A prescriptive parametric model supporting performance-based design exploration at the early stages

Jimmy Abualdenien [1], Benedict Harder [1], Jan Clever [1]

[1]TU München, Chair of computational modeling and simulation

Abstract

Using Building Information Modeling (BIM), a model is developed through multiple design stages to satisfy various design and engineering requirements. The decisions made throughout the design stages, especially the early ones, steer a project's success and results. The impact of the decisions made in the early design stages is significant, as they form the basis of the following stages. In these stages, the uncertainty on how the design may evolve is high, as many decisions have not yet been made. Hence, several researchers have emphasized the advantages of complying to regulations and standards early on, to reduce the design space to feasible designs. This paper introduces a framework for exploring design options of train stations at the early stages. In more detail, two of the Deutsche Bahn regulation documents were analyzed, evaluated, and transformed into logical statements. Afterwards, based on the extracted rules, a parametric model was developed, where changing a parameter results into a new train station design option. The generation of a design option took into account the relationship between the parameters, avoiding any standard violations. The proposed approach was realized using the combination of both Autodesk Revit and Dynamo. The outcome of this study makes it possible to explore and evaluate the possible combination of design parameters, which supports design decisions.

Keywords: Early Stages, Parametric Modeling, Design Exploration, Prescriptive Design

1 Introduction

The process of designing a construction project is multidimensional and involves multiple interdependencies. Typically, a project is developed through multiple design stages, where in each it must fulfill various regulations and requirements, such as building codes and owner specific requirements [3]. In general, there are two main types of design: (1) prescriptive-based design: where the placement, dimensions, and detailing (such as material layers) of building elements are exactly adhering the information specified by regulations documents, (2) performance-based design: opens up the exploration of alternative possibilities by evaluating and identifying equivalent design options, such as exploring multiple material types with an equivalent fire rating [10]. Prescriptive-based design is a popular approach since decades [7]. However, with the advent and adoption of Building Information Modeling (BIM), performance-based design is becoming more common [10]. That is mainly because BIM provides the necessary means for managing the building's geometric, semantic, and topological information, which facilitates the exploration and evaluation of the feasible design space [1].

Prescriptive regulations provide broad requirements for building elements and functionalities [7], such as the minimum width of an escalator step or its raiser height. However, such requirements are useful for occupants' safety during the use of the building, however, they are not enough for identifying their influence on occupants' behavior during circulation or evacuation. Using performance-based design it is possible to address the unique features provided by the design and use of every building, considering the performance of the feasible design options. Accordingly, it promotes a better understanding of the interdependencies and interactions in the different use cases, such as in the event of fire. Prescriptive regulations could dictate that there must be at least two exists with minimum dimensions of a train station, however, designers need to identify whether these minimum requirements are capable of producing the optimal design solution for the pedestrians using the station through peak and normal day times (e.g., defining a specific level of service as a design goal). Therefore, it is necessary to combine both, prescriptive and performance-based design approaches to produce optimal and valid design solutions [2], such as the minimum width of an escalator step or its raiser height. However, such requirements are useful for occupants' safety during the use of the building, however, they are not enough for identifying their influence on occupants' behavior during circulation or evacuation. Using performance-based design it is possible to address the unique features provided by the design and use of every building, considering the performance of the feasible design options. Accordingly, it promotes a better understanding of the interdependencies and interactions in the different use cases, such as in the event of fire. Prescriptive regulations could dictate that there must be at least two exists with minimum dimensions of a train station, however, designers need to identify whether these minimum requirements are capable of producing the optimal design solution for the pedestrians using the station through peak and normal day times (e.g., defining a specific level of service as a design goal). Therefore, it is necessary to combine both, prescriptive and performance-based design approaches to produce optimal and valid design solutions.

To leverage the benefit of performance-based design, designers follow three main steps: (1) the generation multiple design options at every stage, for example by varying the window-to-wall ratio or material layers, (2) the evaluation of each design option according to multiple aspects, including evacuation or heating and cooling demand, (3) the comparison of the evaluated designs, informing the decisions through design process. The decisions made throughout the design stages, especially the early ones, steer a project's success and results [3]. The impact of the decisions made in the early design stages (conceptual and preliminary stages) is significant, as they form the basis of the following stages [8]. These stages, the uncertainty on how the design may evolve is high, as many decisions have not yet been made. Hence, several researchers have investigated and developed frameworks for managing [6], exchanging [11], and visualizing [4] the information uncertainty. Additionally, numerous techniques such as graph databases, simulations, and artificial intelligence were leveraged for evaluating the performance of building models. However, such techniques do not tackle the generation of design options that are valid and applicable with respect to regulations. This process is time intensive since it requires a careful integration of numerous rules [1].

This paper introduces a framework for exploring design options of train stations at the early stages. In more detail, two regulation documents of the German railways were analyzed, evaluated, and transformed into logical statements. Afterwards, based on the extracted rules, a parametric model was developed, where changing the values of the available parameters result into the generation of new valid design options. The generation of a design option took into account the relationship between the parameters, avoiding any standard violations. The proposed approach was realized using the combination of both Autodesk Revit[1] and Dynamo[1]. The contribution of this study is threefold: (1) we report on the process of transforming regulation documents into a computer-readable logic, identifying the percentage of convertible regulations, (2) we demonstrate the capabilities and limitations of using parametric models for the automatic generation of valid designs, (3) we propose an approach that facilitates the exploration of the feasible design space.

2 Background & Related Work

2.1 Design process at the early design stages

To emphasize on the interaction among the project participants during the early design stages, Figure 1 demonstrates an example of involving different domain experts in the design process to support the architect in making decisions based on the analysis and simulation results. In the beginning, the architect sends a rough building design including information regarding the building site as a request to the structural engineer asking for possible structural systems. The structural engineer develops suitable systems by taking into account the feasible variations of material, load-bearing walls, columns, and slabs thickness. Accordingly, the structural engineer responds with multiple options in which the architect reviews and chooses the systems that fulfill the boundary conditions and the owner's needs. The architect develops the selected designs further and sends an analysis request to the energy engineer to evaluate their performance, including *CO2* emissions as well as heating and cooling demand. Based on the performance provided as feedback, the architect makes a decision on which design option should be developed further in the next stage.

---

[1] https://www.autodesk.com/products/revit/overview | https://www.autodesk.com/products/dynamo-studio/overview

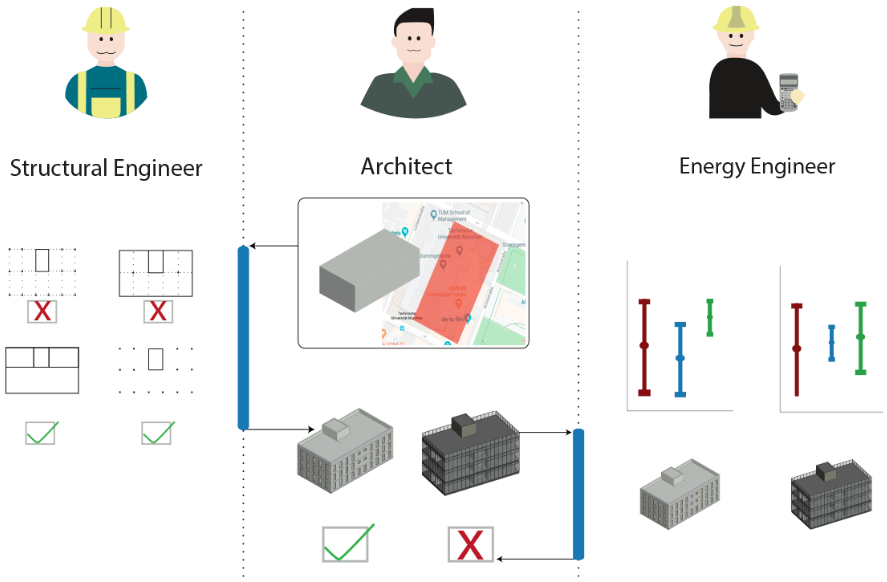


Figure 1: Design process in the early design stages - Simulation driven design: an example of the interaction and collaboration between the different disciplines to support design decisions in a way that fulfills the project requirements.

2.2 Parametric modeling

Developed by Shah and Mäntylä, parametric modelling or parametric CAD is a concept of modelling 2D sketches equipped with constraints such as parallel, coincide, tangential, etc. [9] and is widely implemented into a variety of software such as Autodesk Inventor, Siemens NX and more. The sketches are comprised of geometric objects such as lines, while the parametric constraints define the topology of these objects. The entirety is eventually solved by a Geometric Constraint Solver. Parametric modelling has a vast range of application. Parametric modelling can be leveraged for example to find suitable layout designs for various photo voltaic modules, accounting for different surface types and building components using BIM. Currently, there are multiple parametric modeling systems already integrated with the BIM-authoring tools, which facilitates they usage and integration in the design process. Examples of parametric modeling systems include, Autodesk Dynamo, which is integrated within Revit, and Grasshopper 3D, which is integrated within Rhinoceros 3D.

3 Methodology

The aim of this paper is to evaluate the ability of parametric modeling system to capture various and interconnected design requirements and produce regulations compliant design options for support design exploration from the early stages of design.

3.1 Embedding Regulations in a Parametric Model

In order to implement a given set of regulations into a parametric model they need to undergo a process of translation to be in a computer-readable form such as code before being used for the creation of a 3D-model (see Figure 2). This translative process is highly variable depending on the style of the regulation, which can take on forms such as a table, a formula, natural text and more. Accordingly, the translation from regulations to code must account for its corresponding form. The assessment of the regulations' types, must be first established:

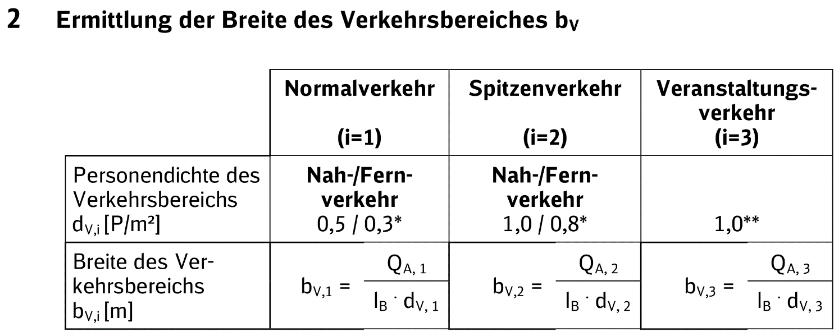
1. Natural language: regulations are described using natural text, with a specific language, where the requirements and conditions are specified.

2. Formula: the regulation is in the form of a mathematical formula with parameters and variables.

3. Table: certain parameters are determined dependent on other parameters, where their dependencies are described in the form tables.

By categorizing the rules according to their form, we can achieve said consistence by now determining the translative process needed to write code that directly corresponds to the regulation. Of course, natural language poses the largest problem here. This kind of regulation can either be handled manually by a person who must read, understand and then translate, or it can be handled by a natural language processing techniques. This software processing can translate natural language in order for computers to understand it. This knowledge can then be used for the computer to write a program. Using NLP successfully would be especially convenient for large-scale translation of regulation documents, but considering the scope of this paper and the number of regulations to translate, a manual approach is followed given the complexity of such techniques. As for the other types (Formula and Table), simpler approaches can be used. Programming languages such as Python offer a multitude of ways to achieve this. A determination table such as the one seen in the figure below for example can be easily translated into code by using nested if statements:

```
i= 2
regional=False
if i==1:
   if regional:
      d_v1=0.5
   else:
      d_v1=0.3
   b_v1=Q_A1/(I_B*d_v1)
elif i==2:
   if regional:
      d_v2=1
   else:
      d_v2=0.8
   b_v2=Q_A2/(I_B*d_v2)
elif i==3:
   d_v3=1
   b_v3=Q_A3/(I_B*d_v3)
```



2 Ermittlung der Breite des Verkehrsbereiches b_V

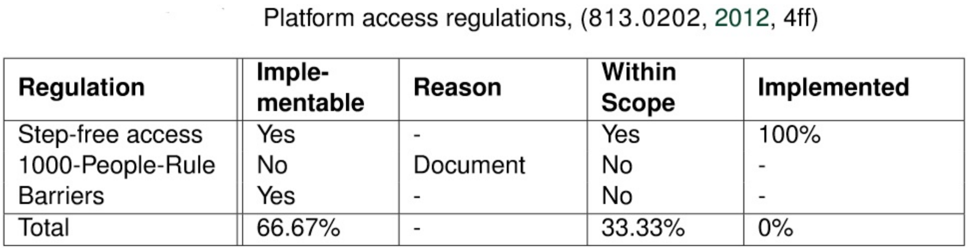
	Normalverkehr **(i=1)**	**Spitzenverkehr** **(i=2)**	**Veranstaltungsverkehr** **(i=3)**
Personendichte des Verkehrsbereichs $d_{V,i}$ [P/m²]	**Nah-/Fernverkehr** 0,5 / 0,3*	**Nah-/Fernverkehr** 1,0 / 0,8*	1,0**
Breite des Verkehrsbereichs $b_{V,i}$ [m]	$b_{V,1} = \frac{Q_{A,1}}{l_B \cdot d_{V,1}}$	$b_{V,2} = \frac{Q_{A,2}}{l_B \cdot d_{V,2}}$	$b_{V,3} = \frac{Q_{A,3}}{l_B \cdot d_{V,3}}$

Figure 2: Example of calculating the dimensions according to an equation from the regulations document. This table helps to determine certain parameters based on other given parameters.

4 Results

4.1 Statistics from RIL guidelines

The regulations from the RIL guidelines 813.0201 “Bahnsteige konstruieren und bemessen” and 813.0202 “Bahnsteigzugänge konstruieren und bemessen” were sorted to analyze the possibility and success of implementation statistically. The regulations were sorted based on their respective category, such as “Platform Length”, followed by an assessment of the individual rules on whether they are implementable and if not, why, whether they are within the scope of the thesis and lastly to what percent they have been implemented into the actual model. An example of the results of those categories, in this case the category “Platform Access”, can be seen in Figure 3.



Platform access regulations, (813.0202, 2012, 4ff)

Regulation	**Implementable**	**Reason**	**Within Scope**	**Implemented**
Step-free access	Yes	-	Yes	100%
1000-People-Rule	No	Document	No	-
Barriers	Yes	-	No	-
Total	66.67%	-	33.33%	0%

Figure 3: An example of the analysis of regulations

The analysis of regulations was done for each category, 13 in total, and then summarized in order to assess the results as a whole (shown in Figure 4).

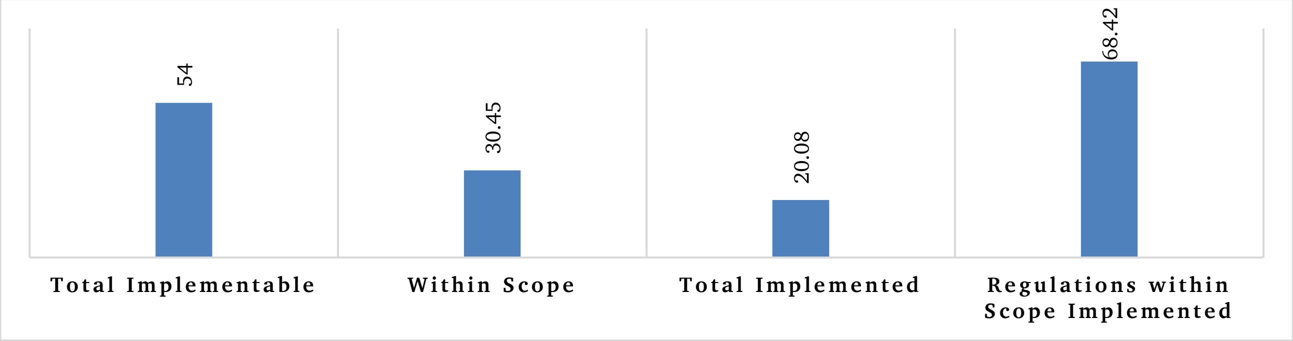


Figure 4: analysis results of both regulation documents [1, 2]

At first, around half of the total regulations seemed to be implementable. Many of the reasons why a certain regulation could not be implemented, where either it relies on the existence of a certain family type provided (whilst it was also out of scope of this research to create a similar family type oneself) or it was referring to an external guideline document, which we did not have in hand. Both of these problems can be bypassed, given that in a “real world application” one would have access to both. If we eliminate those two reasons our results look a lot more promising (shown in Figure 5).

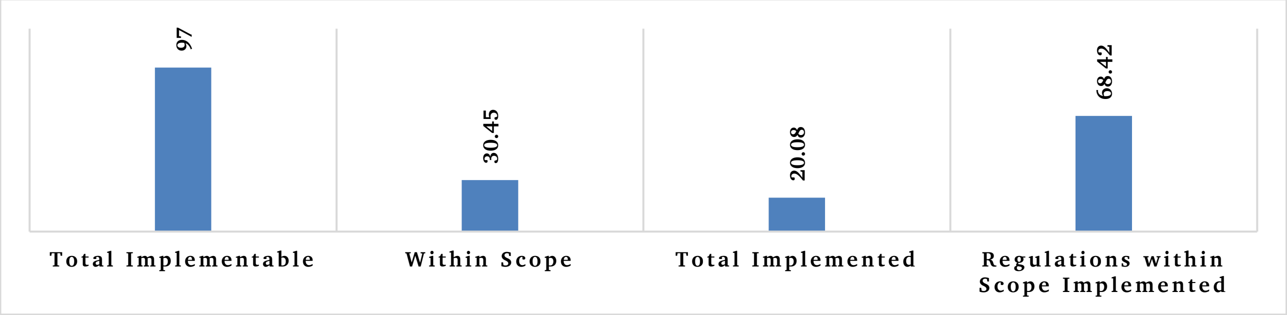


Figure 5: regulations analysis results after omitting rules that point to an external document that is not available in hand, or rules that rely on the presence of a particular family of building elements

One must consider, though, that even though 97% sounds quite promising as opposed to 54% (between both figures 4 and 5), the validity of these results can be questioned in some ways. For example, the way a regulation was deemed „implementable“ or not was purely by reading it and the assessing the possibility of implementation without a real „proof“ by attempting to implement it. This, of course, does not apply to the regulations that were actually implemented to by us in the Dynamo model.

4.2 Dynamo model

The implementation of the regulations into code was achieved by a mixture of Autodesk Revit and Dynamo. Using a Revit model including all necessary family types as a basis, a Dynamo script containing the input parameters as well as the Python scripts is run which then creates the model. The input parameters are mostly “Number Sliders” and “Booleans” which can easily be adjusted by the user. Based on the parameters, Dynamo interacts with the RevitAPI and constructs a regulation compliant model. An image of the entire Dynamo script as well as a sample train station generating using it can be seen below (see Figure 6). To the left, we can see the input nodes, which feed into the Base Script in the middle. This script collects the parameters as well as some basic construction such as slabs and walls, and then redirects the parameters to the appropriate script where more specialized tasks are done such as tack line creation, escalator placement and column creation. As explaining every aspect in detail would be out of the scope of this paper, a single example of how regulations can be implemented will be looked at in closer detail.

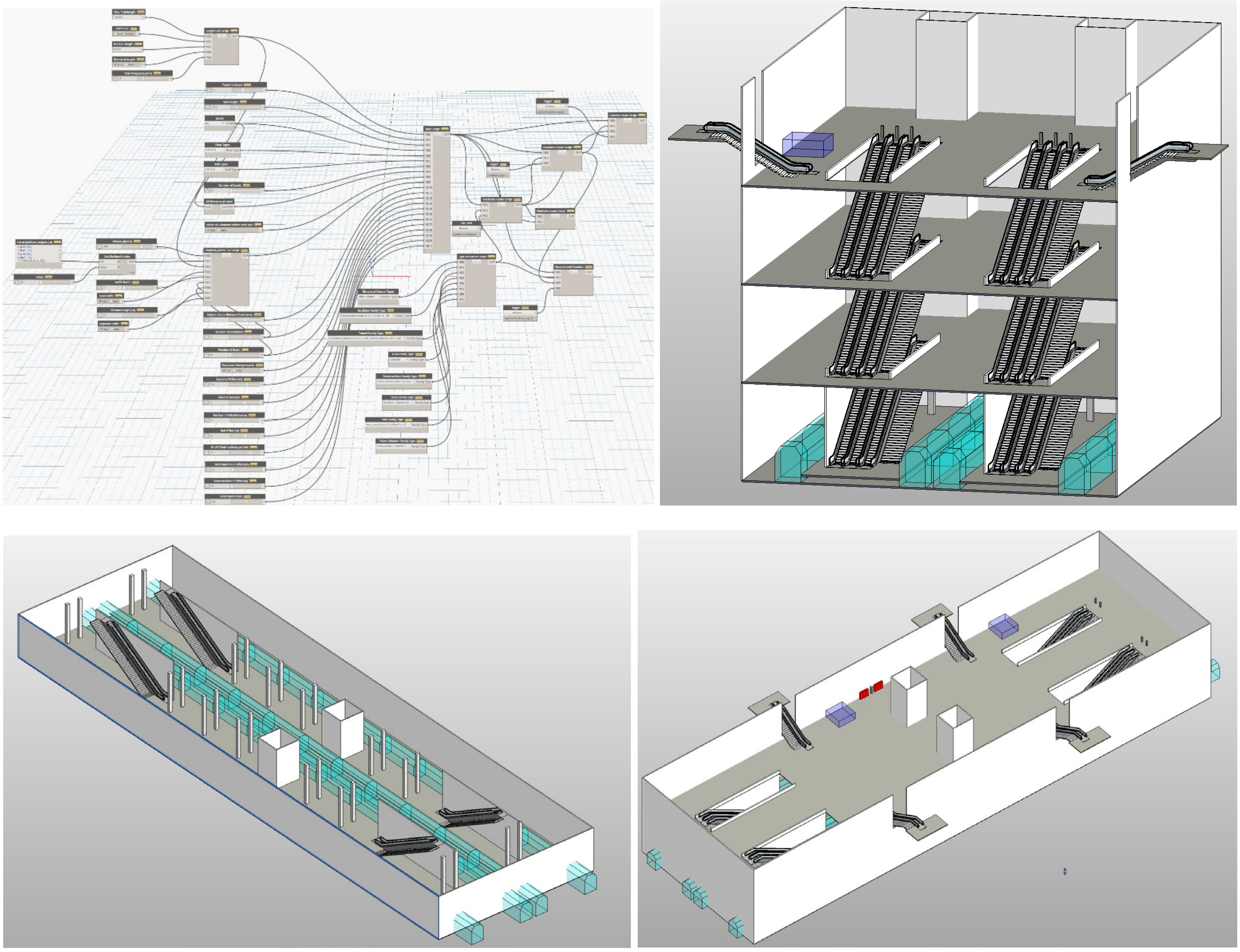


Figure 6: the developed dynamo script and a sample of the generated train station

4.2.1 Platform Length Calculation

The length of the platform is mainly determined by the maximum length of the trains halting at the station. Added to that are values such as the buffer of five meters (813.0201,2012, p. 7) and bumper-length (when the station is a terminus). Lastly, if selected, the length is raised to the next standard length, according to the regulations (813.0201, 2012,p. 7). So first of all, we add our values to the train length, dependent on whether we have a terminus or not:

```
if terminus==False:
   length=length+buffer
else:
   length = length + buffer + bumper
```

Depending on whether „Standard Length“ are wanted, the resulting length is increased based on the standard length immediately above it:

```
if standard_length==True:
   if length>= 60 and length< 90:
      length= 90
   elif length>= 90 and length< 140:
      length= 140
```

```
          …

          …
   elif length>=370 and length< 405:
      length = 405
```

4.2.2 Escalator Positioning

Apart from the platform length calculation, there also is the escalator placement. Objects on platforms must fulfill a minimum distance to the edge of the platform based on their respective length. Since the amount of escalators to be placed next to each other is variable (can be anything from 0 to 3), certain calculations must be done in order to guarantee that all escalators fulfill the requirement. Additionally, the user can specify a „Y-Offset“ which causes the escalators to move closer to either end of the platform, so this must also be considered and, if necessary overwritten. Let us have a look at the underlying code:

```
if UpperEscMoved:
        escY=escY+difference

#Check if the minimum distance is fulfilled (first escalator) and if not, move the escalator. This overwrites the
#EscalatorY-Offset
if n== 0:
       if (escY-frontEnd<minDistEsc):
               difference=(minDistEsc-(escY-))
               escY=escY+difference
               UpperEscMoved=True
        if backEnd-
(escY+(numberOfEscalators*revitApiEscalatorWidth+numberOfStairs*revitApiStairWidth))<minDistEsc:
                  difference=(minDistEsc-(backEnd-
(escY+(numberOfEscalators*revitApiEscalatorWidth+numberOfStairs*revitApiStairWidth))))*(-1)
                  esc=escY+difference
                  UpperEscMoved = True
```

Essentially, the distance to each platform edge (frontEnd and backEnd) is calculated and should it be lower than the minimum distance (minDistEsc) the difference is added for it to be regulation-compliant. Lastly, the UpperEscMoved flag is set to be true so that following escalators that are placed (note that this code is in a loop that iterated over the amount of escalators to be placed) are also moved the apporiate distance.

5 Conclusions & Future Research

As resource efficiency becomes more important, architects and engineers need to consider and evaluate numerous designs as an opportunity to increase the quality of their building. What was unachievable in the past, where every possibility had to be thought of, sketched, and evaluated, today’s technology allows Computer-Aided Design to reach new levels. Generative Design allows for designs to be automatically generated and assessed in quantities so large that they might have been seen as impossible in a not-so-far past. As discussed, this is especially important for the early stages of design, where decisions have the most impact. Additionally, the early stages are a great opportunity to implement the approach because of the level of detail, or even the lack thereof, that early-stage design has. An additional use-case for the developed parametric model is to generate a sufficient dataset for training neural networks [5], where the generated designs can be evaluated with energy or pedestrian simulations and then used for training a neural network that would be able to predict the designs performance in real-time.

As for possible future research and work, translating between natural language and code must be addressed. As useful as natural language processing and even natural language programming are, they create far too much overhead for the issue at hand. Processing natural language into executable rules is a complex and error-prone process. Hence, there is a clear need for a computer-readable prescriptive rules language that is capable of satisfying both conditions, human and computer readability. Developing such a language requires an extensive evaluation of national and international regulations to guarantee its flexibility and applicability.

Acknowledgements

We gratefully acknowledge the support of mFUND – Bundesministerium für Verkehr und digitale Infrastruktur (BMVI) for funding the research project BEYOND.

5 References

[1] 813.0201. 2012. *Bahnsteige konstruieren und bemessen*.

[2] 813.0202. 2012. *Bahnsteigzugänge konstruieren und bemessen*.

[3] Abualdenien, J. and Borrmann, A. 2019. A meta-model approach for formal specification and consistent management of multi-LOD building models. *Advanced Engineering Informatics* 40, 135–153.

[4] Abualdenien, J. and Borrmann, A. 2020. Vagueness visualization in building models across different design stages. *Advanced Engineering Informatics* 45, 101107.

[5] Clever, J., Abualdenien, J., and Borrmann, A. 2021. Deep learning approach for predicting pedestrian dynamics for transportation hubs in early design phases. In *28th International Workshop on Intelligent Computing in Engineering*, Berlin, Germany.

[6] Exner, H., Abualdenien, J., König, M., and Borrmann, A. 2019. Managing Building Design Variants at Multiple Development Levels. In *Proc. of the 36th International Council for Research and Innovation in Building and Construction (CIB W78)*, Newcastle, UK.

[7] Hurley, M. J. and Rosenbaum, E. R. 2019. *Performance-based fire safety design*. CRC Press.

[8] Schneider-Marin, P. and Abualdenien, J. 2019. A framework to facilitate an interdisciplinary design process using BIM. In *Proc. of the 31th Forum Bauinformatik*, Berlin, Germany.

[9] Shah, J. and Mäntylä, M. 1995. *Parametric and feature-based CAD/CAM: concepts, techniques, and applications*. John Wiley & Sons.

[10] Siddiqui, A. A., Ewer, J. A., Lawrence, P. J., Galea, E. R., and Frost, I. R. 2021. Building Information Modelling for performance-based Fire Safety Engineering analysis – A strategy for data sharing. *Journal of Building Engineering* 42, 102794.

[11] Zahedi, A., Abualdenien, J., Petzold, F., and Borrmann, A. 2019. Minimized communication protocol based on a multi-LOD meta-model for adaptive detailing of BIM models. In *Proc. of the 26th International Workshop on Intelligent Computing in Engineering*, Leuven, Belgium.