to parse the IFC file. Instead, it consists of triangular faces with indices referring to the position of the vertex in the list of the vertices in the new format.

Rotation and translation Once the vertices of the external elements are retrieved, the faces are not placed correctly. All walls have the same origin and rotation, if only the Cartesian coordinates are taken into account (see figure 5.6).

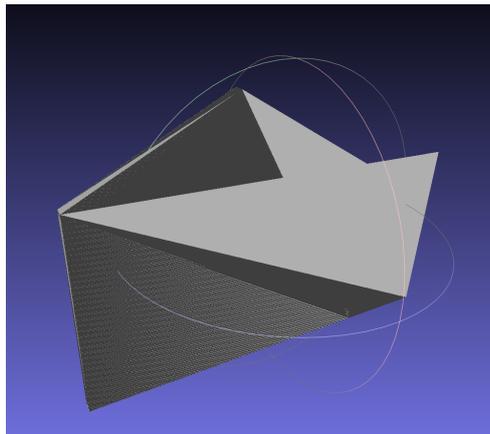


Figure 5.6: Walls before rotation and translation

Therefore, the relative placement of the elements have to be retrieved. In IFC the relative placement is represented by the entity **IfcAxis2Placement3D**. **IfcAxis2Placement3D** can have three optional attributes: *Axis* (represented by **IfcDirection**), *RefDirection* (represented by **IfcDirection**), and *P* (represented by **IfcDirection**) [buildingSmart, 2022]. *Axis* is the exact direction of the local Z axis, *RefDirection* is the attribute used to determine the direction of the local X axis, and *P* represents the normalized directions of the placement X axis, the placement Y axis, and placement Z axis.

However, when the IFC file is parsed, the *Axis* holds the local coordinates of the origin of the element. The coordinates are represented by the entity **IfcCartesian-Point**.

The attribute *RefDirection* actually represents the *Axis* in this IFC file. It is represented by the entity **IfcDirection** and is usually set to (0.0, 0.0, 1.) for the Z axis.

The third attribute found in the IFC file represents the *RefDirection* and is represented by **IfcDirection**. This attribute holds the rotation around the Z axis.

Using the information from the point of origin, the *Axis*, and the *RefDirection*, the vertices are rotated and then translated. The result can be seen in figure 5.7.

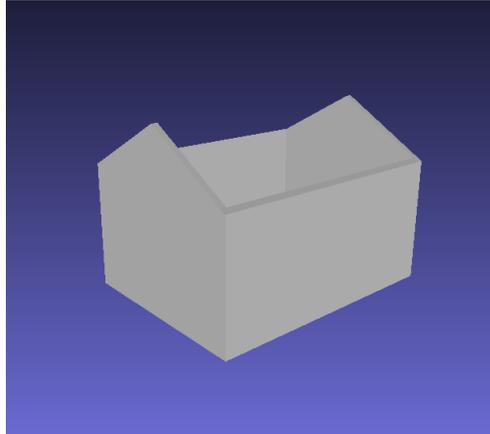


Figure 5.7: Triangular faces after rotation and translation

Horizontal and vertical faces For the walls, it is important to make a distinction between the vertically oriented and horizontally oriented triangular faces. The horizontally oriented triangular faces represent the faces attached to the ground or the roof. The vertically oriented triangular faces are the ones visible since these are covered with materials, brick on the exterior or gypsum on the interior. The importance of the distinction of the orientation will be explained in section 5.4.3.

The distinction between vertically and horizontally oriented triangular faces is made by calculating the normal vector for each triangular face. The normal vector is a vector perpendicular to the triangular face. It is calculated from the cross-product of two (non-parallel) edges of the triangular face. The result is a normal vector with values in the X, Y, and Z direction.

The normal vectors where the value for the Z direction is not zero, means that it is pointed upwards, downwards, or diagonally. In this project, only vertical walls are assumed and so all the normal vector with a non-zero value in the Z direction are removed from the selection.

Create surfaces At this point, the selection of triangular faces consists of vertical triangles. However, the end result should be b-rep surfaces. To be able to model surfaces, the triangular faces need to be extended to surfaces. The wall consists of several surfaces, since it is created as a Sweep Volume in IFC. This means that each wall in the simple BIM consists of six surfaces.

To model these surfaces, coplanarity is used. For every triangle, the coplanarity is calculated for each vertex in the IFC file. If the vertex is coplanar, it is added to the triangle. If it is not coplanar, nothing is done. This process is repeated until all triangles are checked. This step will result in a list of lists of vertices that are coplanar to each other. For the simple BIM, eight vertically oriented surfaces are created. These are not all surfaces that face the outside, but also surfaces that face the interior of the BIM.

Select the outer surfaces At this point, only the vertical surfaces of the entities are selected (see figure 5.8).

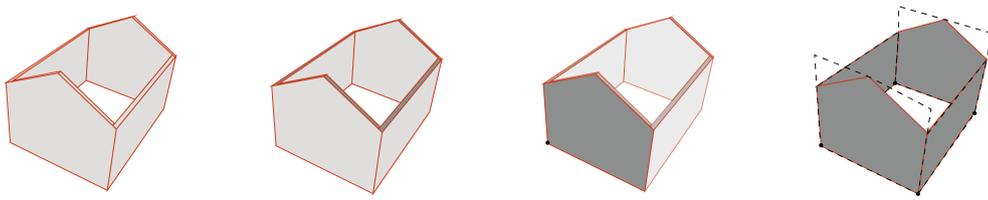


Figure 5.8: Selecting the outer surfaces: (1) The surfaces of the BIM, (2) Removing horizontally oriented surfaces, (3) Find the vertex closest to bottom left corner and calculate the bounding box of the surface it belongs to, (4) Repeat for each outer surfaces, until all are found.

An extra step in the conversion is needed for the walls. The walls in IFC are most often created as a Sweep Volume. This means a shape, rectangular in this example, is extrapolated along a curve which can be drawn manually. Although the external walls are retrieved in the first step of the process, the external walls also include internal parts. This is clearly visible in figure 5.9.

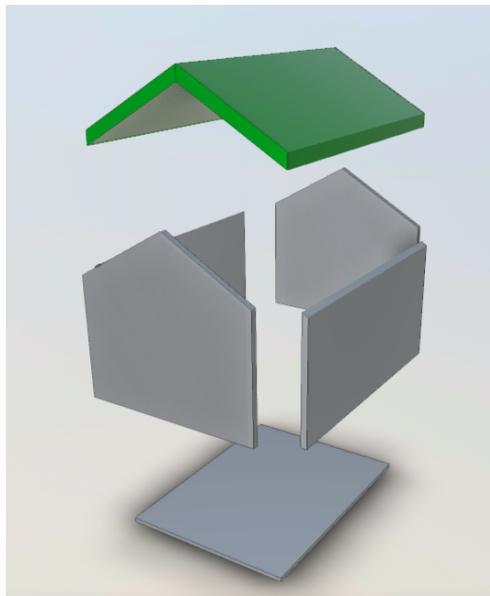


Figure 5.9: Exploded view of Simple.ifc

The first step to select only the outer walls, is to calculate the bounding box of all the walls. For this research, the bottom left corner of the bounding box is selected and simple rectangular footprints of buildings are assumed in order to select the corner vertex of the IFC file. For Simple.ifc, any other vertex at the bottom of the bounding box should work, but this is not the case for more complex shapes.

The next step is to find the vertex of the IFC file closest to the bottom left corner of the bounding box. When it is found, it is selected as a starting vertex to find the other outer surfaces. This assumption is safe in this case, since it is sure that this vertex belongs to an outer surface (see figure 5.10). Then it is checked to which surface this vertex belongs to.

Now that the first outer surface and starting vertex are determined, the other outer surfaces can be determined (see figure 5.10). For each surface, a bounding box is created to check whether the starting vertex is present. If so, the surface is added to the outer ring and the starting vertex is set to the next vertex at the bottom within in

the bounding box. The next vertex is now the bottom left corner of a new adjacent wall. This process is repeated until no more elements can be found.

It is important that the horizontal faces are left out of the selection of walls, since this process assumes the adjacency of the outer surfaces. The horizontal surfaces are also adjacent to the outer surfaces. Therefore, these must be left out in order to retrieve only the outer vertical surfaces. As a result, a list of vertical outer surfaces of the walls is created.

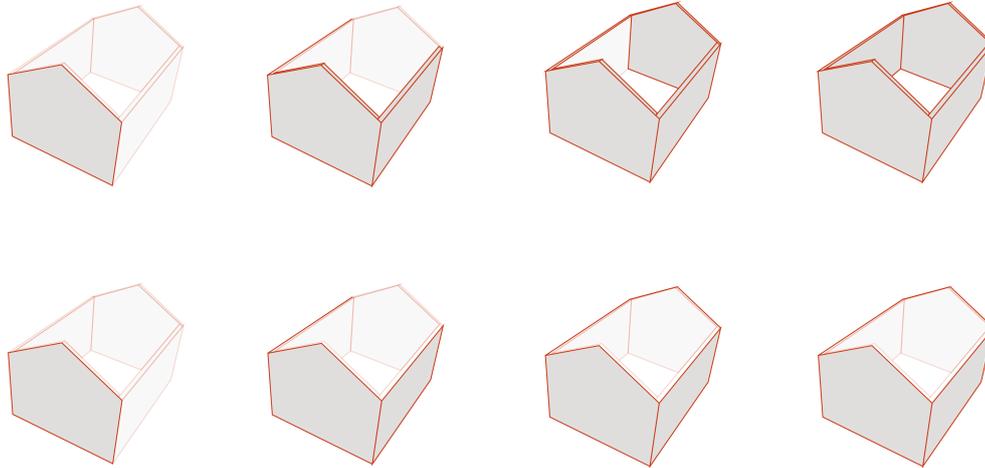


Figure 5.10: Upper row: if the horizontally oriented surfaces are included, this will lead to internal surfaces being added, bottom row: without the horizontally oriented surfaces, only the outer surfaces are added.

B-rep surfaces Finally, the vertices of the surfaces must be ordered counterclockwise in order to be represented in CityJSON. The creation of b-rep surfaces is done in two steps: the orientation of the triangular faces and the merging of the faces so that the vertices of the outer surfaces are in counterclockwise order.

First of all, the orientation of the triangular faces must be set correctly. An overview of this process is given in figure 5.11.

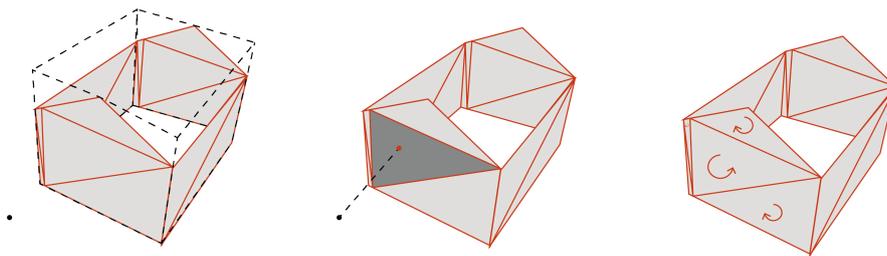


Figure 5.11: Orientation triangular faces: (1) Create viewpoint outside of bounding box, (2) Determine orientation triangular face nearest by the viewpoint, (3) Set closest triangular face to counterclockwise orientation and set orientation of neighboring faces to the opposite orientation.

The first step is to create a viewpoint outside the bounding box of the triangular faces. The distance between the viewpoint and the centroid of the triangular faces is calculated. The shortest distance is picked and checked on its orientation using the viewpoint, the centroid, and the normal vector of the triangle. The orientation must be counterclockwise. If the vertices are in clockwise order, the order of vertices is flipped. This triangle will be set to the starting triangle for the rest

of the triangular faces and is added to a new list. For the starting triangle, all its neighboring triangles are checked on orientation, which must be the opposite of the starting triangle. If the orientation of all triangles is set correctly, these are added to the new list containing the triangles with the correct orientation. The next triangle is picked from this list and the process is repeated until all triangles are checked.

Now that the orientation of the triangles is set correctly, the next step can be executed (see figure 5.12). For each outer surface, its corresponding outer triangular faces are found. From this set of outer triangular faces, a triangle is picked which orientation is counterclockwise and is set as the starting triangle. The vertices are added to a new list containing the vertices in the correct order of the b-rep surface. The starting triangle is checked for adjacent triangles in the same outer surface.

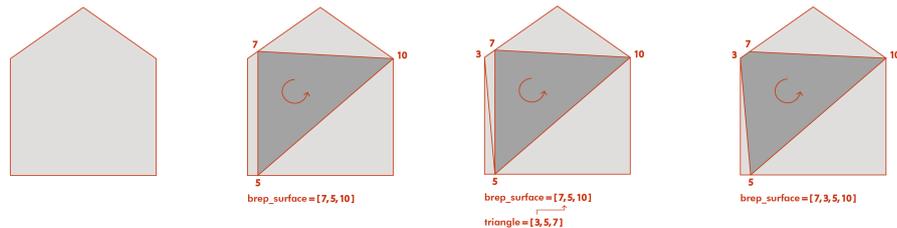


Figure 5.12: Creating b-rep surfaces: (1) One of the outer surfaces, (2) Starting triangular face with counterclockwise orientation, (3) Determine the position of the vertex that is not already included in the b-rep surface, (4) Vertex is added between vertices of the b-rep surface.

The indices of the adjacent triangles present in the list of vertices of the b-rep are saved. This is important for determining the position of the remaining index, which is new, and must be added correctly to the b-rep surface. If the present indices are next to one another, the remaining index must be located between the present indices in the b-rep surface. If the present indices are the first and the last in the list, the remaining index is added at the end of the list. The end result is a list of vertices ordered in counterclockwise order per outer surface.

GROUND SURFACE The ground floor was not mentioned in any of the rules. However, it is vital to include it in the 3D city model, since the model must be watertight [Ledoux and Dukai, 2022]. This means there may not be any holes, for example. The ground surface is created by selecting all the bottom corner vertices of the outer walls. These vertices form the footprint of the wall and form the ground surface.

ROOF The roof has a similar process as the walls. It is somewhat different and therefore the steps that are different from the process of the walls are discussed.

Retrieve external elements **IfcSlabs** can represent the roof. However, these can also be used to represent the floors. It is important to select the slabs, where *IsExternal* is set to True, since these slabs represent the roof. From the external **IfcSlab**, the vertices and the triangular faces can be extracted. The placement from **IfcRoof** is used to translate the coordinates of the slabs.

Rotation and translation The relative placement of the roof and the geometry of the roof is stored in separate places. The entity **IfcRoof** contains the relative placement. The roof consists of slabs, which are represented by **IfcSlab** and contain the geometry. The rotation (*RefDirection*) is not given for the roof as it is orientated correctly. The origin is stated by the entity, **IfcCartesianPoint**. This information is passed down to the slabs that represent the roof.

Create surfaces The creation of surfaces is done similar to the approach applied to create the walls.

MERGE WALLS AND ROOF At this point, the wall, roof, and ground surfaces are created. The original triangular faces are written to an OBJ file (see figure 5.13). This is done to view the triangular faces of the BIM. This simple OBJ file contains all the vertices used in the BIM and all the triangular faces, but it does not hold any semantic information.

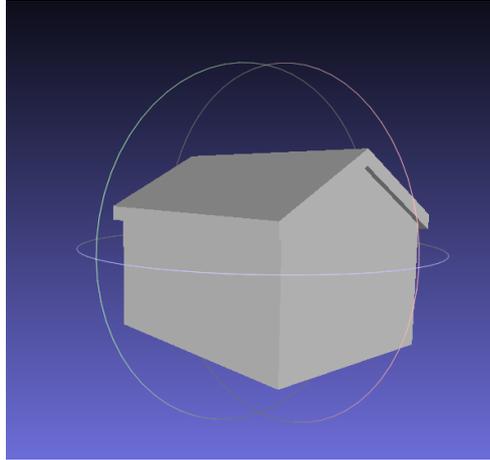


Figure 5.13: OBJ visualization of Simple.ifc

The roof surfaces are all modeled, but the roof is overlapping the walls (see figure 5.13). In CityJSON, only b-rep surfaces must be used. This means parts of the roof surfaces must be cut out where it is attached to the outer surfaces of the walls. A hole is created in the roof surfaces on the vertices where the wall surfaces are attached to the roof surfaces. This is done in several steps (see figure 5.14). First of all, the downward oriented surfaces are selected by their normal vectors. Once the surfaces are found, a bounding box per surface is created. All the wall vertices inside the bounding box are selected. The vertices are checked whether they lie on one of the edges of the surface. If so, the vertices are added in a counterclockwise order to the roof surface. In doing so, the roof connects to the wall and the model is watertight. It is important to note that it is assumed the walls are attached to the roof surfaces. This can be checked in Revit manually.

COORDINATES Another issue that must be overcome in the conversion, is the CRS. The BIM is georeferenced, however, only the Survey Point is assigned a *RefLatitude*, *RefLongitude*, and an *Elevation*. On top of that, the *IfcSite* is assigned with a rotation towards the *True North*.

Latitude and longitude The latitude and longitude information in Revit are, WGS84 coordinates, while all maps in the Netherlands, including the *SOR*, have RD-coordinates. A translation from one CRS to another is needed.

The latitude and longitude information can be retrieved from the IFC file by selecting the *IfcSite* in which the whole building resides (see table 5.6). The latitude and longitude are however, described in degrees, minutes, seconds. To change from one CRS to another, the latitude and longitude must be rewritten to decimal degrees.

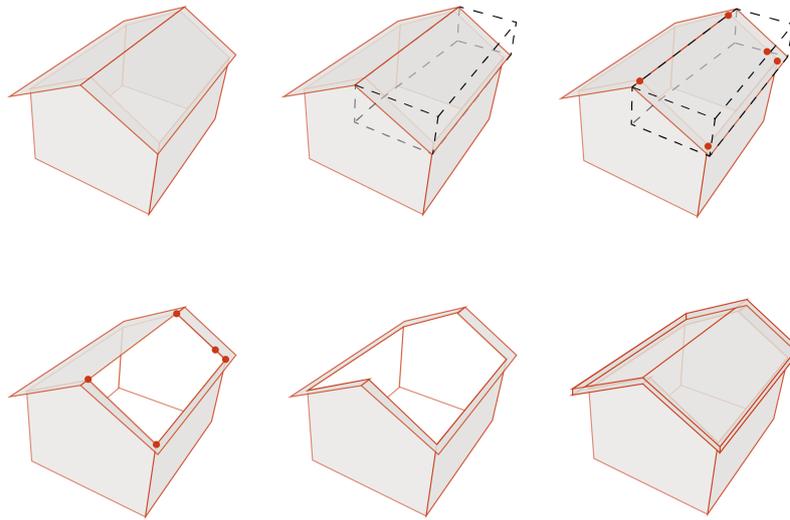


Figure 5.14: Merging the roof and wall surfaces: (1) select downwards oriented roof surfaces, (2) create a bounding box of this surface, (3) select the wall vertices that lie within the bounding box, (4) add the points in counterclockwise order to surface, (5) apply to all downwards oriented faces, (6) result.

BIM	Latitude	Longitude
Schependomlaan	51,50,30,345153	5,50,10,186958
Haviklaan	52,4,50,66528	4,15,28,166198
Rabarberstraat	52,4,11,792907	4,16,9,558906
Witte de Withstraat	52,4,44,751892	4,17,32,133178
Wijndaelersingel	52,3,28,218383	4,13,32,444114
Jufferstraat	51,54,57,697448	4,29,12,796554

Table 5.6: Latitude longitude (DMS) of Survey Point

True North The BIM is rotated towards its *True North*. The rotation can be extracted from the `IfcDirection` of the `IfcSite`. The entity usually has values between -1 and 1 as can be seen below:

```
#149= IFCDIRECTION((0.939692620785911,0.342020143325663,0.));
```

The degrees can be calculated by taking the inverse cosine of the first element of `IfcDirection`.

BIM	Rotation (degrees)
Schependomlaan	20
Haviklaan	148
Rabarberstraat	118
Witte de Withstraat	38
Wijndaelersingel	120
Jufferstraat	109

Table 5.7: Rotation to True North

Transformation Now that the coordinates are known, the library `pyproj` is used to transform the WGS84 coordinates to RD-coordinates using their respective EPSG codes (EPSG:4326 for WGS84 and EPSG:28992 for the RD-coordinates).

However, at this point only the Survey Point of the BIM is transformed. The rest

of the vertices are transformed respectively to the Survey Point. This can be done using the member **"transform"** in CityJSON. The member consists of two factors: *scale* and *translate*. Both are represented by an array of three values for each dimension. The array of *translate* represents the Survey Point in RD-coordinates in this research.

In order to write the correct coordinate to the CityJSON file, the correct order must be applied: scale and then rotate. However, the first step is to calculate the difference per vertex regarding the Survey Point. These differences are then divided by the scale. Finally, the vertices are rotated with the angle in degrees of the *True North* from table 5.7.

Peil dataset

The municipality of Rotterdam created a file containing the information of the peil. This file is read and a selection of the information is retrieved, namely the object, its X coordinate, its Y coordinate and the corresponding peil value. The vertex closest to the Survey Point of the BIM is selected and its peil value retrieved. The coordinates are already in RD-coordinates.

3D Basisvoorziening

The 3D Basisvoorziening consists of three files. One of those three includes the geometries of roads, which is called the 3D Basisbestand Volledig. 3D Basisbestand Volledig is written in CityJSON and consists of several city objects, namely **LandUse**, **GenericObject**, **Building**, **PlantCover**, **Bridge**, **Road**, and **WaterBody**. The city objects of type **Road** are selected. The Roads are further filtered, since there are many subtypes that belong to road. The classification is based on the classification of the BGT. The only subtypes that must be taken into account are:

- rijbaan autosnelweg
- rijbaan autoweg
- rijbaan regionale weg
- rijbaan lokale weg

The vertices of these roads are used to calculate which is closest to the Survey Point of the BIM. The height of this vertex is used as the peil value. The roads are modeled with an LoDo [PDOK, 2022] which means the height is the same over the whole object [Biljecki et al., 2016]. However, the closest vertex to the Survey Point is selected, since it belongs to the road the BIM will border.

AHN4

From the AHN4 dataset, the ground height is retrieved by selecting the point from the **AHN viewer** closest to the Survey Point. An overview of the ground height is given in table 5.8.

BIM	Ground height (m)
Schependomlaan	10,96
Haviklaan	3,51
Rabarberstraat	1,18
Witte de Withstraat	1,17
Wijndaelersingel	1,38
Jufferstraat	2,73

Table 5.8: Ground height closest to the Survey Point from AHN4

3D city model

At this point, the geometries and semantics of the entities from the simple BIM are processed as well as the coordinates. The 3D Basisbestand Volledig and the peil dataset are parsed so that the peil value can be assigned to the Survey Point. Besides, a ground height is assigned to the Surecy Point.

GEOMETRY The type of geometric primitive of the city object is set to *MultiSurface*. *MultiSurface* is a set of planar surfaces. The surfaces themselves are stored in the member "**boundaries**". These are arrays which can consist of two arrays if an interior ring is being added. Otherwise, only one array is stored. The surfaces are indicated with indices referring to the position of vertex in the member "**vertices**".

In the member "**vertices**", the coordinates of all vertices of the 3D city model are included relative to the Survey Point. In the member "**transform**", the transformed coordinates of the Survey Point are stated in the attribute *translate* and scaling was applied within the attribute *scale*.

SEMANTICS A feature in CityJSON regarding the semantics is the use of semantic surfaces. Under the member "**geometry**" of a city object, semantic information can be added. Each type of surface and additional information can be added to a dictionary. The boundaries of the city object can then be linked to the semantic surfaces using indices. In this research, the types created according to the documentation by Ledoux and Dukai [2022] are: *WallSurface*, *RoofSurface*, and *GroundSurface*. The result is visualized in ninja (see figure 5.15). The *WallSurfaces* are colored white and the *RoofSurfaces* are colored red due to the semantic surfaces included in the model.

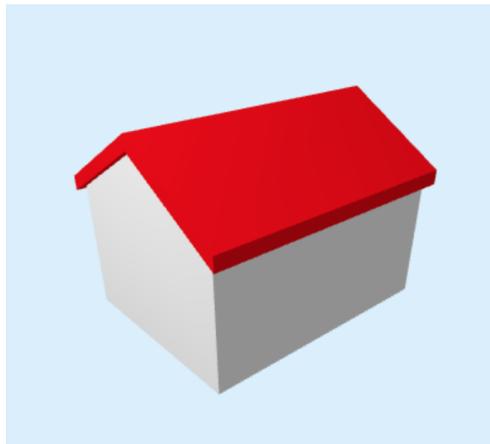


Figure 5.15: 3D city model visualized in ninja

PEIL VALUE The peil value is added to the 3D city model. The building under CityObjects. Each city object can hold attributes, which can be added freely [Ledoux and Dukai, 2022]. The peil value is added as an attribute to the city object.

GROUND HEIGHT The ground height is added to the 3D city model as an attribute, just as the peil value.

5.4.4 Validation

The newly created 3D city model based on the simple BIM consists of four *WallSurfaces*, one *GroundSurface*, and two *RoofSurfaces*. The 3D city model has one attribute, namely the peil value. The validation of the 3D city model is done to have a correct

result of the automatic rule checking. It is done simultaneously as the writing to detect errors immediately, so these errors can be debugged in the tool.

The validation in this research consists of two validation methods. The first method checks the model on its geometric primitives using `val3dity`. The following command is prompted in the terminal.

```
val3dity [path to file] --report [path to report]
```

It will produce a report in JSON format containing all errors. `val3dity` does not check the model on the CityJSON schemas [3D Geoinformation (TU Delft), 2020]. Therefore, another validator is used through `cjval`. The model can be validated using the following command in the terminal:

```
cjval [path to file]
```

The 3D city model passes both validations and can be used for the rule checking.

5.4.5 Rule checking

With a correctly validated 3D city model, the rules can be tested as formulated in section 5.1.

THE MAXIMUM BUILDING HEIGHT To calculate the maximum building height of the new building, information from the 3D city model and from the land use plan is needed. From the 3D city model, the peil value, the ground height, and all the height values of the model are extracted. The total height of the new building is calculated by adding the highest Z coordinate of the building to the peil value.

The land use plan corresponding to the location is used. The maximum height stated in the land use plan and the peil value is added to set the maximum height which cannot be exceeded by the BIM.

Afterwards, the height of the BIM combined with the ground height is compared to the maximum height stated in the land use plan.

If the height of the 3D city model is lower than the maximum height, it satisfies the rule. If the height is higher, there are two options. The first one takes a threshold of 10% of the maximum height in account. If the height of the 3D city model lies within this range, a deviation within the land use plan is the result (in Dutch: binnenplanse afwijking). If the height exceeds the maximum height with the threshold, it does not satisfy the rule. The results are written to a report in JSON format. `Simple.ifc` satisfies the rule on each location and this is stated in the report like this:

```
{
  "building_id": "3XgvSNiMzD7BG3JQ5TA73n",
  "rule": "Maximum building height",
  "maximum building height in rule": 23.92,
  "current building height": 17.44,
  "result": "Building satisfies the rule."
}
```

THE MAXIMUM GUTTER HEIGHT Checking the maximum gutter height needs slightly different information. The gutter is attached to the lowest part of the roof. Therefore, the vertices that belong to the roof surfaces as well as the Survey Point and the peil value from the 3D city model are needed. From the corresponding land use plan, the maximum gutter height must be retrieved.

The lowest height value of the roof surfaces combined with the ground height is added to the height value of the Survey Point. The peil value of the 3D city model

is added to the maximum gutter height from the land use plan. These two heights are checked against each other. The result is treated the same as for the maximum building height. Simple.ifc satisfies the rule on each location, which can be seen in the report:

```
{
  "building_id": "3XgvSNiMzD7BG3JQ5TA73n",
  "rule": "Maximum gutter height",
  "maximum gutter height in rule": 19.92,
  "current gutter height": 14.96,
  "result": "Building satisfies the rule."
}
```

5.4.6 Summary

First of all, Simple.ifc is georeferenced to all locations of the received BIMs. The lower left corner of the building when seen from above, is used as the Survey Point and the BIM is rotated to its *True North*. Afterwards, it is validated and no errors occurred.

The 3D Basisvoorziening Volledig is checked. No errors occur using `cjval`, but they did when using `val3dity`. The peil dataset is checked in QGIS. The AHN4 dataset could not be validated, however, the viewer is working properly and the required information for this research can be retrieved from this dataset.

The walls and roof from the BIM could be converted from triangular faces to b-rep surfaces in CityJSON. The peil values could be extracted from the peil dataset from the municipality of Rotterdam and the 3D Basisvoorziening Volledig. The information combined produced a 3D city model containing one city object, the building.

This city object contains two attributes, which is the peil value and the ground height. The b-rep surfaces are classified as semantic surfaces. Doing so, the wall, roof, and ground surfaces can be distinguished from one another. The distinction becomes clear when the 3D city model is visualized.

This 3D city model is then used to check the rules determined in section 5.1, namely the maximum building height and the maximum gutter height. The semantic surfaces are practical when the maximum gutter height must be checked. When using the 3D city model as an input for the rule checking, the result can be one of the following:

- Building satisfies the rule
- Building less than 10 % above the rule
- Building does not satisfy the rule

Simple.ifc satisfies both rules on each location.

The results show that the required information from a simple BIM can be converted into a 3D city model, which can be used for automatic rule checking. In the next chapter, the implications of this research and automatic rule checking could have in general are stated.

6 | IMPLICATIONS

The tool created in this research to perform automatic rule checking is able to successfully check a 3D city model for the rules stated in the land use plans. Implications are drawn from the results of the research indicating the effects of this research in the digital building permit process and in the integration.

6.1 BUILDING PERMIT PROCESS

In this sub-section, it is analyzed what the effects of automatic rule checking in the digital building permit process are.

6.1.1 Transparency

It is important to note that this tool could be used at different stages in the building permit process. First of all, the tool can be used before requesting a permit by architects. The BIM created by the architects can be used directly and the report will show whether the BIM satisfies the rules. Since the permit is not requested yet, the BIM can easily be adjusted to satisfy the rules. Checking the BIM before requesting the permit can avoid issues and delays later in the building permit process.

Municipalities can use the tool in two stages. When the permit request with the BIM has been received, it can be checked immediately by the municipality. If it satisfies the rule or the results lie within 10%, the digital building permit process can be continued. It is possible for the municipality to share the results of the automatic rule checking in the publication of the application. In the decision stage of the municipality, the results of the automatic rule checking will be taken into account.

For other stakeholders it is interesting to use this tool as well, for example the neighbors. For other stakeholders, the tool is useful when the application has been published by the municipality. There are two scenarios of publications explained. In the first scenario, the tool can be used by other stakeholders at this stage of the digital building permit process. The other one shows only the results of the report of the automatic rule checking.

Tool

There are some issues, when it comes to other stakeholders using this tool. The main one is information that is not available to other stakeholders. For example, the BIM should be shared with stakeholders in order to use the tool. The peil value could be shared, however, it could also be retrieved from the 3D Basisvoorziening. This value is an estimate and can be inaccurate. A solution to this issue could be the creation of a national peil dataset. This could be an extension of the peil dataset created by the municipality of Rotterdam. Users can download a relevant tile of the dataset and acquire a peil dataset manually from QGIS, but some knowledge is required to use this software.

Report

The second scenario includes the transparency of the report. The results of the report are shared by the municipality to other stakeholders. Doing so, the stakeholders have clear insight into whether the building satisfies the rules or not.

Both scenarios have drawbacks and advantages. The first scenario is more transparent, since the input and output models can be visualized and other information is known, for example, the peil value. A national peil dataset would offer more transparency, however, it would cost time and money to create a dataset as such and municipalities already have their own dataset of peil values. Even if the peil value could be retrieved from a national peil dataset, the BIM also has to be acquired from the architects, who also can acquire the peil value more easily from the owners of the building lot.

The second scenario is less transparent and the models cannot be visualized. However, the publication of the results of the report can easily be done by the municipality and acquires insight to other stakeholders. Presenting the results of the automatic rule checking is more transparent than only publishing that there is an application for a permit. Both scenarios provide more transparency in the digital building permit process for all stakeholders.

6.1.2 Municipalities

If a perfectly working tool were to be used in the digital building permit process, it would change the way rules are checked at the municipality. First of all, the rules stated in the land use plan need to be translated into computer-readable checks. Currently, these are written for humans to interpret. These rules can often be interpreted in different ways, as stated by Noardo et al. [2020a] and in A.2.2. If a tool was to be used for automatic rule checking, the rules should be written unambiguously and computer-readable. The land use plans and its rules are also updated every couple of years. The writing and updating of rules must be done by the municipality or done for the municipality.

On top of that, there will be a shift in several jobs. The tool takes over the checking of the rules, which takes away a part of the jobs of employees of the municipality. The job will not consist of checking the rules, but checking whether the results are correct. It could also lead to a lower amount of employees (see section A.2.1), since less work force is needed when the result only need to be checked.

6.1.3 Process

Finally, the digital building permit process itself will be more efficient. On the one hand, the municipality does not manually check the rules anymore. Only the results of the automatic rule checking are checked by the municipality, which means less time is spent and the result is less prone to errors.

Errors in the BIM can be detected at an early stage. Since the checking can be done before a permit is requested or directly after the request, the architects can adjust the design at this stage. Errors occurring at a later stage, will cost more time and money to correct.

If architects and municipalities utilize this tool, confusion and disagreements can be avoided. The rules of land use plans are sometimes open for interpretation, however, the rules in the tool are already rewritten so that these are unambiguous. Besides this tool being unambiguous, it also enhances the transparency of the process. The

report produced by this tool shows which rules are being tested and whether the rules are satisfied or not.

6.2 INTEGRATION

Besides the effects of this tool in the digital building permit process, it bears consequences for the integration between the AEC and the geospatial domain. This research shows that these two domains do not align at all times. At the moment, there are many experts in the field, at universities, and companies belonging to one of the domains. Most often they do not have knowledge of both domains. Currently, it is known that several use cases benefit from the combined knowledge. For example, this tool can only be developed when knowledge from both domains is combined. Experts, companies, the government, and more have to cooperate in these use cases to combine knowledge in order to work with certain use cases. For use cases to be solved more efficient like the tool created in this research, knowledge should be combined and spread. There are two target groups that must acquire the combined knowledge.

The first one consists of students. The knowledge can be spread in education by dedicating lectures, projects, or entire courses to integrating both domains. This means that both students in both domains should acquire knowledge about the other domain. Students of the AEC domain learn the basics and possibilities of the geospatial domain and vice versa. This will result in students with combined knowledge that eventually will enter employment.

The second target group are experts already working in one of the domains. Knowledge is spread by use cases, since experts are working together and sharing knowledge in these use cases. It can also be spread at fairs or lectures in both domains. At these fairs, lectures are given which can be followed by experts attending the fair.

Once students and experts have knowledge about the other domain, the possibilities of further integration can be explored. Integration is data operability between the two domains [Zhu et al., 2018], which means that the data operability will increase. Besides, multi-disciplinary challenges can be solved more easily with further integration [Ohori et al., 2017].

This chapter described the implications of automatic rule checking for the digital building permit process and for the integration. The next chapter provides a discussion based on the results and experiments of this research.

7 | DISCUSSION

The aim of this research is to overcome current issues in the building permit process by performing automatic rule checking based on a 3D city model. This research shows a 3D city model can be created by converting information from the BIM and that it can be checked on rules from the land use plans. In this chapter, a discussion based on the results will be provided. Some of the issues occurring during the research are discussed in the following sections. Afterwards, a reflection is made regarding the building permit process and the domains. Finally, the limitations of this research are described.

7.1 PARSING

When parsing the created BIM, some errors appeared regarding wrongly set attributes of entities. When the entity was parsed, other information was retrieved than stated in the documentation of IFC [buildingSmart, 2022].

7.1.1 Relative Placement

The first entity that was encountered with this issue is the **IfcAxis2Placement3D**. This entity holds three attributes: *Axis*, *RefDirection*, and *P*. When this entity was parsed using the *IfcOpenShell* library, other information was retrieved than expected from the IFC documentation. In practice, the first attribute of **IfcAxis2Placement3D** are the local coordinates of the origin of the entity. The second attribute represents *Axis* and the third one *RefDirection*.

7.1.2 True North

During this research, the BIM had to be rotated towards the true north. This was done using the **IfcDirection** of the **IfcSite**. This entity consists of three attributes, namely *DirectionRatios* for three axes. However, it was discovered using information stated by Bossche [2017], that the first element of the **IfcDirection** contained the angle of rotation towards the *True North*. *Simple.ifc* is saved in multiple versions to test whether the IFC schema plays a role in this. The most used version of IFC is IFC2x3 and the newest is IFC4x1. Both versions hold the rotation to *True North* in the first element of **IfcDirection**.

7.2 ROUNDING

When the vertices are retrieved from the IFC file, they are rotated and translated accordingly. However, the coordinates of the vertices are also rounded. This is done in order to find the coplanar vertices per triangle. With a higher accuracy, the coplanar vertices were not always detected and so surfaces could not be created. By adding rounding to the vertices, these were detected as coplanar.

The amount of digits can be set manually. If the amount of digits is too low, vertices could not be detected as coplanar. In conclusion, the rounding has a large effect.

7.3 MISSING LINK

In section 5.1, the land use plans of the received BIMs were analyzed. From these land use plans, the most frequently used rules are selected for the automatic rule checking. Some rules could, however, not be checked after all. One of those rules is the maximum height of property division.

To check the rule, the property division itself must be linked to an entity in IFC or CityJSON. There was no entity in IFC that corresponds to property divisions. This is because it is mostly not included in building designs. In CityJSON however, it could be modeled under several entities. For example, under the city object **OtherConstruction** with the type *Fence* or under the city object **PlantCover**. There is not one solution to model it, which makes it difficult to link it.

On top of that, the maximum height of property division is not a rule that is checked in large numbers by the municipality. Therefore, this rule was not taken into account.

7.4 PEIL

There are two issues regarding the peil. The first one has to do with the given definition in the land use plans. The peil is determined by the entrance of the BIM. If the BIM is bordering the road directly, the height of the crest of the road must be set as the peil. If the BIM is not bordering the road directly, the average ground level must be set as the peil. And in other cases, the ground level is stated in NAP.

When a building is bordering the road directly, it is not mentioned in the definitions of the land use plans. It is also not mentioned how to calculate the average ground level of the building. This leaves the rule open for interpretation.

The second issue has to do with information about the peil. The peil is determined from two different datasets in this research. The 3D Basisvoorziening is used to retrieve the height of the crest of the road instead of the BGT, since the BGT is mostly 2D data. However, roads are modeled with an LoDo, which means that the whole geometry of the road is extruded to a single height [Biljecki et al., 2016]. If buildings are bordering, a road in a hilly landscape, the actual peil value can differ quite a lot from the height extracted from the 3D Basisvoorziening.

Several municipalities offer more accurate information. The municipality of The Hague created a **web interface** where parcel owners can request the peil value relevant to their parcel. Within 15 days the municipality provides the peil value.

The municipality of Rotterdam created a dataset of peil values. Compared to the 3D Basisvoorziening it has much more fluctuations. For each parcel the peil is determined by selecting the highest Z coordinate of the road segment which is adjacent to the parcel.

7.5 GEOREFERENCING

Georeferencing is an important part of using the BIM in the building permit process. There are still some issues georeferencing BIMs using Revit. If an IFC file is opened in Revit, building elements are unjoined or even deleted. Because of this, informa-

tion is lost. Therefore, it is not practical to georeference existing IFC files in Revit.

Another issue that stems from georeferencing, is the precision. The Survey Point can be pinpointed on a location in the map (see figure 7.1). After the coordinate

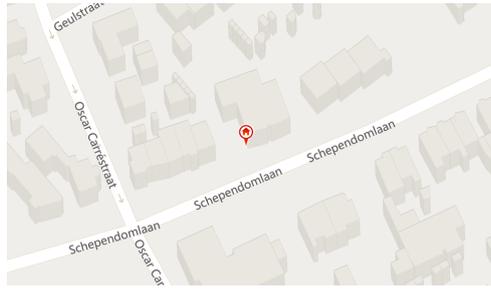


Figure 7.1: Pin located on Survey Point in Revit on the location Schependomlaan

transformation, the latitude and longitude are expressed in RD-coordinates. The results are checked in a viewer and can be viewed in figure 7.2. There is a differ-

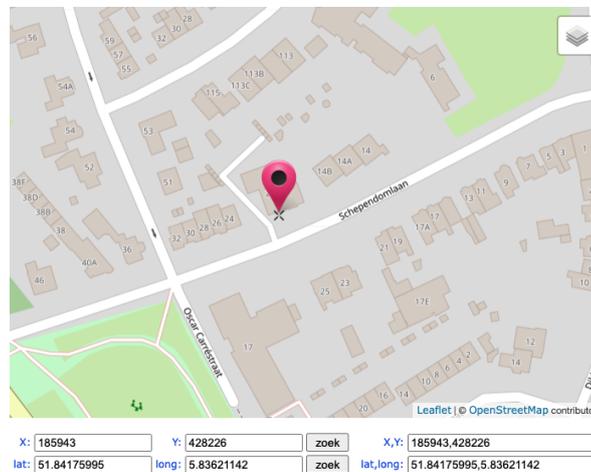


Figure 7.2: Coordinates pin checked in viewer.

ence in the location compared to the map provided in Revit. It is unclear where the offset stems from. The latitude and longitude in WGS84 coordinates are checked in the viewer. These results are equal to the ones shown in figure 7.2. The latitude and longitude in WGS84 coordinates are also checked in Google Maps. These results are equal to the viewer. In the following table the difference between the Survey Point translated from Revit and in reality are stated. On each location there is a noticeable

	Revit	Viewer	dx (m)	dy (m)
Schependomlaan	(185943, 428226)	(185938, 428227)	4	1
Haviklaan	(77595, 455321)	(77584, 455297)	11	24
Rabarberstraat	(78359, 454107)	(78355, 454102)	4	5
Witte de Withstraat	(79954, 455102)	(79941, 455095)	6	7
Wijndaelersingel	(75337, 452808)	(75340, 452803)	3	5
Jufferstraat	(93057, 436777)	(93059, 436776)	2	1

Table 7.1: Deviation in X and Y dimension for each location of the Survey Point from Revit and the viewer.

difference. The size of the differences differs as well. The smallest differences can be found on the Jufferstraat, while the largest is found on the Haviklaan. Since the

coordinates of the latitude and longitude are one the same location in the viewer as in Google Maps and the results in the viewer are sometimes near the Survey Point selected in Revit, the map in Revit could be the cause of this slight offset. However, it is uncertain whether the map in Revit is the cause of the offset.

7.6 REFLECTION

In this section, a reflection is made, which will state the drawbacks and advantages of decisions made in this research. A reflection is made regarding the digitalization of the building permit process in general. Besides, the drawbacks and advantages of performing automatic rule checking in the geospatial domain are weighed against the [AEC](#) domain.

7.6.1 Digitalization

This tool enables further digitalization of the building permit process. There are drawbacks and advantages whenever a tool like the one created in this research will be in use. The first drawbacks are the rules that need to be rewritten so that these are unambiguous and computer-readable. A lot of effort needs to be put into this. Note that there are many rules in a land use plan and a municipality has several land use plans. This means that there is a large amount of rules that need to be translated.

Since rules can be translated to a computer-readable format, it can be combined with datasets. As a result, more rules can be checked than before (see section [A.2.1](#)). A tool like this does not have to be that much faster, if there are a lot of rules being checked that were not checked before.

The tool takes over tasks that are currently fulfilled by employees of the municipality. If a more extended tool than the one produced in this research is created, employees have to spend less time on checking. It could also mean that fewer employees are needed eventually (see section [A.2.1](#)).

On the other hand, less errors occur once the rules are written correctly for the computer. If less errors are made, less delays will happen in the digital building permit process. This saves time, and money for the municipality, the architects, and the builders.

On top of that, a lot of rules can be checked simultaneously, which makes it a lot quicker than if an employee would check the same set of rules manually.

This tool can also prevent delays by letting architects use this tool before requesting a permit. The architect can check the [BIM](#) by themselves and correct it, before requesting a permit.

7.6.2 Domain

In this research, automatic rule checking is performed in the geospatial domain. The rules checked in this land use plan are solely focused on the building itself, except for the peil value and the ground height. Rule checking in the [AEC](#) domain could therefore also be possible, by converting the rules into a 3D land use plan into the [BIM](#) as done by [Dijkmans and van Berlo \[2013\]](#). This has an advantage over automatic rule checking, namely, that it can be done visually. The municipalities can visually check the [BIM](#) in the 3D land use plan. There is still a drawback to this, since employees need to learn to work with the software and this software has to be purchased by the municipality. This will already be an issue when the Environment and Planning Act will be established (see section [A.2.2](#)).

There are two reasons to prefer automatic rule checking in the geospatial domain above the AEC domain. The first reason is the automation of this tool. [Dijkmans and van Berlo \[2013\]](#) show that it is possible to create a 3D land use plan and load it together with the surrounding geometries into a BIM. However, the 3D land use plans must be loaded into BIM software and then checked manually. Doing so, only the downside of information will be avoided. The main advantage of the tool created in this research, is that only the BIM must be set as an input correctly.

The second reason is that this tool can check multiple rules at the same time. 3D land use plans need to be opened in BIM software and checked one by one by the municipality. The tool takes the BIM and checks several rules at the same time, which shortens the duration of the process.

7.7 LIMITATIONS OF THIS RESEARCH

In this research, a tool was created that could convert information from a simple BIM encoded in IFC to a 3D city model encoded in CityJSON. Some assumptions are made in this research to perform automatic rule checking, which can potentially limit the research for future use.

7.7.1 Straight walls

For this research, only straight walls can be assumed when selecting the outer surfaces. To create a subset of solely the outer surfaces of the external walls, the normal vector is used. If the Z component of the normal vector is not equal to zero, it is left out of the selection. This means that horizontally or diagonally oriented surfaces are not taken into account. With only vertical surfaces left, a clear outer ring of surfaces can be detected. If the horizontal components of the walls were included, it would be hard to find the adjacent outer surfaces. The tool assumes the adjacency of outer surfaces, which could not have worked with the horizontal components.

7.7.2 Overlapping roof

In the tool, it is assumed that only overlapping gabled roofs can be taken into account. When merging the walls and the roofs, the points that lie within the bounding box of the roof and the walls are selected. This selection of points is then inserted into the roof surfaces creating a hole. This approach would only work if the roof is overlapping with the walls. This is the case for a significant amount of buildings, however, this would not suffice for all.

A small experiment was executed in which the simple BIM had a flat roof (see figure 7.3). When this BIM was used as an input for the tool, it showed errors during the developing of the 3D city model. The errors occur when the wall and roof surfaces are merged. For future use, multiple roof shapes should be accepted before it can be used properly.

7.7.3 Other building elements

Another experiment was conducted, adding windows and doors to the simple BIM. When the results were visualized as an OBJ file in MeshLab, the windows and doors were represented by holes within the model. This means that no triangular faces were created for the windows or doors. When the 3D city model is created, the windows cause invalid geometries (see figure 7.4).

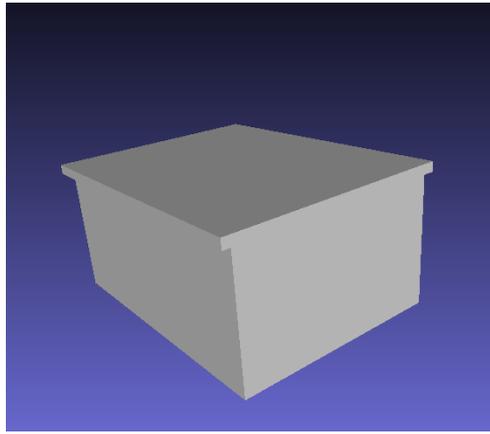


Figure 7.3: The simple BIM with a flat roof.

The tool must be further developed to handle other building elements. Besides, the doors and windows must be written correctly in the 3D city model, namely with a parent-children relationship to the city object **Building** in CityJSON. It is important that the coordinates are ordered in clockwise order, since this is the opposite orientation from the b-rep surfaces.

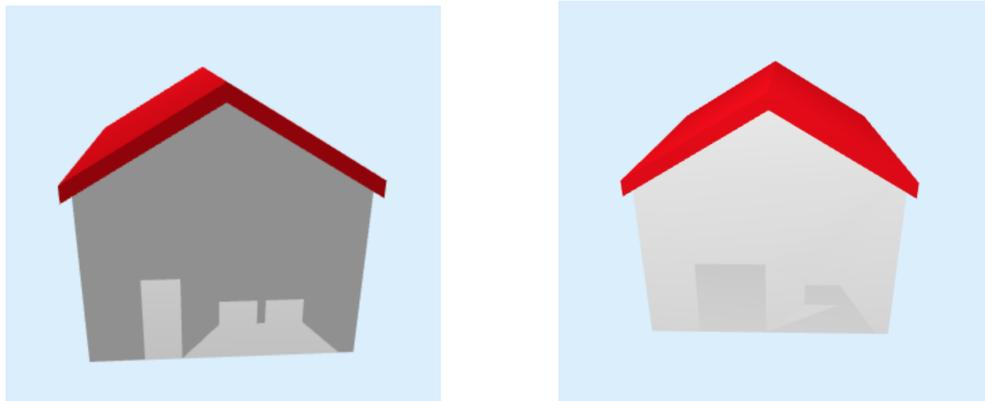


Figure 7.4: Other building elements added to the simple BIM.

If other more realistic BIMs were to be used as input for this tool, the tool itself should be extended. Currently, only walls and roof are accepted in this tool, however, other building elements can be required in order to check the rules from the land use plans. This tool shows that each entity from IFC must be handled carefully, otherwise the tool will not produce valid 3D city models.

7.7.4 More complex footprint

A third experiment was executed concerning the footprint of the BIM. In this research, a rectangular shaped footprint was taken as input. Therefore, a more complex footprint was created to test the tool. This resulted in errors, since the bounding box of outer walls is used to find adjacent outer surfaces. If a footprint of a building is any different than rectangular, inner surfaces can be selected. An example is shown in figure 7.5.

This will result in an invalid CityJSON file and cannot be used for further automatic rule checking. It remains an issue to select the outer surfaces of an entity.

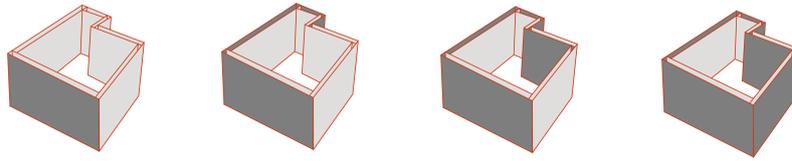


Figure 7.5: Selecting the outer surfaces when the footprint is not rectangular

In conclusion, another algorithm needs to be produced in order to select the outer surfaces of the BIM. Other BIMs with a non-rectangular shaped footprint cannot be handled with this tool at the moment.

In this chapter, the results of the experiments were discussed. Firstly, the issues encountered during this research were discussed. Afterwards, a reflection was made concerning the building permit process and the domain. Finally, the limitations of this research were stated. From the experiments and this discussion chapter, guidelines can be drafted.

8 | GUIDELINES

In the previous chapter, the results of the research were discussed and the limitations were laid out. The guidelines based on the results and the discussion are drafted. This chapter consists of guidelines that enhance the integration between the [AEC](#) domain and the geospatial domain and the use of [BIMs](#) in the digital building permit process. In the first section, the current guidelines are laid out. Afterwards, general guidelines regarding the integration are stated. The last section consists of guidelines specified for the digital building permit process.

8.1 CURRENT GUIDELINES

Several guidelines are analyzed for this research. In the guidelines proposed by [BIM Loket \[2022\]](#), the emphasis lies on the exchange of [BIMs](#) between different stakeholders in the [AEC](#) domain. The exchange of information will be more efficient and the information will be more reusable. Guidelines regarding the exchange, the geometry, and the semantics. [BIM Loket \[2022\]](#) states a [BIM](#) must be exchanged using [IFC](#) as a standard. A [BIM](#) may not contain any duplicates or intersecting geometries. The use of [IfcPropertySets](#) are recommended.

For [IFC](#) itself, many rules and recommendations are given in the [IFC](#) standard, implementation guide, implementer guide [IFC2x3](#) (and other versions), and the implementer agreements [[Ohori et al., 2018a](#)]. Besides the documentation of [IFC](#), the specifications of [CityJSON](#) [[Ledoux and Dukai, 2022](#)] are followed.

Other guidelines that are relevant for this research are stated by [Ohori et al. \[2018a\]](#). These guidelines focus on processing [IFC](#) files for later use in applications in the geospatial domain. The guidelines include: georeferencing, the use of valid volumetric objects, no intersecting geometries, forming enclosed spaces with [IfcSpace](#), and using specific entities.

8.2 GUIDELINES INTEGRATION

Currently, standards in the [AEC](#) and geospatial domain develop within their domain. However, this research shows that the integration of the domains can be useful. In this section, some guidelines are formulated for the standards to be more useful for not only the use case of this research, but other use cases as well.

8.2.1 Adding features in IFC

FOOTPRINT During this research, it was complex to determine the outer faces of the external walls. Only the outer surfaces were needed for creating the new 3D city model in which b-rep surfaces are solely used. It would be useful to distinguish which faces are external.

In Revit, it is already possible to set "Spot Coordinates". This can be done by select-

ing "Manage" > "Dimension Panel" > "Spot Coordinates". This can be done for each point in the 3D model in Revit, for example, for the footprint. The Spot Coordinates are not exported to the IFC file. Technically, it would be possible to represent these in IFC. The Spot Coordinates would be represented by **IfcCartesianPoints** as an attribute of **IfcPolyLine**. The **IfcPolyLine** would be an attribute of **IfcShapeRepresentation** which would be an attribute of **IfcProductDefinitionShape**. These are all entities representing the geometries of building element entities, for example, **IfcWall**. Since this polyline would not belong to a building element, it would need an entity itself. In this research, Spot Coordinates would be used to detect the footprint of the building and, therefore, **IfcFootprint** would be a correct naming of the new entity.

FRONTAL BUILDING LINE For some rules, information was missing in IFC. For example, the frontal building line could not be distinguished from other facades. This would be a practical feature which could be included in a **IfcPropertySingleValue** that looks like the current attribute *IsExternal*.

8.2.2 Georeferencing

Currently, it is possible to georeference the building. However, it has an offset of within two meters in Revit. In the case of the digital building permit process, this would not always lead to errors. In other use cases, a higher precision might be required, for example, 3D registration.

8.3 GUIDELINES DIGITAL BUILDING PERMIT PROCESS

In this section, guidelines are stated that would support the municipality in the digital building permit process. The guidelines can be seen as requests from the municipality to BIMs that will be checked on the rules or as guidelines for architects who are designing buildings.

8.3.1 IFC schema

The roof of a building can be represented by the entities **IfcRoof** and **IfcSlab**. **IfcRoof** is a description of the total roof and can consist of one or more **IfcSlabs** [buildingSmart, 2022]. In the case of the BIM, there were two **IfcSlabs** that formed the **IfcRoof**. However, this is not always the case. The received BIMs were analyzed on these entities. In some cases, the **IfcRoof** was not present. Therefore, it is a must that BIMs must follow the IFC Schemas in which these entities are explained. The IfcCheckingTool can be used to check the BIMs against such requirements defined in the schemas.

8.3.2 Georeferencing

Georeferencing the BIM is a must for urban rule checking. Without information about the latitude and longitude of the BIM, it would be impossible to determine the correct peil information and therefore, check the rules. On top of that, it is important to always state which vertex is the Survey Point for the digital building permit process. Doing so, it becomes clear which point in the BIM is georeferenced and thus how the 3D city model is located on the map.

8.3.3 Join building elements

To take a **BIM** as input for the conversion, the building elements must be watertight so that the 3D city model encoded in CityJSON is a valid geometric primitive. Creating a watertight **BIM** can be done in Revit in multiple ways. First of all, the walls can be joined to the roof by selecting the wall and selecting "Attach Top/Base". This is important when the walls and the roof surfaces are merged during the conversion. Besides, there is another way to assure that building elements touch. This can be done by selecting a building element and go to "Modify">"Geometry Panel">"Join">"Join Geometry".

In this chapter, several guidelines were described. First of all, current guidelines were stated for the use of **BIMs** for example. Afterwards, guidelines for further integration and the digital building permit process are formulated. From this process, conclusions are made in the next chapter.

The aim of this research is to perform automatic rule checking on a 3D city model to answer the research question: How could BIM support the digital building permit process in the Netherlands? The answer to the research question will be given and the sub-questions will be answered as well. Finally, future research is pointed out.

9.1 OVERVIEW OF RESEARCH

In this research, a selection of rules from land use plans were checked using received BIMs as input. Only the most frequently used rules were checked. The required information of these rules was analyzed in the standards IFC and CityJSON. Afterwards, the entities that represent the required information were checked on its presence in the input models. The information was then completed before the the conversion could be performed. The conversion integrates the BIM with data on the peil dataset and the 3D Basisvoorziening to check the rules. From this process guidelines are formulated.

9.2 CONCLUSION

In this section, the research (sub-)question(s) raised in chapter 1 will be answered. Before answering the main question of this research, the sub-questions are answered:

1. Which information from the BIM and the SOR is needed for the digital building permit process?

Six land use plans were analyzed for this sub-question. Even though land use plans describe rules extensively, only three rules occur in almost all land use plans. These rules are: the maximum building height, the maximum gutter height, and the maximum height of other constructions. These rules have in common that the height of the building, gutter, or other construction must be determined. This is done from the peil to the highest point of the building, the gutter height, or the highest point of the other construction.

For this research, it is checked whether the entrances of the locations of the received BIMs are bordering a road. If so, the roads must be taken into account for all the rules, which is the case for all locations. For the maximum building height, the walls and the roof are taken into account, since all building elements should be taken into account except for subordinate elements like chimneys, antennas, etc. [Gemeente Rotterdam, 2015] [Gemeente Den Haag, 2020][Gemeente Den Haag, 2014]. The roof is also required information to calculate the maximum gutter height. To measure the maximum height of other constructions, the whole object that represents a construction other than a building being must be taken into account.

All the required information related to single buildings can be represented by IFC as well as in CityJSON. CityJSON is able to represent roads and other constructions.

2. Which required information in the BIM and the SOR is present? Which required information is absent?

In this research, the required information was present in the BIM. IfcRoof and IfcWall were included in Simple.ifc. In the smaller experiments conducted in this research, additional building elements were added to the Simple.ifc. This leads to invalid geometry in the 3D city model and so the tool is not yet able to handle other building elements. In these experiments, only a couple of elements were added. However, most BIMs have even more building elements than shown in this research, which must be handled as well to avoid errors.

In both standards, IFC and CityJSON, many entities are supported. However, only the required information of the rules in the land use plan that are wished to be checked must be supported. This means that there are more entities present in both standards than required for most rules of the land use plans. As stated before, these entities must be handled and analyzed as well, since these can cause errors.

3. Which information from the BIM and the SOR should be converted regarding the digital building permit process?

The walls and roofs from the BIM can be converted from IFC to CityJSON. For the peil, other datasets than the SOR are used. One of the locations resides in Rotterdam. The municipality of Rotterdam has created a peil dataset that is used in this research. For all other cases, the 3D Basisvoorziening is used. However, it must be noted that roads are modeled with an LoDo. This means that the whole object has the same height. For most of the Netherlands, this will not differ greatly, but in hilly landscapes the height can quite differ and so results can be inaccurate.

Unfortunately, it is impossible to represent other constructions. Although there are entities in CityJSON that would make it possible, this information is not present in the SOR. Since the information is absent in both input models, the maximum height of other constructions cannot be checked.

4. How to perform a conversion from a BIM and the SOR supporting the digital building permit process?

The conversion consists of several parts which will all be discussed separately. First of all, the entities from the IFC are converted to CityJSON. This is done using the triangular faces. From these triangular faces, surfaces are created by finding coplanar points. Afterwards, only the outer surfaces are retrieved and rewritten to b-rep surfaces using the normal vector and the orientation of the triangular faces. The surfaces of the wall and the roof were then merged together to one single building containing b-rep surfaces. The coordinates of these surfaces were translated, scaled, and then rotated regarding the Surevy Point. The Surevy Point is the only point containing latitude, longitude, and elevation information. This information had to be transformed from WGS84 coordinates in degrees minutes second to RD-coordinates.

The peil was retrieved from the peil dataset of the municipality of Rotterdam and from the 3D Basisvoorziening. From both dataset the vertex closest to the Surevy Point of the BIM was used and its value set as the peil.

The building with b-rep surfaces, its coordinates and the peil value were written to a 3D city model in CityJSON. Finally, this 3D city model was checked on the rules determined in the first sub-question.

From this process, guidelines could be formulated. The guidelines not only concern the integration in general but also the digital building permit process specifically.

After answering the sub-questions, the answer of the main question remains:

How could the BIM support the digital building permit process in the Netherlands?

The aim of this research is to find a solution for the downsize of information, manual checking of building permits, and the duration of the building permit process. This is done by combining the required information from the 3D Basisvoorziening, the peil dataset of Rotterdam, and the simple BIM in a new 3D city model, which was automatically checked on selected rules of the land use plan. The rule checking was performed successfully, which shows that BIM could support the digital building permit process. However, it must be noted that only two rules were checked which is a small part of all the rules that are currently being checked by municipalities. Nevertheless, this research shows that BIMs can be used in automatic rule checking.

Automatic rule checking can overcome current issues. It can shorten the duration of the process, since rules are automatically checked instead of manually. Besides, it is less prone to errors since manually checking is avoided. However, there are downsides regarding automatic rule checking. It could potentially take over jobs of employees, which could lead to unemployment. Another downside is the interpretation of the rules, which can be corrected by humans in the current building permit process, but not by a computer. This means that rules should be translated in a way that these are unambiguous and computer-readable.

Writing unambiguous rules for the land use plan is an issue already occurring in the building permit process. Even though it would be technically possible to perform automatic rule checking, practice shows that the current building permit process is not about to change that direction. Not only would the rules have to be rewritten, but also builders are not always working with BIMs. Most smaller projects are done by means of 2D drawings based on a BIM or not. Since these projects are not based on BIMs, automatic rule checking cannot be performed. Before automatic rule checking can be successful, it is important that all stakeholders are using BIMs. Even if all builders were accustomed to BIMs nowadays, the Environment and Planning Act will be a large operation for all municipalities in the Netherlands. As of today, the start date of the law is set on the 1st of January 2023. Municipalities will be focused on this transition rather than exploring options of automatic rule checking.

In conclusion, this research shows the possibilities of BIM supporting the digital building permit process. It could be a first step to even further digitalize the process and to overcome current issues in the building permit process.

9.3 FUTURE WORK

During this research, several unclear formulations were encountered in the land use plans. For example, it was unclear when a building was bordering a road directly or how the average ground level must be calculated. To use automatic rule checking in the future, it is essential that all rules are formulated clear and unambiguous. As a result, the land use plans should be revised on these ambiguous statements and corrected wherever needed to more precise statements.

Georeferencing as it is done in this research in Revit has an offset. The coordinates of the Survey Point are used to retrieve the correct peil value. If the offset is off, it could cause the wrong peil value to be selecting. Some municipalities in the

Netherlands already work with datasets with more accurate information than the 3D Basisvoorziening contains. These datasets are currently not freely available.

Revit did introduce new features in its newest version in 2022. In this version, the georeferencing was not updated. To improve the integration between the [AEC](#) domain and the geospatial domain, an increased precision and accuracy of georeferencing is crucial.

Multiple guidelines have been formulated in this research regarding the [BIM](#) and the digital building permit process. The guidelines enable further integration between the [AEC](#) domain and geospatial domain, if they are followed. The digital building permit process can experience further digitalization, which means less downsize of information, less manual checking, and hopefully a reduced duration of the process. However, the tool must be able to handle more complex geometries, an higher amount of entities, and more rules from the land use plans before it can really support municipalities in the digital building permit process.

BIBLIOGRAPHY

- 3D Geoinformation (TU Delft) (2020). CityJSON Documentation.
- Autodesk (2022). About revit. [Accessed 20-04-2022].
- Biljecki, F., Ledoux, H., and Stoter, J. (2016). An improved LOD specification for 3D building models. In *Computers, Environment and Urban Systems*, pages 25–37.
- BIM Loket (2022). BIM BASIS ILS. Eenvoudig samenwerken. [Accessed on 11-2-2022].
- Bossche, J. (2017). How are Revit coordinates exported to IFC? [Accessed on 29-04-2022].
- buildingSmart (2022). IFC4x3. [Accessed 20-01-2022].
- CHEK (2021). Change toolkit for digital building permit. [Accessed 09-03-2022].
- de Laat, R. and van Berlo, L. (2011). Integration of BIM and GIS: The Development of the CityGML GeoBIM Extension. [Accessed 22-01-2022].
- Diakité, A. (2018). About the Geo-referencing of BIM models. [Accessed 15-02-2022].
- Dijkmans, T. and van Berlo, L. (2013). 3D bestemmingsplannen & BIM: Showcase van beschikbare 3D technologie ten behoeve van digitaal toetsen. [Accessed 21-01-2022].
- Floros, G., Ellul, C., and Dimopoulou, E. (2018). Investigating Interoperability Capabilities Between IFC and CITYGML LOD 4 – Retaining Semantic Information. *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-4/W10:33–40.
- Gemeente Den Haag (2014). Kijkduin-Ockenburgh. [Accessed 20-01-2022].
- Gemeente Den Haag (2016). Vogelwijk. [Accessed 20-01-2022].
- Gemeente Den Haag (2020). Vruchten- en Heesterbuurt. [Accessed 20-01-2022].
- Gemeente Den Haag, Gemeente Rotterdam, Gemeente Amsterdam, VNG (2020). Programma Totaal Driedimensionaal (T3D).
- Gemeente Nijmegen (2015). Nijmegen Oud West. [Accessed 20-01-2022].
- Gemeente Rotterdam (2015). Waterstad. [Accessed 20-01-2022].
- Geo-Informatie Nederland (2021). Programma Totaal Driedimensionaal. [Accessed on 04-05-2022].
- Idrizi, B. (2021). Redefining of “epsg” and “proj” for current state coordinate reference system of the republic of north macedonia. *Proceedings of the ICA*, 4:1–8.
- Jaud, Š., Donaubaauer, A., Dr, O., and Borrmann, A. (2020). Georeferencing in the context of building information modelling. *Automation in Construction*, 118:103211.
- Kadaster (2022a). Fout in registratie melden. [Accessed 28-02-2022].
- Kadaster (2022b). Over BAG. [Accessed 09-02-2022].
- Kadaster (2022c). TOPNL. [Accessed 09-02-2022].

- Kadaster (2022d). WOZ-waarde van uw huis. [Accessed 09-02-2022].
- Karlsruhe Institute of Technology (2022). IfcCheckingTool. [Accessed 14-02-2022].
- Laakso, M. and Kiviniemi, A. (2012). The IFC Standard - A Review of History, Development, and Standardization. *Electronic Journal of Information Technology in Construction*, 17.
- Ledoux, H. (2018). val3dity: validation of 3D GIS primitives [According to the international standards. *Open Geospatial Data, Software and Standards*, 3.
- Ledoux, H. and Dukai, B. (2022). CityJSON Specifications 1.1.0. [Accessed 20-01-2022].
- Ledoux, H. and Feenstra, J. (2022). cjval: a validator for CityJSON files. [Accessed on 29-04-2022].
- Ledoux, H., Ohori, K., Kumar, K., Dukai, B., Labetski, A., and Vitalis, S. (2019). Cityjson: a compact and easy-to-use encoding of the citygml data model. *Open Geospatial Data Software and Standards*, 4:4.
- Leijten, S., Rietdijk, M., and Krijtenburg, D. (2021). DiS Geo : Eisen aan model samenhangende objectenregistratie. [Accessed 04-09-2021].
- Noardo, F., Ellul, C., Harrie, L., Overland, I., Shariat, M., Ohori, K. A., and Stoter, J. (2020a). Opportunities and challenges for geobim in europe: developing a building permits use-case to raise awareness and examine technical interoperability challenges. *Spatial Science*, 65:209–233.
- Noardo, F., Wu, T., Ohori, K., Krijnen, T., and Stoter, J. (2020b). Investigating the automation of building permit checks through 3D GeoBIM information. [Accessed 22-01-2022].
- Odiijk, M. (2019). Samenhangende objectenregistratie.
- Ohori, K., Biljecki, F., Diakité, A., Krijnen, T., Ledoux, H., and Stoter, J. (2017). Towards an integration of gis and bim data: what are the geometric and topological issues? *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-4/W5:1–8.
- Ohori, K. A., Diakité, A., Krijnen, T., Ledoux, H., and Stoter, J. (2018a). Processing BIM and GIS Models in Practice: Experiences and Recommendations from a GeoBIM Project in The Netherlands. *International Journal of Geo-Information*, 7:311.
- Ohori, K. A., Krijnen, T., Diakit, A., Ledoux, H., and Stoter, J. (2018b). GeoBIMproject Final report.
- Olsson, P.-O., Axelsson, J., Hooper, M., and Harrie, L. (2018). Automation of Building Permission by Integration of BIM and Geospatial Data. *ISPRS International Journal of Geo-Information*, 7:307.
- PDOK (2022). 3D Basisvoorziening. [Accessed 09-03-2022].
- Pedro, J., Meijer, F., and Visscher, H. (2011). Comparison of building permit procedures in european in european union countries.
- QGIS3.22 (2022). Documentation for QGIS 3.22. [Accessed on 29-04-2022].
- Rijkswaterstaat (2022). Kenniscentrum InfoMil. [Accessed 08-02-2022].
- SAB (2022). Bestemmingsplan en omgevingsvergunning. [Accessed 08-02-2022].

- Wu, J. (2021). Automatic building permits checks by means of 3d city models. [Accessed 09-03-2022].
- Zhu, J., Wright, G., Wang, J., and Wang, X. (2018). A critical review of the integration of geographic information system and building information modelling at the data level. *ISPRS International Journal of Geo-Information*, 7:66.

A | APPENDIX

A.1 FIGURES AND TABLES

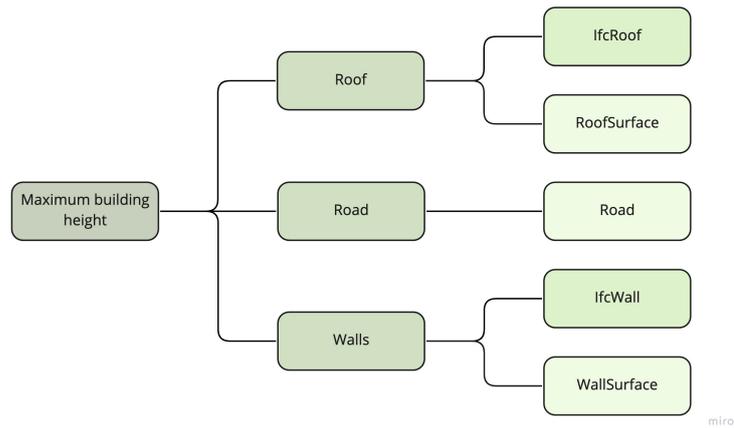


Figure A.1: From maximum building height to architectural elements to entities in IFC and CityJSON

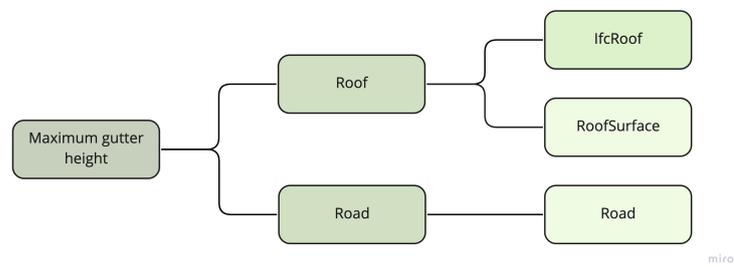


Figure A.2: From maximum gutter height to architectural elements to entities in IFC and CityJSON

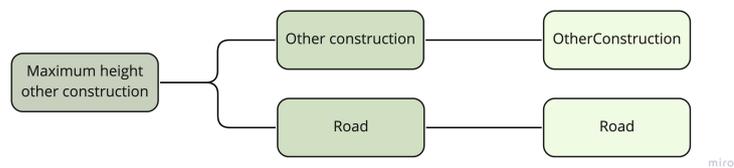


Figure A.3: From maximum height other constructions to architectural elements to entities in IFC and CityJSON

	(a)	(b)	(c)	(d)	(e)	(f)
Maximum building height	13	11	7	13	7	200
Maximum gutter height	9	7	-	8	-	-
Maximum height other constructions	4	3	5	1-2	5	-

Table A.1: Measurements for each rule: (a) Schependomlaan, (b) Haviklaan, (c) Rabarberstraat, (d) Witte de Withstraat, (e) Wijndaelersingel, (f) Jufferstraat

	(a)	(b)	(c)
CONSECUTIVE_POINTS_SAME	2	2	1
SHELL_NOT_CLOSED	15	4	6
NON_MANIFOLD_CASE	15	1	6
MULTIPLE_CONNECTED_COMPONENTS	12	9	7
POLYGON_WRONG_ORIENTATION	8	0	1

Table A.2: Errors 3D BGT detected by val3dity: (a) Schependomlaan, (b) Jufferstraat, (c) Den Haag

A.2 SUMMARY OF TRANSCRIPTIONS

A.2.1 Municipality of Rotterdam (Rolf Jonker)

The municipality of Rotterdam has created a tool to perform automatic rule checking with SPARKLE SGL. The rules in this tool can be divided into five categories: security, structure, aesthetics, rules of the land use plans, and noise. The interface of this tool is used and it can successfully check the rules of the previously mentioned categories. In this tool not all the regulations are checked. On top of that, only two areas in the municipality can be checked. The tool takes about 2 minutes to check the rules, however, that does not mean it will be so much faster as thought before. Some rules were not checked before, but are in the tool.

The prediction is that eventually all rules can be checked automatically. One of the things preventing this are the lack of proper datasets of other municipalities. Also the Environment and Planning Act will be established on the 1st of January 2023 as of now. Once this law is established, municipalities will focus on the huge changes this law will bring and a tool like automatic rule checking will not be further explored. It could be that in the DSO a button will be included in which the user can decide whether a traditional process of the building permit process will be followed or the new one which allows automatic rule checking.

A.2.2 Municipality of Delft (Marianne Schoenmakers en Henry Rienstra)

One of the main issues in the current building permit process are the rules themselves. These are written by a certain department of the municipality, but often a different meaning is behind the rules. The rules are read in a different way than the thought behind it originally was. On top of that, the municipality must check the rules in the land use plan before it is established. This does not happen at all times, which causes errors that need to be corrected later in the process. Correcting errors is a time and labour-intensive effort. To perform automatic rule checking, the rules should be stated unambiguously and measurable.

Another current issue is that not all projects are based on BIMs. The larger projects are and they deliver the municipality with 3D models of the structure and pipes within the building. However, smaller projects are based on 2D drawings and

knowledge in the field. Full automatic rule checking is only possible when these projects are modeled in [BIM](#) as well.

The Environment and Planning Act will be working from the 1st of January 2023. The municipalities are already working environmental plans (in Dutch: omgevingsplannen). An environmental plan will consist of all the rules of land use plans and other regulations (in Dutch: verordeningen). On top of that, they will be valid for the whole municipality and not a part of the municipality. Producing the environmental plan is more work than a land use plan and updating it will also cost more time. It is clearly stated that the Environment and Planning Act is the most important development in the building permit process, which means other developments will have to come afterwards. The opinions are divided whether automatic rule checking can support the digital building permit process, but both state that the rules must be written without any room for interpretation.

