41st IAHS WORLD CONGRESS
Sustainability and Innovation for the Future
13-16th September 2016
Albufeira, Algarve, Portugal

AUTOMATED RULE-CHECKING – A TOOL FOR DESIGN DEVELOPMENT

João Poças Martins[1]*, Bárbara Rangel[1] and Vitor Abrantes[1]

1: Department of Civil Engineering
Faculdade de Engenharia
Universidade do Porto
Rua Dr. Roberto Frias, Porto, Portugal
e-mail: { jppm, brangel, abrantes }@fe.up.pt, web: http://fe.up.pt

Keywords: BIM, Automated, Rule, Checking, Design

Abstract *Although automated rule-checking is often presented in the context of applying for licenses and permits, as a way to assess compliance with building codes, its most immediate application can be found in earlier stages of the design phase. The impact of design options on the performance of buildings in several domains can be assessed automatically using BIM tools. In this article, we discuss the advantages and challenges of adopting automated rule-checking procedure as a tool for design development and present examples of software solutions that can be used for this purpose.*

1. INTRODUCTION

Automated rule-checking is considered a relevant topic for BIM researchers, with significant expected development over the next decade [1, 2]. Indeed, the last five years have shown a growing interest in this field [3], particularly concerning the automation of checking building design models for compliance with building codes and planning restrictions. Although these procedures are often viewed as a part of a process to obtain a building permit, they should also be understood as a powerful tool which is available to designers, from the earliest stages of design development.

Although there are still critical obstacles to its effective implementation, automated rule-checking of design models should be an increasingly integrated feature in BIM tools that are available to construction professionals.

2. RULE-CHECKING AND BUILDING PERMITS

Automatic code-checking of design models has been a field of research within BIM since the early years of this technology. A significant number of previous initiatives in this field are described in the literature [4]. The Corenet platform from Singapore is a notable example, that allows building projects to be submitted and checked automatically for compliance with building codes [5].

The submission of a building design to an authority having jurisdiction and its assessment in order to eventually grant a building permit is an important milestone when considering the flow of information throughout the construction lifecycle. Indeed, this is often the moment when a first formal information delivery occurs from within the group formed by the actors in the early stages of the construction process (promoters, designers, etc.) [2]. Until this stage, the majority of documents are exchanged among these early-stage participants, with a relatively low level of development and a small degree of formalism. The existence of formal rules of representation that apply to design documents at this stage reduces the variability of the content assessed during the design-review process. The high degree of formalism which is imposed during this latter stage of the design phase makes the code-checking process particularly suitable for automation [6]. The fact that this process is mandatory and concentrated in a relatively small group of entities presents an opportunity for the introduction of formal BIM requirements. As a more immediate set of advantages, automated code-checking allows speedier, dematerialized and more transparent review processes. Thus, by providing immediate benefits to stakeholders who adopt BIM to develop and submit design models for review, public authorities will simultaneously specify delivery formats for BIM models which can be adopted beyond the design-review stage, creating a *de facto*, if not a *de jure*, standard for information exchange.

This "technology-push" approach towards BIM dissemination presents obvious challenges, in particular in countries where the BIM maturity level is considered low. The EU Procurement Directive 2014/24/EU [7] addresses this issue by stating that although "For public works contracts and design contests, Member States may require the use of specific electronic tools, such as of building information electronic modelling tools or similar", they should "offer alternative means of access [...] until such time as those tools become generally available".

Besides these maturity-level related issues, other relevant obstacles prevent the short-term adoption of fully automated code-checking procedures during the design-review stage. Indeed, a successful process depends on three essential pillars, which are currently still insufficiently developed for this purpose: (1) regulations, (2) technology and (3) models, as addressed in the following paragraphs:

1. As a general rule, existing building regulations have not been developed to support automated code-checking procedures. Thus, any such process must be preceded by an assessment of the relevant regulations. Rules may be divided into the following groups: (i) those that are verifiable as is, (ii) those that cannot be verified objectively because they rely on qualitative parameters (such as "close to", "easily accessible", etc.) and (iii) those that are not logical propositions, so their validation is not possible or not relevant (as is the case of provisions where concepts are defined). Even if a rule is classified as checkable, it may not be feasible to do so if the task requires a model with an unusually high Level of Development (LOD). Figure 1 shows the increase in the modelling effort which results from an increased LOD. Hence, although it may be technically possible to validate a particular legislative provision automatically, this may result in the replacement of one problem – a problem for the entity that must perform the design review - with another, potentially more significant one - for the designer - so the net overall impact of automating the process will be negative in this case. Examples of rule interpretation processes that have been carried out prior to the development of automated code-checking procedures can be found in the literature for different domains, including domestic water networks [8, 9], fire safety [10] and accessibility design [11].

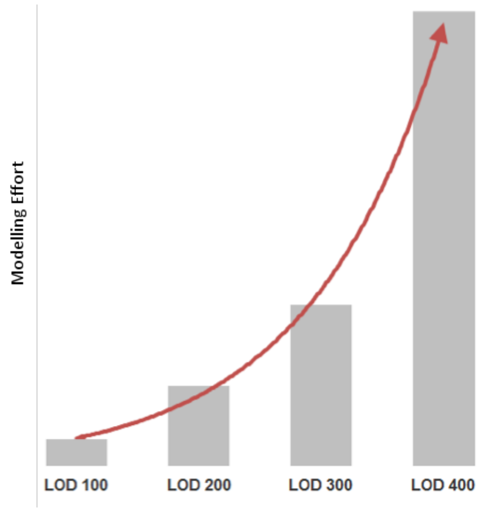

Figure 1. Relationship between a model's LOD and the modelling effort required to develop it [12]

2. The software and hardware must be compatible with the verification operation to be performed. Existing computers can process current building models - usually LOD 300 or less – although the validation of extremely large or detailed models can result in delays or other issues. On the other hand, existing software applications have scope limitations, either because they are only applicable to a specific domain [13], or because they do not allow for full customisation of rules to be checked (as is the case of Solibri Model Checker [11] or ePlanCheck, for example).
3. BIM models must be developed according to the requirements of the code-checking procedures that will be performed. Besides compliance with general modelling requirements (such as satisfying minimum LOD specifications), it is important to ensure the semantic and syntactic validity of the model [14] to guarantee that all entities and relationships have the intended meaning. Non-compliance with semantic, syntactic or general modelling rules may result in a misinterpretation by the code-checking software, even when these issues are not visually detectable in the modelling application. Obtaining valid models depends on the establishment of BIM modelling rules, which are currently under development around the world (in Europe, BIM standards are currently being developed by CEN/TC 442).

3. AUTOMATED RULE-CHECKING DURING THE DESIGN PHASE

Besides the design-review process which follows an application for a building permit, automated rule-checking routines can contribute to the development of building designs by validating design options, checking the impact of these decisions on regulation compliance or on building performance. The rule-checking software can thus be used as an expert system that supports the decisions of designers, assessing the impact of design options.

Unlike the code-checking procedures undertaken when applying for permits, this kind of rule-checking that is performed by the designer has no immediate legal consequences. It is a purely informational procedure that can be performed throughout the development of the design, particularly in its early stages. Partial verification processes (that apply to only some of building code provisions – those that are deemed verifiable, as specified in the previous chapter) are, therefore, admissible in this context.

Besides compliance with building codes, it is possible, for example, to automate design quality assessment procedures [11].

Since automated rule-checking is not a common feature in current BIM modelling applications, dedicated software applications are used for this purpose, such as Solibri Model Checker. The combined use of different modelling and rule-checking applications introduces a set of challenges and limitations:

1. One first challenge is the issue of interoperability. Interoperability has always been a main keyword in BIM research and application because BIM methods are not confined to an individual adoption of a single software tool, or software family. It

is, therefore, necessary to establish a set of standard formats that support the exchange of information between applications. It is possible, for example, to use the IFC format *(Industry Foundation Classes)* to exchange models between modelling and automated code-checking applications or the BCF format *(BIM Collaboration Format)* to support the exchange of messages with relevant BIM context between users of each of these applications. These formats remain in continuous evolution, and they are not fully compatible with native models of each application modelling – lossless information exchange cannot be achieved through naïve model export-import processes.

2. Rule-checking applications must be customised to adapt to new regulations or other rule sets. This requires a considerable degree of flexibility which is not available in existing applications. This means that it is difficult to apply existing software beyond the scope defined by the regulations and other rule sets that have been developed originally. Solibri Model Checker, for example, contains dozens of rule sets for different building codes that are applicable in different regions. The rules are parametric and may be modified taking into account the provisions to be checked. New rules may thus be derived from existing rule sets. However, rules cannot be freely created since there is no open access to the API *(Application Program Interface)*, and it is not always possible to define the desired conditions. Solibri Model Checker users may, therefore, combine existing rule sets and configure individual rule parameters themselves, but since these rule sets are hard-coded into the software [15], new rules must be added by Solibri's own developers whenever needed [16].

The development of domain-specific plugins that can be accessed from the modelling software is an alternative solution [17] that circumvents the challenges presented above. This kind of feature is already partially available in some BIM design applications for structures, MEP (Mechanical, electrical, and plumbing) and infrastructure [18]. Since this allows designers to check models continuously as they are being developed, the impact of design decisions can be assessed in real-time, considering restrictions which are applicable in each project [1]. Naturally, this would require different plugins to be developed for each domain and for each modelling platform, which would result in further challenges for developers.

Several popular BIM modelling applications allow plugins to be developed using Visual Programming Languages (VPL) besides more traditional languages such as C# or Python. This allows the development of plugins for BIM modelling applications without the need to resort to more complex APIs. AEC professionals with little or no coding skills may, therefore, develop code-checking routines according to their needs. As an example, Figure 2 illustrates the development of a Revit plugin named StormWater Runoff using Dynamo, that calculates the rainfall runoff for a building given a BIM model and further input from the user in order to assess sustainability performance [19].

This parametric approach towards design, where formal rules and relationships between entities influence the development of complex building models is currently supported by major software developers such as Autodesk (Dynamo), Bentley (Generative Components) or Rhinoceros 3D (Grasshopper).

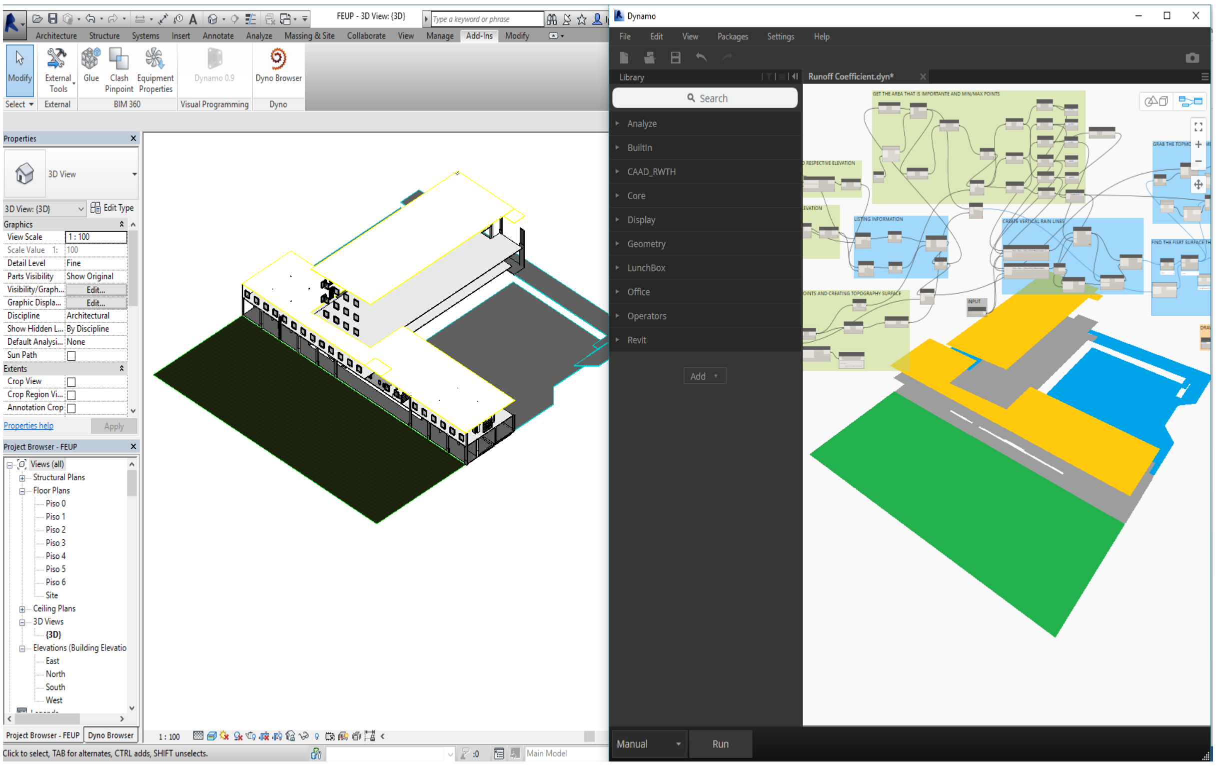

Figure 2. StormWater Runoff: the software prototype in development, showing Revit and Dynamo side by side [19]

4. CONCLUSIONS

Automated rule-checking is not restricted to the design-review phase that is required to obtain a building permit. Despite current technological limitations, it is expected that the coming years will witness the integration of rule-checking features in BIM modelling software. This development will assist designers, providing them with real-time assessments of the impact of their decisions on building performance and compliance with applicable regulations.

Thus, automated rule-checking applications should not be regarded as merely agents of a transition towards digital administrative processes. Indeed, these applications should be seen as expert systems that can assist designers, providing them with a multidisciplinary, informed and detailed view on the consequences of their design options.

REFERENCES

[1] C. M. Eastman, P. Teicholz, R. Sacks, and K. Liston, BIM handbook : a guide to building information modeling for owners, managers, designers, engineers and contractors, 2nd ed. Hoboken, NJ: Wiley, 2011.

[2] J. P. Poças Martins and V. Abrantes, "Automated code-checking as a driver of BIM adoption," *Int. Journal for Housing Science and its Applications,* vol. 34, pp. 287-295, 2010

[3] M. Yalcinkaya and V. Singh, "Patterns and trends in Building Information Modeling (BIM) research: A Latent Semantic Analysis," *Automation in Construction,* vol. 59, pp. 68-80, 11// 2015. doi: http://dx.doi.org/10.1016/j.autcon.2015.07.012.

[4] C. Eastman, J.-m. Lee, Y.-s. Jeong, and J.-k. Lee, "Automatic rule-based checking of building designs," *Automation in Construction,* vol. 18, pp. 1011-1033, 2009. doi: DOI: 10.1016/j.autcon.2009.07.002.

[5] buildingSMART, "The CORENET project in Singapore," *buildingSMART Case Studies,* 2006

[6] J. P. Poças Martins, "Modelação do Fluxo de Informação no Processo de Construção - Aplicação ao Licenciamento Automático de Projectos," PhD Thesis, Faculdade de Engenharia, Universidade do Porto, Porto, 2009.

[7] Directive 2014/24/UE of the European Parliament and of the Council of 26 February 2014 on Public Procurement and repealing Directive 2004/18/EC, O. J. o. t. E. Union Article 22, Paragraph 4, 2014.

[8] J. P. Poças Martins, "Modelação do Fluxo de Informação no Processo de Construção, Aplicação ao Licenciamento Automático de Projectos," Doutoramento Dissertação para obtenção do grau de Doutor em Engenharia Civil, Faculdade de Engenharia da Universidade do Porto, Porto, 2009.

[9] A. Monteiro, "Avaliação da aplicabilidade do modelo IFC ao licenciamento automático de projectos de redes de distribuição predial de água," Tese de Mestrado, Faculdade de Engenharia, Universidade do Porto, 2010.

[10] Y. Li, "Automated Code-checking of BIM models," Master in European Construction Engineering Master, University of Cantabria, Santander, 2015.

[11] R. Moço, "Verificação Automática de Modelos com Recurso a Ferramentas BIM - Aplicação à Avaliação de Qualidade de Projetos de Edifícios de Habitação," Tese de Mestrado, Faculdade de Engenharia, Universidade do Porto, 2015.

[12] J. S. Silva, "Princípios para o Desenvolvimento de Projetos com Recurso a Ferramentas BIM," Tese de Mestrado, Faculdade de Engenharia, Universidade do Porto, 2013.

[13] J. P. Poças Martins and A. Monteiro, "LicA: a BIM based automated code-checking application for water distribution systems," *Automation in Construction,* vol. 29, pp. 12-23,

January 2013. doi: 10.1016/j.autcon.2012.08.008.

[14] Y.-C. Lee, C. M. Eastman, and J.-K. Lee, "Validations for ensuring the interoperability of data exchange of a building information model," *Automation in Construction,* vol. 58, pp. 176-195, 2015

[15] H. v. Bell, L. Bjørkhaug, and E. Hjelseth, "Standardized Computable Rules," Standards Norway, Strandveien 18 P.O.Box 242 NO-1326 Lysaker Norway2009.

[16] H. Lee, J.-K. Lee, S. Park, and I. Kim, "Translating building legislation into a computer-executable format for evaluating building permit requirements," *Automation in Construction,* 2016. doi: http://dx.doi.org/10.1016/j.autcon.2016.04.008.

[17] J. P. Poças Martins. (2015) Verificação Automática de Projectos. Para Além da Desmaterializaão de Processos. *Construção Magazine*. 32-34.

[18] L. Khemlani. (2015) Automating Code Compliance in AEC. *AECbytes*. 3-6.

[19] L. Sanhudo, "BIM assessment for Building Sustainability and automatic Rainwater Runoff calculation," MSc Thesis, Faculdade de Engenharia, Universidade do Porto, 2016.