

# CONVERTING BRAZILIAN ACCESSIBILITY STANDARD FOR BIM-BASED CODE CHECKING USING RASE AND SMC

Eduardo A. Mendonça<sup>1</sup>, Leonardo Manzione<sup>2</sup>, and Eilif Hjelseth<sup>3</sup>

**Abstract:** The Architecture, Engineering and Construction (AEC) industry is regulated by a high number of rules given by laws, codes and standards, at national and international level. Much effort has been made to automate the checking of these rules, which is increasingly more efficient in terms of time and cost, such as the use of BIM tools. However, regulatory standards are written in human language. Wide and complex, they require substantial understanding and knowledge on the part of users. To allow the automation of the verification, it is first necessary a semantic analysis of the normative text, in order to adapt its meaning to a language that can be interpreted by computer software. This article demonstrates the application of the RASE methodology of semantic mark-up within the Brazilian accessibility standard, the configuration of Solibri Model Checker rules from these logical statements obtained and a simulation of these rules checking in a BIM model prepared for this purpose. The results were quantified for this paper and proved to be promising. They also identified the need for research to change the methodology of how the rules should be written so that they can be interpreted computationally.

**Keywords:** Code checking, RASE, Accessibility, BIM.

## 1 INTRODUCTION

Every project for Architecture, Engineering and Construction (AEC), whatever its size, needs to fulfil a series of requirements. The client's own needs program is one of them, which will vary from project to project. But there are sets of requirements that are repeated for different projects, such as the Constructions Codes and Accessibility Standards, for example. In these sets there are prescriptive requirements, more related to quantity, and performance requirements, which are also related to quality.

Verification of compliance with these requirements, made by a human, for a given project, requires long periods and is subject to the analyst's capabilities and the quality of the project's graphic representation, among other variables. This process, in addition to being time consuming, is highly susceptible to errors, omissions and lack of standardization.

In Computer Science, Model Checking refers to the test performed on a model to verify that it meets certain specifications. The use of a computational tool that performs this verification in BIM projects automatically explores two great potentials of the methodology: agility and reliability. However, there is an arduous task that precedes this automation. According to the process proposed by Eastman, et al. (2009), there are 4

---

<sup>1</sup> Specialization student, University of São Paulo (USP), São Paulo, Brazil, [emendonca@outlook.com.br](mailto:emendonca@outlook.com.br)

<sup>2</sup> Ph.D., Professor at University of São Paulo (USP), São Paulo, Brazil, [leonardo@coordenar.com.br](mailto:leonardo@coordenar.com.br)

<sup>3</sup> Ph.D., Professor at Norwegian University of Science and Technology (NTNU), Trondheim, Norway, [eilif.hjelseth@ntnu.no](mailto:eilif.hjelseth@ntnu.no)

stages of computer rule checking: (i) interpretation and structuring, that is, conversion of rules to computational language; (ii) preparation of the BIM model so that it can be correctly interpreted by the verification software; (iii) execution of the rule; and (iv) communication, a stage in which results are presented, which may still need human actions depending on their classification.

The present work, despite going through the four phases of Eastman, et al. (2009) will focus on the first two: rule interpretation and structuring them, and preparation of the BIM model. The following steps (execute and communicate) will be carried out in order to quantify the results. This paper is part of an ongoing research that aims to convert part of the Brazilian accessibility standards into testable logical statements, using the RASE methodology proposed by Hjelseth and Nisbet (2011). These statements will serve as a basis for configuring a commercial rule checking software, Solibri Model Checker, or SMC, in order to quantify and critically analyse its results. In order to achieve this objective, an existing building BIM prototype was developed, whose project had already been subjected to human verification of compliance with ABNT NBR 9050.

## 2 LITERATURE REVIEW

Many efforts have been made in the application of automatic rule checking. According to Dimyadi and Amor (2013), the idea of automating the process of checking compliance with code requirements has been explored since the 1960s. Some expressive cases reported by Eastman, et al. (2009) started in 1995 with CORENET Singapore, started with CAD drawings (Khemlani, 2005). Eastman, et al. (2009) still cite Norwegian Statsbygg's, the Cooperative Research Centre for Construction Innovation in Australia, the International Code Council (ICC), and the General Service Administration (GSA), both in the USA. Since then, there have been several approaches on the topic, however, until the early 2010s, few have evolved into a practical solution (Manziona, 2019).

From the last decade, however, there has been an increase in the number of searches. The dissemination of BIM tools certainly contributed to this, but the results showed that the process prior to the verification itself is essential and as arduous as predicted by Eastman, et al. (2009). Among the four steps proposed by the authors, the first step, of rules interpretation and conversion to a computational logic language, has been little explored.

The current building codes are a collection of many years of constructive experience, trials, errors and a series of accidents. For this reason, they are highly complex and quite extensive (Solihin, 2015). According to Hjelseth (2012), the regulations are written in legal / technical language for interpretation by a qualified professional. For the implementation in model verification software, the regulations must often be restructured to give a more applicable structure.

Some recent and recognized case studies have obtained results that reinforce the need for more research in this subject. Mainardi Neto (2016) simulated the application of Metrô (São Paulo's Subway Company) rules document and concluded that one of the most critical points in the automation in code checking is the translation of an existing rule into an applicable rule in a software for this use. Rodrigues (2015) carried out a similar survey on Portugal's Accessibility Code and pointed out that the way the regulation is drafted influences automatic verification and found that only about 38% of the requirements could be verified directly. Getuli et al. (2017) conducted research focused on systems interoperability for rule checking in Italy and highlight the importance of structuring the rules.

Hjelseth and Nisbet (2010) deepened the study to this issue. According to them, we see from the history of AI (artificial intelligence) and KBE (knowledge-based engineering) and KM (knowledge management) that capturing information in a meaningful and reliable way has not been a direct process. The relevant information in these documents need to be captured as rules for model checking in a time and cost effective way. Two procedures developed by these authors help capturing and converting that information. First, the T3-concept sort out data based on classification of regulations in three main type of methods for design rules: Translate, Transform and Transfer. Then, RASE methodology uses mark-up based on the four operators: requirement (R), applicabilities (A), selection (S) and exceptions (E) on normative text. It is a concept based on semantics for transforming normative documents into simple and well-defined rules that can be implemented in model verification software based on BIM / IFC. To confirm the accuracy of the process Hjelseth and Nisbet (2011), in one case study, re-transformed the logical statement back into prose, successfully.

According to Manzione (2019) it is a problem that will demand innovative solutions, aiming to improve the state of the art of regulatory knowledge, while also allowing other domains, such as the legal one, to develop its rules directly in a computational format. This paradigm shift will alter the current, paper-based standard, which requires human interpretive effort, towards the electronic, precise, controllable and reliable standard. The legal and normative databases may, in the future, have, through interoperability, a direct link to the modelling and content search software, through the Internet.

### 3 METHODOLOGY

This work will follow the Design Science Research (DSR) methodology, according to the methods proposed by Dresch, et al. (2015), through which the problem will be properly studied and understood, the existing artefacts will be researched while carrying out a systematic review of the literature on the theme and a solution to the problem will be proposed, describing the weaknesses and potentialities of the process.

In this paper, the survey of accessibility rules will be restricted to Section 6 of ABNT NBR 9050:2015. The data obtained will be quantified and presented graphically. The general conclusion will seek to identify the strengths and weaknesses, difficulties and needs to improve the process, be it in the elaboration of rules, BIM modelling and automatic verification scheme.

In summary, the work is structured to answer the questions on Figure 1.

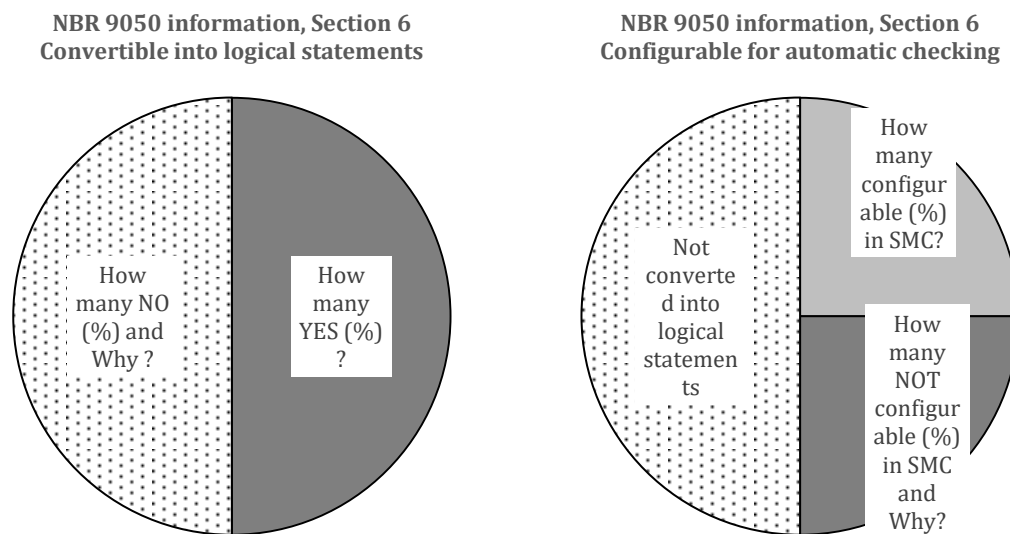


Figure 1: Questions to be answered

#### 4 EXPERIMENT AND RESULTS

This research topic is ordered according to the 4 steps of the automatic rules verification process proposed by Eastman, et al. (2009) with a small adaptation of the last two, in order to deepen the description of the rule execution (third step), and reduce the reporting phase of the software results (fourth step).

The first stage (item 4.1) aims at the interpretation and structuring of the rules. The basic text is Section 6 of NBR 9050, entitled Accesses and Circulation. The second stage (item 4.2) is the preparation of the BIM model of a real building (Figure 2). This is the Professor Cândido de Oliveira public school in São Paulo, which was the subject of a contract with the Foundation for the Development of Education (FDE) for the development of executive projects to adapt the building to accessibility, throughout 2018. The original building was built in 1977, prior to the publication of the first edition of the NBR 9050 standard, dated 1985. That contract with FDE required full compliance with the third edition of the standard, of 2015, currently in force. Thus, the solutions proposed in the project fully considered what the standard under study requires.

It should be noted that this contract was carried out in the "conventional" way of design, coordination and inspection, that is, with the development of the project in CAD software, analysis of "2D" drawings (Plans, Sections, Elevations and Details) and comments made on drawing boards, in addition to visual verification of compliance with design requirements. The project was approved by the FDE, which shows that its solutions have been validated by qualified professionals and that its elements meet the requirements of NBR 9050:2015.

The third stage proposed by Eastman, et al. (2009) is the execution of the rule. Item 4.3 will deepen the description of the SMC configuration process to check the statements produced in step 1 in the model developed in step 2.

The fourth stage is communication, that is, how the software "reports" the occurrences to the user, which can demand new actions according to the result. In this work, item 4.4 will present quantitatively the results of the verification performed in step 3. This will allow to graphically observe which portion of NBR 9050 could be verified in the process.



Figure 2: BIM Model of the real building (Source: author; ArchiCAD screen)

#### 4.1 Step 1 - Conversion of the normative text

This step consists of applying the T3 concept over the text of Section 6 of NBR 9050 and applying the RASE methodology on items classified as T1 and T2.

**T3 Concept.** This topic presents and quantifies the results of the T3 concept applied to the text of Section 6 of NBR 9050 “*Accesses and Circulation*”, composed of 15 sub-items. Its content is written in prose, with numbered paragraphs, including tables and figures. Of the 15 sub-items, 3 were excluded from the scope of this research because they did not apply to the building under study: 6.10 “*Electromechanical circulation equipment*”, 6.12 “*External circulation*” and 6.13 “*Pedestrian walkway*”. The remaining 12 sub-items were transcribed to an Excel spreadsheet, with each paragraph occupying a cell. Each cell was interpreted and classified as recommended by T3:

- T1 (translate): normative text with clear metrics, which can be directly submitted to RASE marking;
- T2 (transform): the text needs to be reformulated based on its intentions, so that it is submitted to RASE from the reformulation;
- T3 (transfer): the text contains generic statements, with subjective metrics or even without clear metrics that, in order to be met, need to be assessed “manually”, whose decision must be made by a specialist, that is, not subject to automatic verification.

Outside the written text, relevant and measurable information was found in 29 figures and 3 tables, consequently indispensable for the RASE marking. This information was transformed into simple sentences with measurable metrics, therefore classified as T2.

Conversely, redundant stretches were perceived with other parts of the Standard itself, as well as definitions and explanatory notes, all without direct relevance to the RASE marking, necessary only for the understanding of the adjacent elements. They were classified as:

- Redundant: content that is repeated in some part of the Standard with the same objective;

- Definitions: content not applicable because it presents definitions, justifications or explanatory notes, necessary only for the general understanding of the text, without implying recommendations or metrics;

Some sentences in the original text needed to be subdivided into smaller sentences that resulted in different classifications. For example, item 6.11.2.6 “*Doors must be able to be opened with a single movement, and their handles must be of the lever type, installed at a height between 0.80 m and 1.10 m (...)*” can be rewritten in two sentences: “Doors must be able to be opened with a single movement” and “door handles must be of the lever type, installed at a height between 0.80 m and 1.10 m”. In this case, the first sentence was classified as T3, given the non-measurable character of the act of opening the door and the second sentence was classified as T1, as it contains a clear subject and predicate, with well-defined metrics.

Examples of the classification applied to the text:

T1 (translate): item 6.6.4.1 “*When there is a door on the landing, its sweep area cannot interfere with the minimum dimension of the landing.*” Its interpretation results in a logical statement of simple verification: there is an application (doors on the landing) and a requirement (do not advance on the minimum threshold size).

T2 (transform): most of the T2 items need only restructuring in order to result in minor logical statements. Item 6.3.3 consists of a paragraph with several sentences. Its content can be rewritten with simple logical statements as shown in Table 1.

Table 1: Restructuring item 6.3.3 of NBR 9050.

Original text	Text rewritten in logical statements
<p>The transverse slope of the surface must be up to 2% for indoor floors and up to 3% for outdoor floors. The longitudinal slope of the surface must be less than 5%. Slopes equal to or greater than 5% are considered ramps and, therefore, must meet 6.6.</p>	<p>Internal floors must have a transversal inclination less than or equal to 2%.</p>
	<p>External floors must have a transversal inclination less than or equal to 3%.</p>
	<p>Internal and external floors must have a longitudinal slope of less than 5%.</p>
	<p>Floors with a longitudinal slope greater than or equal to 5% are considered ramps must meet 6.6.</p>

As for item 6.6.2.6, it presents its information through a figure: “*Every ramp must have a handrail of two heights on each side, as shown in Figure 72*”. Figure 72 of the Standard is reproduced in Figure 3.

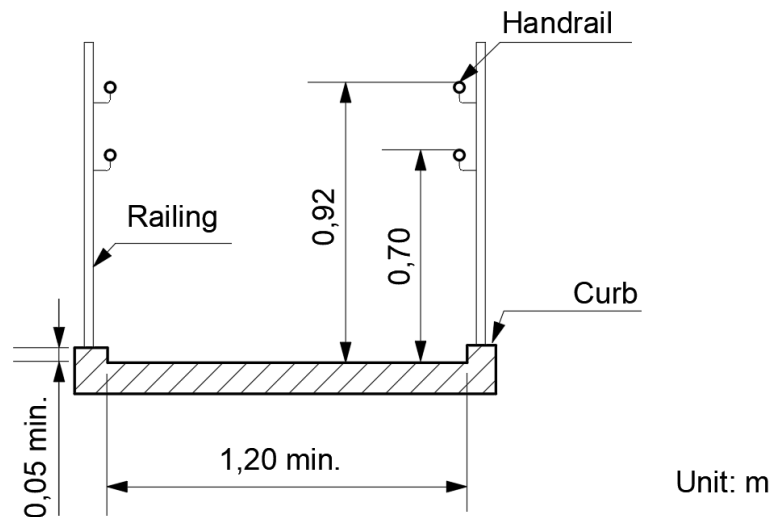


Figure 3: Figure 72 from NBR 9050 (Source: adapted from ABNT (2015))

Measurable data were taken from the figure and rewritten as a clause classified as T2: “ramps handrails must be 70 cm and 92 cm high”. It is important to note that this item would be redundant with 6.9.2.1, subitem of “*Handrails and guardrails*”, but it appears here as a ramp requirement (having a handrail).

T3 (transfer): according to item 6.9.1, handrails “*must be firmly fixed to the walls or to the support bars, ensuring safe conditions of use (...)*”. How to measure or quantify “safe conditions of use”? Should be transferred for manual evaluation by a specialist.

Redundant: The term “*when installed in sports venues, doors must have a minimum clearance of 1.00 m*” appears in items 6.11.2.4 and 6.11.2.12. In the first occurrence it was rewritten (T2) and in the second it was classified as redundant.

Definitions: item 6.6.2 “*To ensure that a ramp is accessible, the maximum slope limits, the unevenness to be overcome and the maximum number of segments are defined*”. There are no parameters applicable to the verification, which appear in the continuation of the item.

Thus, excluding titles, subtitles and items outside the scope, the classifications above resulted in 232 (two hundred and thirty-two) information, distributed as in Table 2. It is noteworthy that less than 10% of the analysed content contains clear metrics (T1), allowing conversion to computational language only using the RASE operators. Items of the Standard that could be classified as T1, given the presence of clear metrics, but written in more than one sentence (phrase), were classified in this work as T2 (transform), because according to Hjelseth and Nisbet (2011) the intention of explicitly using the source text transformation is to expand the number of rules that can be extracted from the original text.

Positive fact is that more than 3/4 of the content (sum of T1 and T2, or 175 items) can be submitted to RASE marking for conversion into testable logical statements, as discussed in the next topic.

Another important data for the research is the “total requirements”. Excluding redundant definitions and phrases, it is the sum of T1, T2 and T3, equal to 211. This value expresses the number of requirements identified in this part of the Standard, regardless of its understanding. It will serve as a comparison with the results sought.

Table 2: Information extracted from the normative text (T3 concept).

Identification	Occurrences	Percentage
T1 (translate)	22	9,5%
T2 (transform)	153	65,9%
T3 (transfer)	36	15,5%
Redundant	6	2,6%
Definitions	15	6,5%
<b>TOTAL</b>	<b>232</b>	<b>100,0%</b>

**RASE Methodology.** RASE consists of identifying, in each statement, the four logical operators: requirement (R); applicability (A); selection (S); and exception (E). The 175 (one hundred and seventy-five) logical statements resulting from T3 will be submitted to RASE marking.

According to Hjelseth and Nisbet (2011), the most obvious and easily identified are ‘requirements’, as they are usually associated with the imperative “must” or “shall”. In the statement “*route between vehicle parking and access must make up an accessible route*”, the requirement is ‘make up an accessible route’.

Then there will always be a term that identifies to whom or to whom the requirement applies. In the previous example, ‘applicability’ is ‘the path between vehicle parking and accesses’.

Eventually the text will present a ‘selection’ of this applicability, something like a subset when it splits. For example, “*for curved ramps, the maximum allowable inclination is 8.33%*”: we have ‘ramps’ as an application and ‘curved’ as a ‘selection’ of the ramp type.

‘Exception’ concept can be as simple as “*ramps mentioned in 10.4 (audience and stages) except for this requirement*”: just exclude audience and stage ramps from the requirement applied to this rule. Or more complex as “*every accessible route must be provided with natural or artificial lighting with a minimum illuminance level of 150 lux (...). Lower levels of illuminance are accepted for specific environments, such as cinemas (...)*”. Here, the second sentence presents an exception to the first: ‘environments, except cinemas, must be provided with lighting with a minimum illuminance level of 150 lux’.

Hjelseth and Nisbet (2011) point out that the sentences marked with the four operators, requirement (R), applicability (A), selection (S) and exception (E), will contain metric phrases, to which an object, a property, a comparator and a target value can be systematically assigned. The object and property should ideally be elaborated from terms classified by standardized systems. In this work they will be transcribed directly from the terms of the accessibility standard, aiming whenever possible to match the terms that will be used in the model. The target value can be numeric, with any unit, for which the comparator will be “equal”, “less than”, “greater than” or its variants. If the target value is descriptive, the only relevant comparators will be “equal” or “different”. If it still refers to a group of elements, comparators can be “include” or “exclude” for any element in the group.

The following example shows the conversion of item 6.6.2.4 from its original text to logical statements.



- Original text: "*Cross slope cannot exceed 2% on internal ramps and 3% on external ramps.*"

Text converted into logical statements:

- Cross slope [R] of internal [S] ramps [A] cannot exceed 2% [R];
- Cross slope [R] of external [S] ramps [A] cannot exceed 3% [R].

Table 3 shows the attribution of objects, properties and values to the respective metric phrases.

Table 3: Metric phrases with assigned objects, properties and values.

Metric phrase	Type (RASE)	Object	Property	Comparator	Target value	Unit
Ramp	Applicability	Element	Type	Includes	ramp	
Internal	Selection	Localization	Internal	=	TRUE	
maximum cross slope 2%	Requirement	Ramp	Cross slope	≤	2	%
Ramp	Applicability	Element	Type	Includes	ramp	
External	Selection	Localization	Internal	=	FALSE	
maximum cross slope 3%	Requirement	Ramp	Cross slope	≤	3	%

After the assignment of objects, properties, comparators and targets to the logical statements resulting from the previous step (T3 concept), a spreadsheet was built with all the data obtained by RASE. These data will be used to feed SMC, a process described in step 3 of setting the rules.

In this spreadsheet, each of the 175 requirements was also marked with "YES" or "NO", depending on its possibility of application to the model under study. Those that are not applicable can refer to elements that do not exist in the real building (curved ramp, for example), or not modelled in the Architecture discipline (visual communication, i.e.). The obtained result showed that 79 requirements (45%) are applicable to the prototype, against 96 not applicable (55%).

## 4.2 Step 2 - Model Preparation

For the rule-checking software to read the model correctly, it must meet certain requirements. For example, for the verification of the inclination of a ramp to be possible, it must be read as such, and not as a tile. According to Eastman, et al. (2009), there must be shared conventions in relation to the coded rules so that they correspond to the properties and structures incorporated in the construction model.

In order to resolve this, among many other issues, there are classification systems. This research sought to adopt the recommendations of ABNT NBR 15.965 "*Construction information classification system*", based on the OmniClass (or OCCS) classification proposed by the CSI (Construction Specifications Institute, Inc.). The modeling software

used (ArchiCAD, version 22) presents this applicable classification system as a simple attribute of the element. In addition to be an international classification recommendation, aiming at the use of this work in other countries.

Within the model, 'Zone' is a spatial delimitation, with its own properties (Figure 4). In addition to the classification, in order to allow accessibility verification, it is necessary to inform additional properties of the Zone in the modelling software. In ArchiCAD, in the same classification window we can enter the following parameters, which will be input data in the verification software:

- renovation status: existing, to be demolished or to be built;
- openings: emergency exit (true or false) and accessible (ditto);
- floor: non-skid (true or false);
- property: private, rented, shared or public;
- lighting level: numerical value expressed in lux;

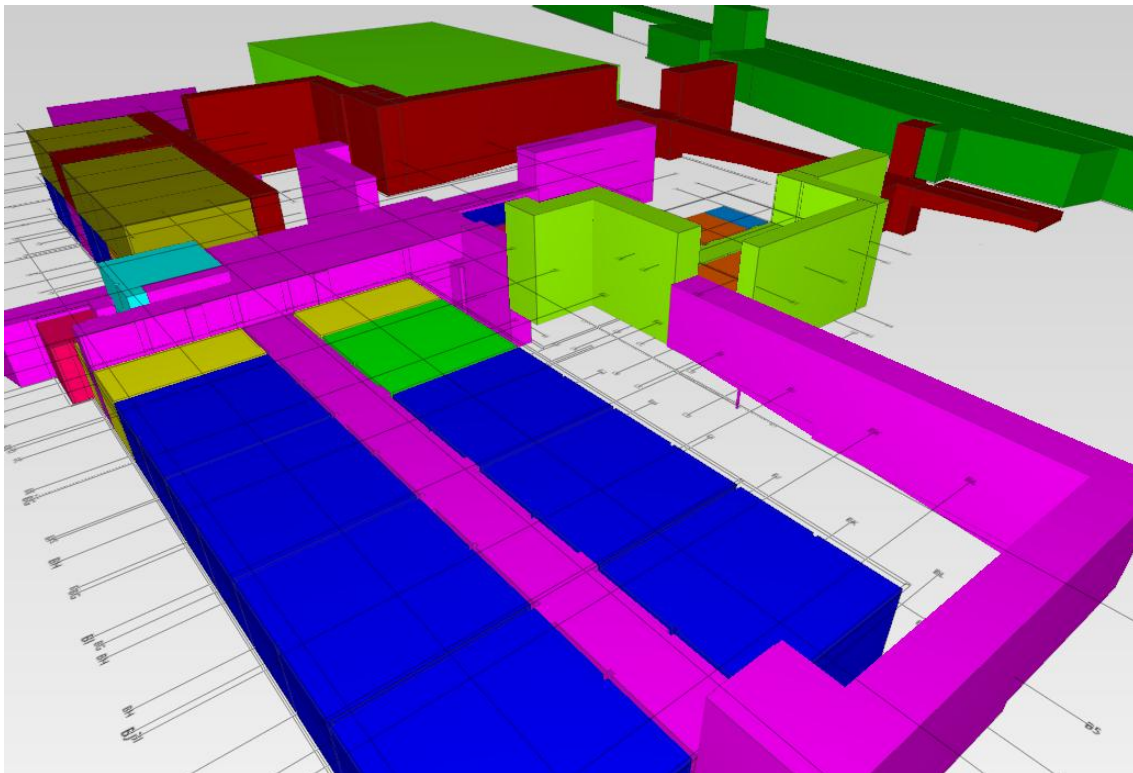


Figure 4: Model Zones (Source: author; SMC screen)

If not configured, the above parameters would be identified as 'undefined' in the model. This would not prevent the verification of rules, but they should be correctly filtered in the verification software to obtain reliable results.

During the tests carried out, it was also noticed the need to add a specific configuration in the zone's *IfcIdentifier* so that the Accessible Route of the model is identified, within the group of properties *PSet\_SpaceCommon*. The simple marking that the opening is accessible does not define a Zone as an Accessible Route. As the definition of this property is descriptive, in this work the term "accessible route" was adopted for the spaces that make up this route.

In addition to the Zones, the correct reading of the model by the verification software also depends on the classification of the elements (ramps, stairs, guardrails, etc.). The

first tests revealed that an “L” ramp will not be recognized in this way just by classifying the Zone that surrounds it as ‘ramp’. The stretch of two flights shown in Figure 5 was modelled with independent elements: flights, landing, guardrails and curbs, and around it was defined its Zone involving all these elements. This was not enough for SMC to consider this set as a single ramp, nor its level as an inter-daily level (which is subject to specific rules, different from the start and end levels).

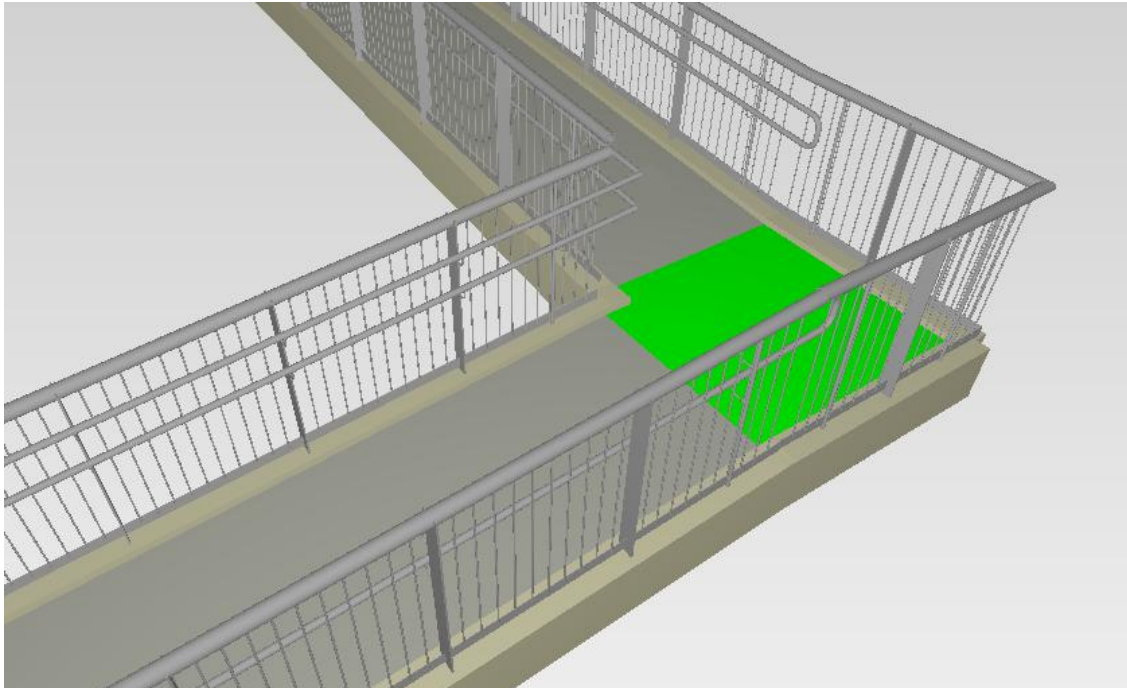


Figure 5: "L" ramp with intermediate landing selected (Source: author; SMC screen)

These properties are not usually reported in a model at the design phases. As well observed by Eastman, et al. (2009), the models created so far do not normally include the level of detail needed for construction codes or other types of rules verification. When including them, however, it is convenient that they go along with the configuration of the rules.

Thus, in this work, all components of a ramp, necessary to comply with NBR 9050 will be classified by the code OmniClass 21-02 10 10 50 Ramps.

Finally, in order to efficiently exchange the model file with other software (SMC, i.e.), the format used was the IFC (Industry Foundation Classes). This format guarantees, according to tests carried out during the research, that the OmniClass classification of spaces and elements is preserved in the export and import processes.

### 4.3 Step 3 - Setting the rules

This step's aim is, from the table of metric phrases produced in Step 1 - Conversion of the normative text, to configure the rules verification software. To expose the logical structure created when applying RASE statements (requirement, applicability, selection or exception), this work produced some flowcharts. The example below (Figure 6) is part of the flowchart created for the ‘floor’ object. The complete flowchart is not only the sequence of decisions of an item of the Standard, but the combination of all the requirements applicable to that object. In another Brazilian research, Andrade and Silva (2017) realizes that efforts to link the requirements of the standard to the SMC check

were not carried out directly, that is, on average, more than 4 software rules to verify a single requirement of the standard.

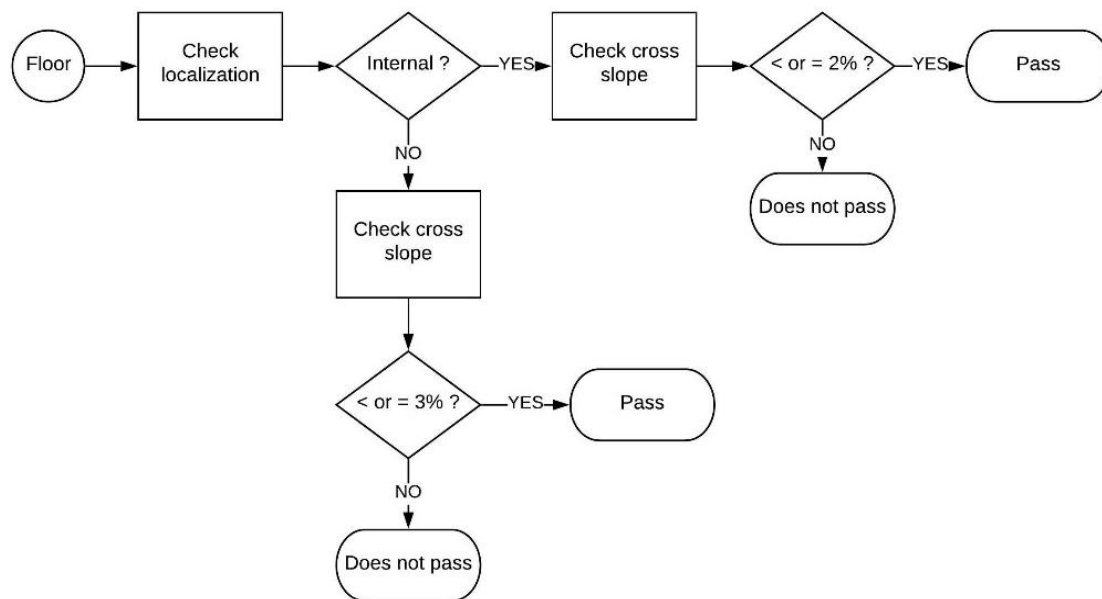
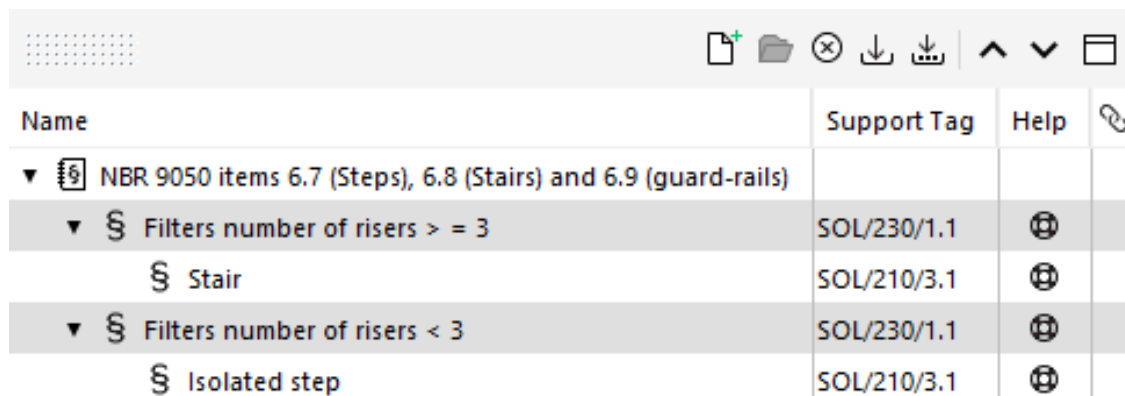


Figure 6: Part of the decision flowchart for the ‘floor’ object (Source: author)

The dynamics of the verification, in a very simplified way, can be described as the comparison between the properties of a model object and the rule parameters, the results of which can be "pass" (totally or partially) or "does not pass". For this, the rule parameters must consider the model classifications. An example is the SMC rule #132 *Space Area*: it checks whether the areas of certain spaces are within a specific range of values, configured by the user. For that, it will search for the spaces determined in her filter, check the ‘area’ property of that space and compare the values. There are more complex rules that check interference between components, connections between spaces, rates, proportions, etc., but it will always depend on the correct configuration of the filters and parameters.

It is also possible to combine more than one rule (its repetition or a different rule) to achieve a certain objective. One of the functions of this combination is called *Gatekeeper*. Nesting one rule under another will cause the bottom rule to be checked only if a certain condition of the top rule is passed. This allows combining rules to comply with the decision sequence assembled in the flowchart. Thus, the Rulesets will result from the Flowcharts, which in turn originated from the RASE methodology.

An example of this combination is the distinction between 'isolated step' and 'stair'. The modelling software creates these two objects with the same tool, and both are likely to receive the same classification, but the accessibility standard has different requirements to them. In NBR 9050, only the sequence of three risers or more will be considered a stair. Figure 7 shows the fulfilment of the following condition: how many risers are there in an object 'stair'? If the answer is greater than or equal '3', the object will be driven to 'stair' rule checking defined in the Standard. Otherwise, the object goes to the 'isolated steps' checking.



Name	Support Tag	Help	
▼ ⓘ NBR 9050 items 6.7 (Steps), 6.8 (Stairs) and 6.9 (guard-rails)			
▼ § Filters number of risers >= 3	SOL/230/1.1	⊕	
§ Stair	SOL/210/3.1	⊕	
▼ § Filters number of risers < 3	SOL/230/1.1	⊕	
§ Isolated step	SOL/210/3.1	⊕	

Figure 7: Nested rules (Source: author; SMC screen)

Besides that, rules can be configured to request some action from the user before their execution. This can be used when the choice between flowchart paths depends on properties not fed into the model, that is, it will not result from the comparison between similar objects. For example, there are requirements for residential buildings that do not apply to public buildings. If this property (building usage) is not part of the model, as it is not characteristic of a specific object, the rule will ask the user for a task to specify the building usage. These tasks will be shown in the SMC To Do View.

In the checking communication, SMC gathers the events according to their level of severity, between high, medium or low. These levels can be configured according to minimum and maximum limits established by the user. In this research, these levels were not adjusted, since its main objective is to assess how much is subject to automation, regardless of the type of non-compliance with Standard.

#### 4.4 Step 4 – Presenting Checking results

In order to answer the question above, the same spreadsheet built for RASE was fed with an additional column, whose content is: Was the requirement possible or not to be configured for SMC Checking? The total number of YES responses will be the comparative data with the number of “automated” requirements in Section 6 of NBR 9050 (the sum of statements T1 and T2 from step 1). It will also be compared with the “total rules” in Section 6, including texts classified as T3. Figure 8 shows the Checking and Results Summary Views of SMC.

To determine whether or not a requirement could be configured in SMC, several simulations were performed. The elements that were not verified due to some deficiency of the model were classified as YES, since their automation proved to be possible. Obviously, the results of rules that “did not pass” were also classified as YES, since the objective is not to ratify the model's compliance with the Standard.

It was also noted YES for rules that can be verified in the way that the Standard describes it, even though it refers to external links. Example: an exception in item 6.1.1.1 was rewritten as “*lower levels of illuminance are accepted for specific environments, such as cinemas, theatres or others, according to specific technical standards*”. Solibri will exclude the verification illuminance level of these environments, and it can be pointed out that this environment needs an action from the verifier, even if the “specific technical standards” are not configured for verification.

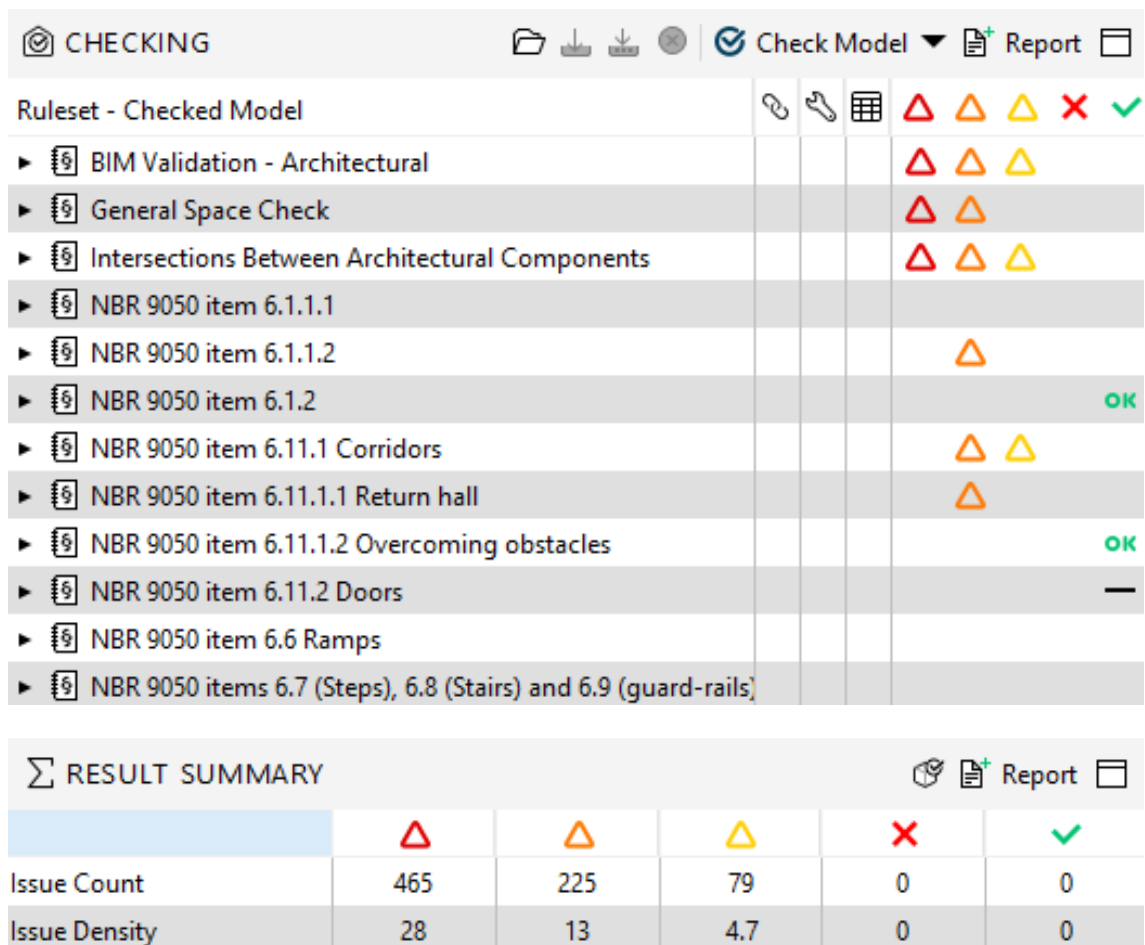


Figure 8: Reported results (Source: author; SMC screen)

Likewise, possible rules for verification by the software, despite the need for an extra action by the verifier, were classified as YES. Example: one of the logical statements in item 6.2.2 is “the main building entrance, or the entrance for the largest number of people, has the obligation to meet all accessibility conditions.” It is likely that the model does not contain the information that an entry, not being the main one, is the access of the largest number of people. In this case, a task prior to the rule's execution must require the verifier to inform the expected number of people for all entries.

Situations in which the Standard determines, for a certain parameter, a minimum value and another recommended value, were classified as YES, however the recommended value was ignored. In the case of “the width for ramps on accessible routes must be at least 1.20 m, and 1.50 m recommended”, a rule was created to check compliance with the minimum value (1.20 m).

Requirements that could not feed any Solibri rule were classified as NO. For example, the statement “handrails must be 70 cm and 92 cm high” could not be verified. Solibri can check if the handrail height is within a certain range, but not a double handrail (two heights).

There were original phrases, broken down into two logical statements, whose classifications were NO for the first and YES for the second. Item 6.2.7 says that “revolving doors should be avoided, but when they are installed (...) another entrance should be provided, which guarantees accessibility conditions”. The software will not be

able to check whether the choice of a revolving door could have been avoided, but whether there is another accessible entry associated with it.

Simple sentences, but with subjective conditions were classified as NO. According to item 6.6.2.2, “*in renovations, when the possibilities of solutions that fully meet Table 6 are exhausted, inclinations greater than 8.33% can be used (...)*”. The “exhaustion” of possibilities cannot be verified automatically. The same goes for expressions like “*as long as technically justified*” and similar.

From the quantitative premises described, the results obtained are as follows, graphically represented in Figure 9:

- Amount of information (including Redundant and Definitions) = 232 (100%);
- Total requirements (T1 + T2 + T3) = 211 (90.9%);
- Logical statements (T1 + T2) = 175 (75.4%);

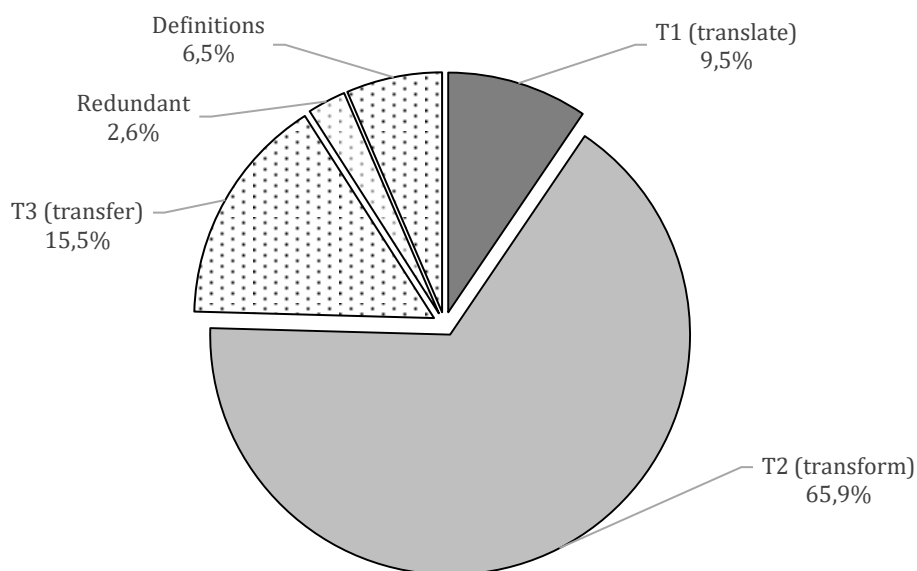


Figure 9: NBR 9050 - Distribution of NBR 9050 Section 6 information according to T3

Of these data, work showed that 108 requirements (46.5% of total requirements or 61.7% of logical statements) are possible to check in SMC. Figure 10 shows how the questions raised in Figure 1 were answered.

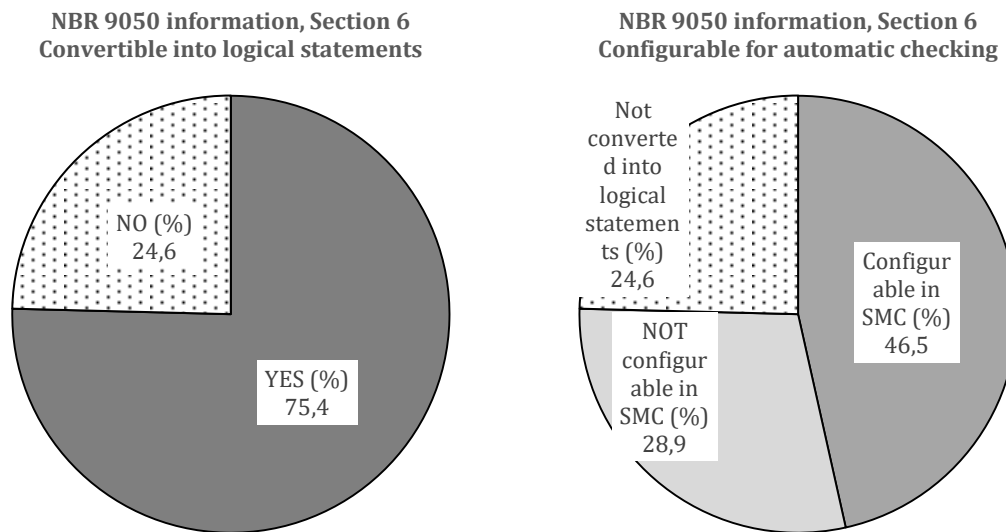


Figure 10: Research results

## 5 CONCLUSIONS

This paper submitted a section of the Brazilian accessibility code to T3 and RASE methodologies and delineated a set of rules in Solibri Model Checker, based on the resulted statements from the process. For this purpose, data within the Standard was quantified and results showed that almost half of that information was able to be checked in SMC and one third part could be converted into formal logic but was not able to be checked, due to software limitations. It also identified how much could not be converted into formal logic and why. This approach can be a systematic way to assessment of code provisions.

The application of T3 and RASE methodologies to the regulatory text revealed issues to be observed when it is intended to prepare it for BIM code checking. These questions mainly refer to deficiencies present in the way it is written. There are intentionally subjective requirements in the standard, like performance-based prescriptions, which gives the designer a desirable freedom. However, even for the technical aspects and quantitative items, the description through prose text, in a language written by and for humans, does not support automation. Even the association with figures, which is intended to facilitate the user's understanding, requires a logical structure for automatic checking.

Standard text should ideally be drawn from terms classified by standardized classification systems (e.g. OmniClass), so that the rule property and the model object are compatible. Likewise, designers must adopt standardized classifications and create a routine of feeding the model with the necessary information during the design process. Verification software alone will not perform its potential if these methodologies are not applied first. The strength of this process relies on the connection between text (codes) and the software (technology), provided by RASE (methodology).

Future researches could focus on evolving normative writing, like a text editor based on RASE, which would inform the legislator of the lack of any logical data. For modelling, developers can incorporate verification tools in their software. The creation of new checking software will be beneficial for the evolution of the systems.



## 6 ACKNOWLEDGMENTS

First author thanks David Oliveira and CADTec, for the courtesy of Solibri Model Checker.

## 7 REFERENCES

- ABNT - Associação Brasileira de Normas Técnicas (2015). *NBR 9050:2015 - Acessibilidade a edificações, mobiliário, espaços e equipamentos urbanos*. Rio de Janeiro: ABNT.
- Andrade and Silva, F. P. (2017). *Verificação automática dos requisitos de projetos da Norma de desempenho pela plataforma BIM Solibri Model Checker*. Masters Diss. Escola de Engenharia da Universidade Federal de Minas Gerais.
- Dimyadi, J. and Amor, R. (2013). Automated Building Code Compliance Checking: Where is it at?. In: *19th CIB WBC 2017*. Brisbane, v. 6, p. 1.
- Dresch, A., Lacerda, D. P. and Antunes Júnior, J. A. V. (2015). *Design science research: método de pesquisa para avanço da ciência e tecnologia*. Porto Alegre: Bookman.
- Eastman, C., Lee, Jeong, Y.S. and Lee, J. (2009). Automatic rule-based checking of building designs. In: *Automation in Construction*. [online] Georgia: Elsevier, pp. 1011-1033: <https://doi.org/10.1016/j.autcon.2009.07.002>
- Getuli, V., Ventura, S. M., Capone, P. and Ciribini, A. L. (2017). BIM-based code checking for construction health and safety. In: *CCC 2017*, Primosten: Elsevier, pp. 454-461: <https://doi.org/10.1016/j.proeng.2017.07.224>
- Hjelseth, E. (2012). *Converting performance based regulations into computable rules in BIM based model checking software*. In: *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2012*. Oslo: Gudni Gudnason, Raimar Scherer, pp. 461-469.
- Hjelseth, E. and Nisbet, N. (2010). Overview of concepts for model checking. In: *CIB W78 2010: 27th International Conference*, Cairo. v. 78, p. 2010.
- Hjelseth, E. and Nisbet, N. (2011). Capturing normative constraints by use of the semantic mark-up RASE methodology. In: *CIB W78-W102 2011: International Conference*, Sophia Antipolis. pp. 1-10.
- Khemlani, L. (2005 *apud* Mainardi Neto, A. I., 2016). *CORENET e-PlanCheck: Singapore's automated code checking system*.
- Mainardi Neto, A. I. d. B. (2016). *Verificação de regras para aprovação de projetos de arquitetura em BIM para estações de metrô*. Masters Diss. Escola Politécnica da Universidade de São Paulo.
- Manzione, L. (2019). *Desenvolvimento de um sistema semântico interoperável com base em BIM para a verificação automática de conformidade com o "Regulamento de segurança contra incêndio das edificações e áreas de risco do Estado de São Paulo"*. Postdoctoral research project. Faculdade de Arquitetura e Urbanismo da Universidade de São Paulo.
- Rodrigues, J. P. P. (2015). *Verificação automática de projetos: Plano de Acessibilidades*. Masters Diss. Faculdade de Engenharia da Universidade do Porto.
- Solihin, W. (2015). *A simplified BIM data representation using a relational database schema for an efficient rule checking system and its associated rule checking language*. Ph.D. Thesis. Georgia Institute of Technology.