

GEOBIM INFORMATION TO CHECK DIGITAL BUILDING PERMIT REGULATIONS

N. Hobeika^{1*}, J. van Liempt¹, F. Noardo¹, K. Arroyo Ohori¹, J. Stoter¹

¹ 3D Geoinformation group, Delft University of Technology, Delft, The Netherlands —
(n.hobeika, f.noardo, k.ohori, j.e.stoter)@tudelft.nl

KEY WORDS: Digital building permit, GeoBIM, Industry Foundation Classes, data integration, web-viewer

ABSTRACT:

Municipalities invest a lot of time and person-hours into manual building permit checks. With the increase in computational power and the use of Building Information Models (BIM) in the building design life cycle, several municipalities are investing in automating these checks using BIM and geo-data sets. However, few examples exist of tools effectively using geo-information with BIM. In order to address this gap, a project was developed with the municipality of Rotterdam (NL). In a previous phase, a tool was implemented, able to analyse the BIM data to extract the needed information for a few representative regulations. In order to extend and improve the previously developed tool, a web-based interface is now implemented and geo-data sets are integrated to the process allowing more powerful GeoBIM analysis. Three checks using both BIM and GIS data were implemented and tested: (i) The parcel limit check evaluates if the building's footprint derived from BIM falls within the parcel limit provided from the municipalities' parcel data sets. (ii) The height check evaluates the maximum building's relative height to the road's height. Finally, (iii) the road overhang check detects neighbouring roads to the parcel and evaluates the admissible overhang over that road. This paper presents these developments, including the type of input data that is needed for the checks, the tools for the three new GeoBIM checks (parcel limit check, height check, road overhang check) and the implementation in the web-based tool.

1. INTRODUCTION

In an effort to automate building permit checks, many studies and demonstrators are being proposed to digitalise the process, the involved data, and the used tools (Noardo et al., 2022a). For many of the regulations to be checked, both the information about the city (to be retrieved in the geo-data, or 3D city models) and the information about the designed buildings (in the Building Information Model - BIM) are relevant. For example, many urban parameters depend on the zoning, the distance from other buildings, public or private facilities, public transport and so on. For this reason, the automation of the checking of those kinds of regulations can only be effective if both geo-data and building data are considered in the processing. However, although many examples exist to automate the checking of regulations based on BIM (Kim et al., 2020; Olsson et al., 2018; Choi and Kim, 2017; Kasim et al., 2018; Krijnen and Van Berlo, 2016; Zhang and El-Gohary, 2015; Kincelova et al., 2020; Ciribini et al., 2016) few studies are available, which can effectively use both kinds of data integrated in a unique environment to support automatic planning checks (Van Berlo et al., 2013). Integrating BIM and Geographic Information System (GIS) data sets represents one further challenge, having different semantics, levels of detail, and geometric representations (Arroyo Ohori et al., 2018).

The issue was addressed within a project developed with the municipality of Rotterdam (NL), Noardo et al. (2020, 2022b), including the preliminary challenges of formalising the regulations and extracting rules, analysing the available data sets as produced in practice and developing a tool to support municipality officers when performing building permit checks. In the project, we investigated the process of digitalisation of building permits in a practice-related perspective and we solved several issues related to the use of the BIM, a very detailed model, to

extract the needed, more abstract, information for the regulations checks. The used BIM were stored in the Industry Foundation Classes (IFC) format, which is the buildingSMART reference Open standard for BIM.

The previous phase of this project mainly focused on analysing the BIM data rather than integrating BIM and GIS data sets. This means that some checks required users to manually add information that could be automatically extracted from geo-data sets. Additionally, the result was a Python tool which needed to be installed by the user, and there was no focus on user-friendliness, for example in the presentation of the analysis results.

To address these shortcomings, a follow-up project was developed aimed at implementing a web-based tool in which IFC models can be visualised and analysed for a clearer presentation of results, which is an important requirement for the municipality officers. In addition, the tool has now been extended by implementing functionalities to conduct three new regulation checks based on analysis of the integrated geo-information with BIM (GeoBIM). This paper describes the developed new solutions for GeoBIM-based automated regulation checking and presents the data requirements for the input data to facilitate this.

2. METHODOLOGY

Following a bottom-up approach, the follow-up project integrated geo-data sets with BIM data based on a case study building in Rotterdam. After discussing with the municipality officers, it was decided to implement three checks:

- The parcel check: the building's ground floor footprint should lie entirely within the parcel of the project;

* Corresponding author

- The height check: according to the Dutch building regulations, the admissible height is measured from the highest road level to the building's highest level;
- The overhang check: the Dutch building regulations specify a certain admissible overhang distance with respect to each road. Thus, every side of the building should be checked based on its adjacent road and its respective admissible overhang.

First, openly available data sets were tested to see what is needed to conduct the checks. Testing with this data led to important insight on which information should the geo-data input contain. Once the input data had all the needed information, the GeoBIM checks were implemented. In parallel, the web-based tool was developed in a Citrix environment (a user requirement) and regular meetings with the municipality officers were held to get their feedback while using the tool.

3. INPUT DATA NEEDED AND THEIR REQUIREMENTS

To automate certain checks, integrating GIS data sets with the BIM is required. This section presents the data sets used in the tool and the information that needs to be included to successfully run the checks.

3.1 Parcel data set

The parcel data set contains the boundary limits of a piece of owned land. The data set should represent parcels as polygons. Fully and partially overlapping polygons should be avoided to prevent ambiguities when automatically choosing the correct parcel of the project.

3.2 Roads data set

The roads data set should represent roads as polygons with their name as an attribute. The name of the roads is important because Dutch regulations are written according to specific roads.

3.3 Height points data set

The height points data set contains geo-referenced 2D points with their height as attribute. These height points should preferably coincide with the corner points of the parcel.

3.4 IFC file

In the IFC file, storeys should be consistently grouped and represented and numerically indexed when they are named, as follows: '00' = GF; '01' = 1st floor; '-01' = 1st underground floor; etc.

3.5 JSON-formatted regulations file

Representing regulations parameters and requirements in a machine-readable format is a major challenge in the digitalisation of building permitting.

It would enable automation in the performance of the corresponding analysis, to evaluate whether a check passes or fails.

A custom JSON-formatted file is proposed specifically for this project. An example is shown in the following lines. Other alternatives exist, such as using the Gherkin language (Moult and Krijnen, 2020).

```
{
  "overhang": {
    "rule1": {
      "streetname": "Hertekade",
      "floor": 1,
      "maxMetres": 5
    },
    "rule2": {
      "streetname": "Boompjes",
      "floor": 12,
      "maxMetres": 3
    }
  },
  "overlap": {
    "rule1": {
      "floor": 1,
      "maxPercentage": 89
    },
    "rule2": {
      "floor": 12,
      "maxPercentage": 92
    }
  },
  "height": {
    "maxMetres": 103.47
  }
}
```

4. THE GEOBIM CHECKS FOR DIGITAL BUILDING PERMITS

The regulations considered to perform GeoBIM integrated analysis using the original geo-data sources are: the overhang check with automatic detection of corresponding roads; the height check with respect to the highest road level, and the parcel boundary check.

To automate certain checks, integrating GIS data sets with the BIM project is required. The data sets needed for the tool are the parcels (geo), the roads (geo), the reference height points (geo), the IFC file (BIM), and the JSON-formatted regulations file.

4.1 Parcel detection

Since all three checks require detecting the parcel in which the project is located, this section presents the steps taken to extract the cadastral parcel. First, the centroid of the ground floor (GF) footprint is computed from the IFC file and used to find the parcel it lies within in the parcel data set.

It should be noted that buildings and parcels with an L-shaped footprint might result in an incorrect parcel detection and a more robust detection method is needed. However, in Rotterdam, parcels are mostly rectangular and thus the assumption holds.

4.2 Parcel limit check

To perform the parcel limit check, after the parcel of the project is detected as in Section 4.1, the GF footprint is checked against the parcel polygon. If the footprint completely fits the parcel, the result is a "Pass"; otherwise, it is a "Fail" (Figure 1).

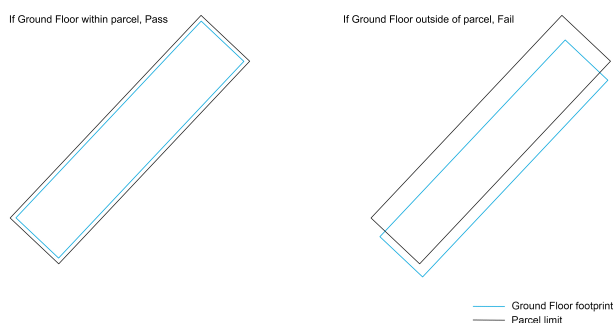


Figure 1. Steps to automatically check whether the building's GF is within the parcel boundary

4.3 Height check

To perform the height check, the parcel, the height points of roads and the maximum allowed height are needed. First, the overall height of the building is computed from the IFC file. Then, the parcel of the project is detected (see Section 4.1) and is used to select the corresponding height points from the height points data set. Afterwards, the highest road level is automatically determined from the selected height points. The overall height is adjusted according to the highest road level (Figure 2).

This means that if the highest road level is lower than the entrance level of the building, then the difference between the entrance level and the highest road level is added to the overall height. This includes the height difference between the building and the road to the total height of the building. Otherwise, the difference between the entrance level and the highest road level is subtracted from the overall height, thus, excluding the the height difference between the building and the road from the overall height. The final adjusted height is then compared to the maximum allowed height (Figure 3) and the result is given as a "Pass" or "Fail". The maximum allowed height is visualised as a plane in the tool with a green colour if it passed the check or a red colour if it failed the check.

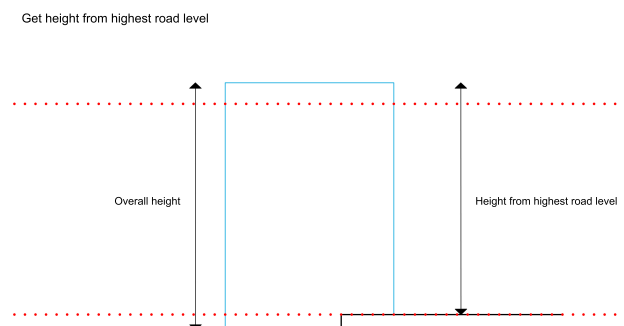


Figure 2. Getting the height of the building from the highest road level

4.4 Road overhang check

To perform this check, the parcel of the project, the roads adjacent to the parcel and the regulations related to those roads are needed. First, the parcel that contains the project is automatically determined (see Section 4.1 and Figure 4a). The parcel's boundary is buffered and used to automatically detect the adjacent roads from a roads' data set (Figure 4b). Once adjacent roads are detected, each floor of the building, starting from the first floor, is simplified into its Oriented Bounding Box (OBB)

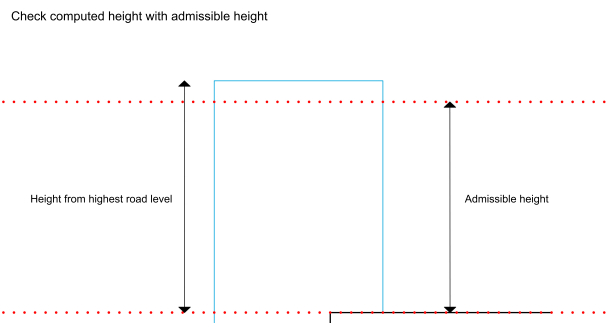


Figure 3. Compare computed height to admissible height

(Figure 3c). The normal of each vertical side of the OBB is computed and is checked against the adjacent roads (Figures 3d–1g).

If the normal intersects a road, the name of the road is checked for the admissible overhang and the distance from the furthest 2D point of the tested vertical side to the GF is calculated (Figures 3d and 2e). Those two values are then compared to see if the side of the floor passes the overhang check or not. A side is marked in green if it passes the check and in red if it fails the check (Figure 1h).

If the normal does not intersect any road, it is marked in black as a side not facing a road (Figures 2f and 1g).

5. THE WEB-BASED TOOL

The implementation of these checks in a web-based tool¹ increases the user-friendliness, as users do not have to install any software and the heavy processes required for the automated checks are executed on the server rather than on the user's computer. The application is a fork of Krijnen (2021), which is a full-stack application allowing users to upload IFC files and view them in the browser. IFC files get converted to glTF on a Flask server using IfcOpenShell, and are then visualised using BIMSurfer2.

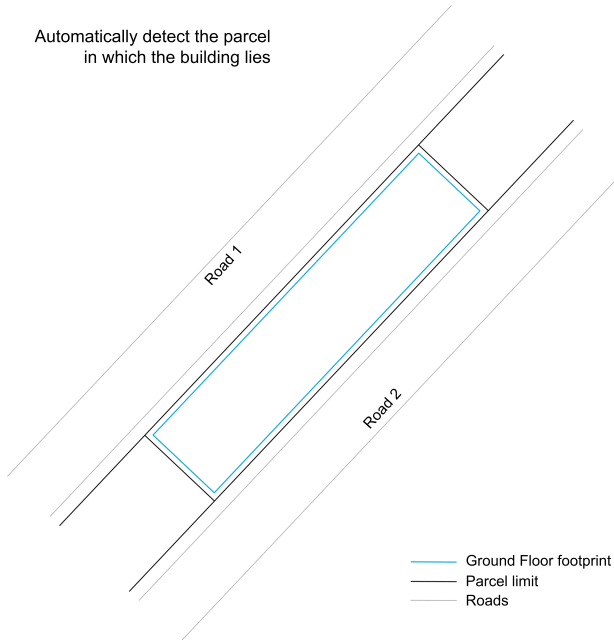
An overview model of the architecture of the application is shown in Figure 2. As for the data part, users can either upload a geo-referenced IFC file, or choose a pre-loaded one. Pre-loaded files are already converted to glTF, which enables the model to be visualised quicker. Additionally, the geo-information data sets are stored as Shapefiles on the server. These are all used for analysis, and the roads are used for visualisation as well. For now, the roads data set is in 2D, which means that the roads do not follow the terrain's topography.

The analysis is performed on a Flask server, mainly using IfcOpenShell. The client can make a request for an analysis to be performed, and the server will return a JSON-formatted response containing the outcome. The server handles the uploading, storage, and conversion of IFC models to glTF as well. The front-end is based on Vue.js and Three.js. For the visualisation of IFC, BIMsurfer2 is used, and for the visualisation of geo-information together with the model, a parser for GeoJSON to Three.js meshes is written.

While it is not difficult to use Shapefiles for analysis with Python, for visualisation with Three.js, GeoJSON is much

¹ <http://godzilla.bk.tudelft.nl/geobim-tool/viewer>

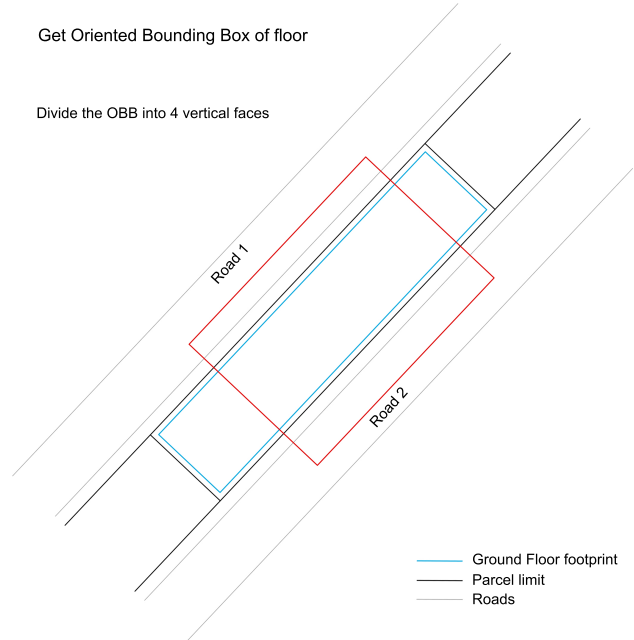
Automatically detect the parcel
in which the building lies



(a) Detecting the parcel of the building

Get Oriented Bounding Box of floor

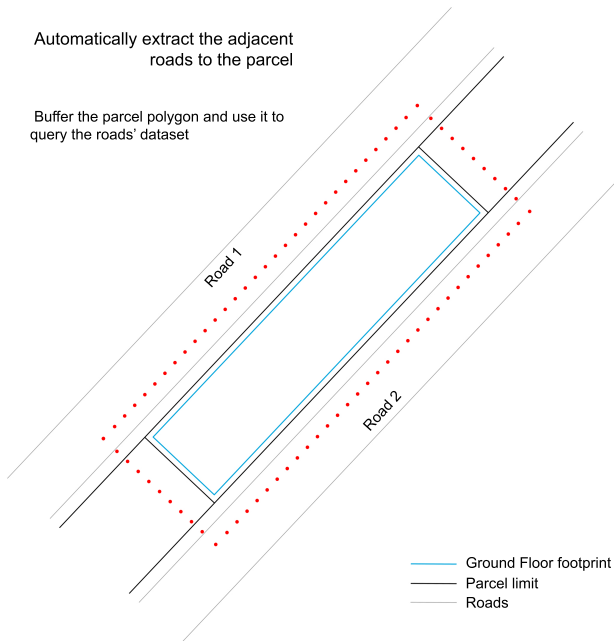
Divide the OBB into 4 vertical faces



(c) Simplifying a floor into its OBB

Automatically extract the adjacent
roads to the parcel

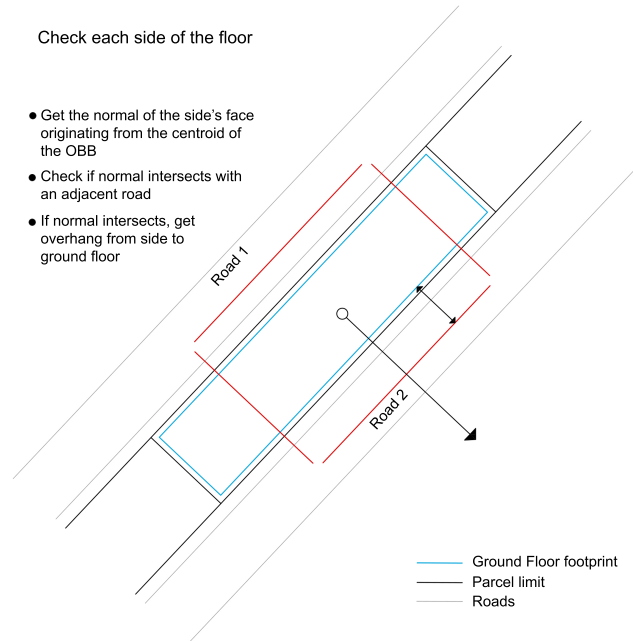
Buffer the parcel polygon and use it to
query the roads' dataset



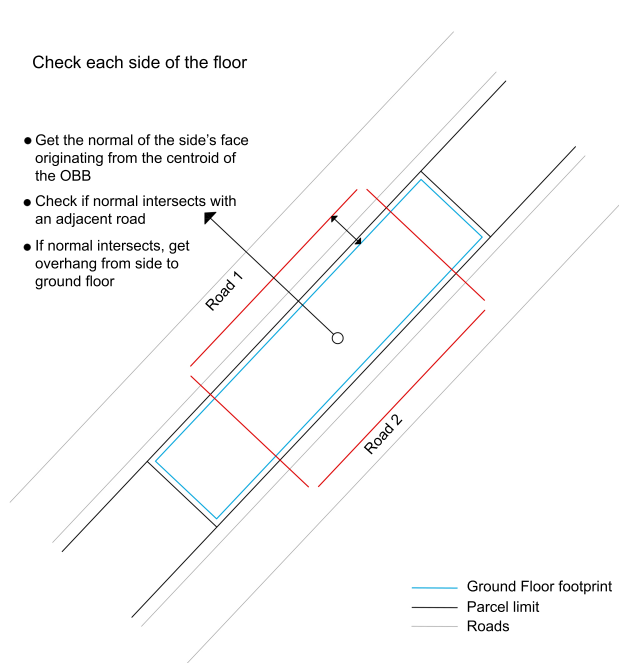
(b) Detecting the adjacent roads to the parcel

Check each side of the floor

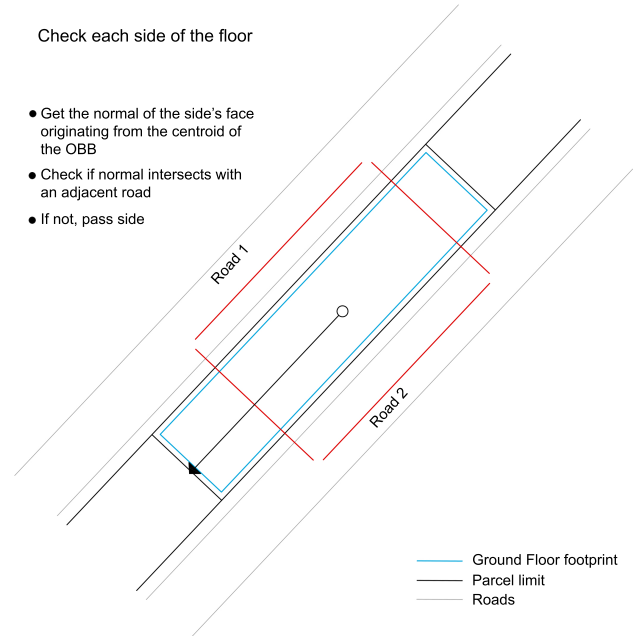
- Get the normal of the side's face originating from the centroid of the OBB
- Check if normal intersects with an adjacent road
- If normal intersects, get overhang from side to ground floor



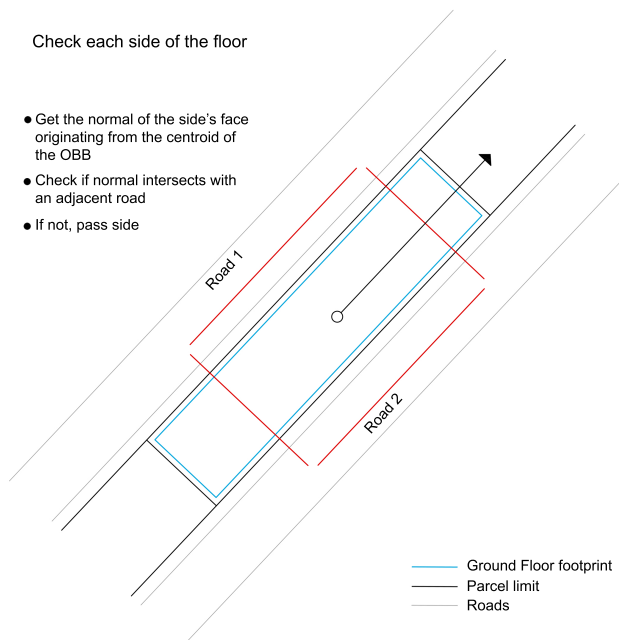
(d) Checking if a side overhangs over a road - case that does overhang



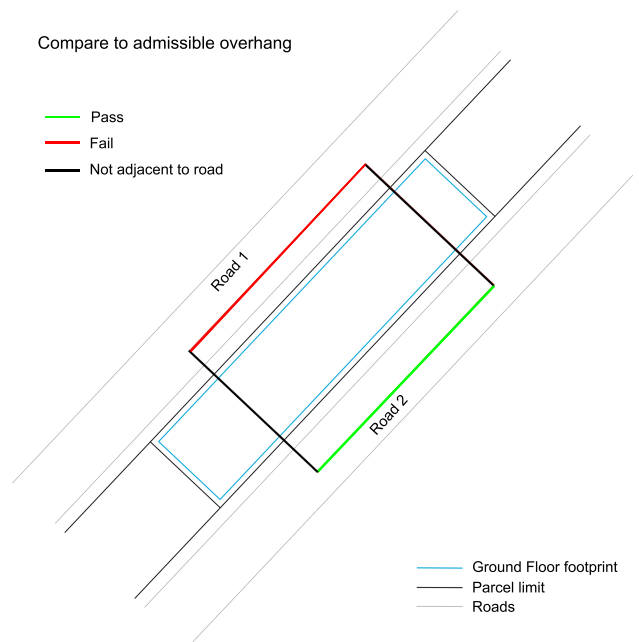
(e) Checking if a side overhangs over a road - case that does overhang



(g) Checking if a side overhangs over a road - case that does not overhang



(f) Checking if a side overhangs over a road - case that does not overhang



(h) Checking if sides pass or fail the regulation's guideline

Figure 1. Steps to automatically check the overhang over roads

easier. Only the road data is visualised, and it gets converted to GeoJSON in the client. Lastly, the application can be deployed using Docker Compose.

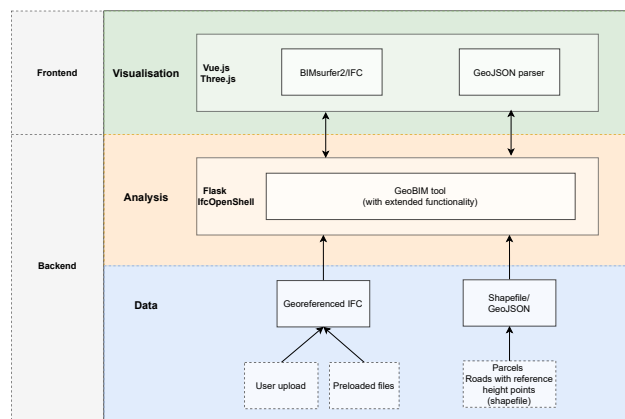


Figure 2. The architecture of the web-based tool

6. DISCUSSION - PERFORMANCE AND POSSIBLE IMPROVEMENTS

The implementation of a tool able to use both geo- and BIM information to check planning regulations is a step forward in the automation and digitalisation of building permitting, enabling the check of a number of otherwise excluded regulations, needing both kinds of information. This improves both the efficiency and the transparency of building permit checks. With the proposed tool, the manual intervention of the user is reduced, consequently, lowering the potential errors and increasing the efficiency of municipality officers. In addition, the introduction of a more user-friendly web-based interface to access and use the proposed tool allows a wider range of operators to take advantage of and to support the municipal officer to perform the checks.

However, since it is necessary to run the tool on a Citrix environment, initially the performance of the tool was problematic due to the visualisation of the model being heavy on the GPU. Therefore, a quick performance improvement in BIMsurfer2 was implemented by merging all individual Three.js meshes together. This greatly helped the visualisation performance, but, as a consequence, it is not possible to select individual IFC elements anymore. It can be made possible by parsing the data to Three.js meshes in a similar way as the Vitalis (2021) viewer for CityJSON² does.

It can also be interesting to try out a new IFC loader for Three.js (https://threejs.org/examples/webgl_loader_ifc.html) that was not mature yet when this phase of the project started. The new IFC loader is not only used for performance reasons, but also to make the application simpler by removing BIMsurfer2 as dependency and the need for IFC to glTF conversion.

It should also be noted that dealing with geo-data sets almost always comes with edge cases that the tool does not take into account. In the three checks that were implemented, some faulty cases might arise due to the data sets. For example, from the

parcel data set, the wrong parcel can be detected in case of overlapping polygons. This could particularly happen in the case of a larger zonal polygon that encompasses a whole region in the data set. Therefore, the cleaner the input data sets, the more reliable the checks are. For better automation of checks, regulations can also be generalised. For instance, for the overhang check, regulations can refer to road codes instead of road names.

Lastly, currently some of the analysis functions can take up to around 10 minutes. Besides improving their efficiency, it would be beneficial to make them run properly in the background, using for example Celery Celery (2021).

7. CONCLUSIONS

The combined use of geo-information with BIM is essential to automate several planning checks and achieve the currently more and more discussed topic of digitalisation of building permitting.

This paper describes the development of a web application to perform three automated building regulation checks using integrated geo- and BIM information, within the context of a project developed for the city of Rotterdam. The related input data requirements to run the web application were presented. Among these, specific attention was paid to the encoding of regulations parameters in JSON, in a way that makes them machine-readable and facilitates automation.

The challenges given by the heterogeneity of the different kinds of data were overcome by using the information contained in them. This work represents a relevant step towards the actual use of the integrated geo-information with BIM for regulations compliance checking, which is a critical progress towards building permit digitalisation. The increased usability of the tools will help the uptake of this technology in practice, supporting also new related research. It is possible to download and reuse the source code of our application by following the instructions provided at <https://github.com/tudelft3d/GeoBIM-building-permit-tool>.

For future work, the new functionalities developed for this project should be tested with other use cases (e.g. other buildings and other regulations) to check whether there is a need for more general assumptions or whether edge cases might arise. The tool can also be further developed by adding new features and automating more analyses. Some of the additional work that can be done can be divided into front-end side and back-end side improvements. The front-end improvements include using a better-performing viewer and analysis for faster and better results visualisation and visualising the roads data set in 3D in the viewer to place the building correctly relative to those roads. The back-end improvements include implementing the required number of parking places analysis, automatic reading of building permit documents (“toepasbare regels Omgevingswet”), conducting the checks using the 3D alpha shape of the building or the storeys, using an API (BGT API or an internal municipality API) to retrieve geo-information (rather than shapefiles), and ideally, extending the tool for more building permit check related applications, for example flood analysis, noise calculations, etc.

² <https://ninja.cityjson.org>

References

- Arroyo Otori, K., Diakité, A., Krijnen, T., Ledoux, H. and Stoter, J., 2018. Processing BIM and GIS models in practice: Experiences and recommendations from a GeoBIM project in the Netherlands. *ISPRS International Journal of Geo-Information*.
- Celery, 2021. Celery. <https://github.com/celery/celery>.
- Choi, J. and Kim, I., 2017. A methodology of building code checking system for building permission based on openBIM. In: *ISARC 2017 - Proceedings of the 34th International Symposium on Automation and Robotics in Construction*, pp. 945–950.
- Ciribini, A., Ventura, S. M. and Paneroni, M., 2016. Implementation of an interoperable process to optimise design and construction phases of a residential building: A bim pilot project. *Automation in construction* 71, pp. 62–73.
- Kasim, T., Li, H., Rezgui, Y. and Beach, T., 2018. Integrated Framework to Manage Building's Sustainability Efficiency, Design Features and Building Envelope. In: Y. CamarinhaMatos, LM and Afsarmanesh, H and Rezgui (ed.), *IFIP Advances in Information and Communication Technology*, IFIP Advances in Information and Communication Technology, Vol. 534, IFIP WG 5.5 COVE; Soc Collaborat Networks; Cardiff Univ; Nova Univ Lisbon; UNINOVA; Univ Amsterdam, pp. 650–660.
- Kim, I., Choi, J., Teo, E. A. L. and Sun, H., 2020. Development of kbim e-submission prototypical system for the openbim-based building permit framework. *Journal of Civil Engineering and Management* 26(8), pp. 744–756.
- Kincelova, K., Boton, C., Blanchet, P. and Dagenais, C., 2020. Fire safety in tall timber building: A BIM-based automated code-checking approach. *Buildings*.
- Krijnen, T., 2021. ifc-pipeline. <https://github.com/AECgeeks/ifc-pipeline>.
- Krijnen, T. and Van Berlo, L., 2016. Methodologies for requirement checking on building models: A technology overview. In: *Design and Decision Support Systems in Architecture and Urban Planning - 13th International Conference on Design and Decision Support Systems in Architecture and Urban Planning, DDSS 2016*.
- Moult, D. and Krijnen, T., 2020. Compliance checking on building models with the gherkin language and continuous integration. In: *Proceedings of the EG-ICE 2020 Workshop on Intelligent Computing in Engineering*, Technische Universität Berlin.
- Noardo, F., Guler, D., Fauth, J., Malacarne, G., Mastrolemba Ventura, S., Azenha, M., Olsson, P.-O. and Senger, L., 2022a. Unveiling the actual progress of digital building permit: Getting awareness through a critical state of the art review. *Building and Environment* 213, pp. 108854.
- Noardo, F., Wu, T., Arroyo Otori, K., Krijnen, T. and Stoter, J., 2022b. Ifc models for semi-automating common planning checks for building permits. *Automation in Construction* 134, pp. 104097.
- Noardo, F., Wu, T., Arroyo Otori, K., Krijnen, T., Tezerdi, H. and Stoter, J., 2020. Geobim for digital building permit process: Learning from a case study in rotterdam. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* VI-4/W1-2020, pp. 151–158.
- 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(8), pp. 307.
- Van Berlo, L., Dijkmans, T. and Stoter, J., 2013. Experiment for integrating Dutch 3D spatial planning and bim for checking building permits. In: *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Vol. 2number 2W1, ISPRS, pp. 279–284.
- Vitalis, S., 2021. ninja. <https://github.com/cityjson/ninja>.
- Zhang, J. and El-Gohary, N. M., 2015. Automated extraction of information from building information models into a semantic logic-based representation.