Pre-Processing IFC Building Models for Code Compliance Checking based on Visual Programming

Cornelius Preidel[1], Nicolas Bus[2], André Borrmann[3], Bruno Fies[4]

1) Ph.D. Candidate, Chair of Computational Modeling and Simulation, Technical University of Munich, Munich, Germany. Email: cornelius.preidel@tum.de
2) IT Senior Project Manager., Centre Scientifique et Technique du Bâtiment, Sophia Antipolis, France. Email: nicolas.bus@cstb.fr
3) Full Professor, Chair of Computational Modeling and Simulation, Technical University of Munich, Munich, Germany. Email: andre.borrmann@tum.de
4) R&D Engineer, Centre Scientifique et Technique du Bâtiment, Sophia Antipolis, France. Email: bruno.fies@cstb.fr

Abstract:
The use of building information models for automatically checking building designs for compliance regarding codes and regulations promises a significant reduction of today's laborious and error-prone manual processes. Therefore, the digitalization of regulations forms a key part in many national or international efforts, such as the French Plan de Transition Numérique du Bâtiment. However, the challenges involved are manifold. On the one hand side, the existing rules and regulations have been written for interpretation by humans and do in most cases not allow for a direct translation into machine-readable representation. At the other hand, Building Information Models require a careful preparation to provide information in a manner that is usable for code compliance checking procedures. In this regard, a research project is conducted by the CSTB, which aims at proving the general feasibility of digital rule checking and developing a methodology which works under real-world practical conditions. The methodology allows to capture the domain's knowledge concerning the interpretation of the regulations and to transfer it into a representation that is processable by computers. It is based on the application of a semi-formal language and the identification of key terminology. The domain expert groups analyze the existing regulation texts and translate it into semi-formal rule definitions consisting of if-then clauses. The project is based on a strong involvement of experts from the different domains and the regulation-setting committees. In many cases key concepts identified in the regulatory texts do not have a direct counterpart in the IFC data schema. Accordingly, a pre-processing step is required to process the original IFC building model and derive or compute the required information and add it to the IFC model. Available tools for perform preprocessing steps as part of the code compliance have major drawbacks: (1) the computing code is hidden and cannot be validated by domain experts, and (2) they are inflexible with respect to adapting the procedures to national requirements or adding new ones. For this reason, the authors propose the application of a visual programming language for defining IFC pre-processing procedures. The language allows domain experts to define the procedures on their own and to check the outcomes of individual processing steps in an interactive way. The paper discusses the features and the syntax of the language in detail and illustrates its application by way of practical examples.

Keywords: Building Information Modeling, Code Compliance Checking, Visual Programming

1. INTRODUCTION

According to a study by McGrawHill Construction (2007) architects, engineers, contractors, construction managers as well as building owners have a considerable amount of effort with the design review. As a result, nearly half of architects and owners spend more than 26 hours on code checking during a typical project - in some cases even more than 100 hours. For this reason, there is a great interest in finding a solution to automate the checking process, especially for architects and engineers, who must deal with this process daily. The establishment of digital methods in construction industry, such as the Building Information Modeling (BIM), which is based on the use of digital building models enriched with geometric and semantic information, enables an advance of this automation, called Automated Code Compliance Checking (ACCC). The digital building models, used in a BIM-based project, represent in their entirety the holistic design planning and contain various information for all individual building phases, which must be checked for conformity with various requirements. Next to basic design requirements and functional descriptions, these include the applicable guidelines and standards of the building industry, which ensure technical standards and thus, among other things, the quality and safety of the building. Based on an analysis of the priority and frequency of used types of references for design validation, Uhm *et al.* (2015) state that there are diverse sources from which the planning engineers derive their regulation knowledge. In addition to typical sources of rule knowledge, such as literature material, standards and guidelines, these sources include specific requirements regarding the functionality of the building as well as empirical knowledge. From this it can be deduced that the experience and knowledge of users plays an essential role.

However, to automate the compliance checking, the first step is to formalize and digitize the regulation contents, so that these are understandable for machines (Eastman *et al.*, 2009). This step involves several important challenges: On the one hand side, the existing rules and regulations have been written for interpretation by humans and do in most cases not allow for a direct translation into machine-readable representation. At the other hand, Building Information Models require a careful preparation to provide information in a manner that is usable for code compliance checking procedures.

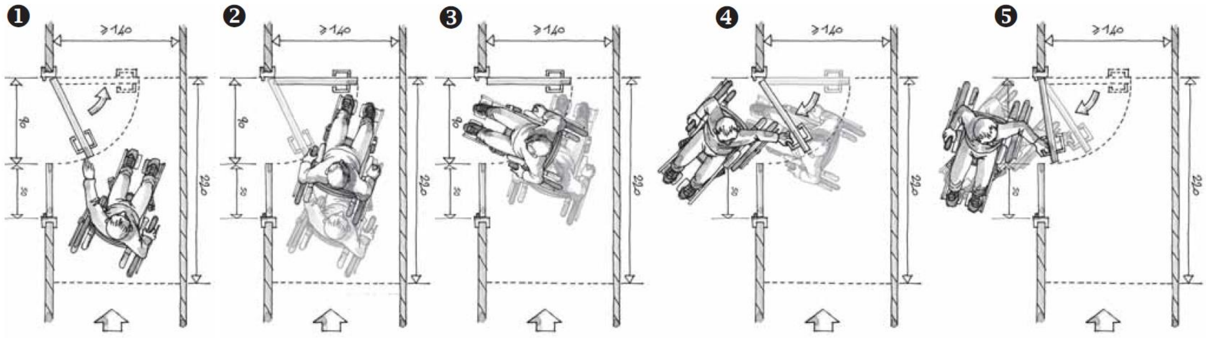

Figure 1. Extract from the French standard Annexe 8: requirements for the operating space in front of a door in a corridor situation so that the room is accessible for wheelchairs (Ministere du logement et de la ville, 2008)

When reviewing the design planning, for various demanding checks specific information is required which is not contained in the digital building model or even not foreseen in standardized data schemata of the digital construction industry, such as the Industry Foundation Classes (IFC). An example of such missing information is the computation of operating spaces, which are required for accessibility checks, as these are exemplarily shown in Figure 1. The operating spaces (blue) are not to be confused with the door swing range (green), as they describe the minimal required space in front of as well as behind doors so that there is sufficient space for wheelchair users to maneuver while opening and closing the door. In principle, the operating spaces are directly dependent on the corresponding door, since the required dimensions result from the position and type of the door. However, usually, the building model does not contain this kind of information. Some of the nowadays commercial checking environments cover such accessibility checks, but intermediate process steps (i.e., the creation of the operating spaces and the subsequent check) is hidden and bound to specific international or national guidelines/requirements. Consequently, the user only receives only the results of such a check but cannot determine whether the actual requirement was correctly implemented and executed by the software. A second example is the subsequent identification of the general accessibility of a building. Based on the determination of accessible doors, an accessibility graph (see Figure 1) can be created, which is used to determine accessible routes through the building. This graph, which represents all possible routes through the building, is typically not included in a building model.

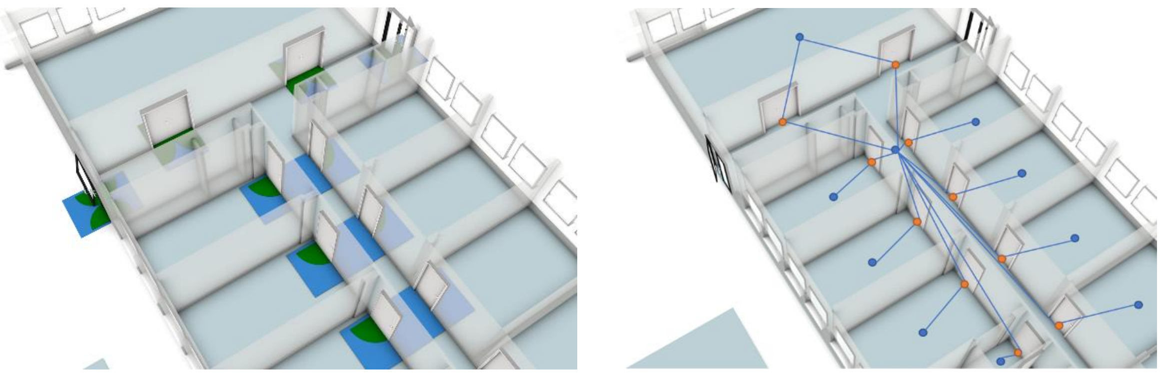

Figure 2. Representation of derived information sets that are typically not included in a building model. Left: Operating Spaces, Right: Accessibility Graph

2. SELECTED COMMERCIAL SOLUTIONS AND PREVIOUS STUDIES

In ACCC area, a distinction has to be made between existing, commercial software solutions, which usually focus a basic model checking with additional code checking features, and research approaches, that pursue a more holistic, advanced building code checking.

One of the major commercial software solutions in the model and building code checking is the Solibri Model

Checker (SMC) by the Finnish company Solibri Inc. (2017). The SMC key feature is a rule assembly editor, with which the user can build up an individual checking and validation process from a library of sophisticated rules. These rules can be adjusted by selected, given parameters, but the content and the process of the check is not visible to the user. This means that the rules can only be adapted to users' needs to an insufficient extent.

The CORENET system of the Building Construction Authority in Singapore is a central digital platform for digitized building regulations and e-submission. An essential part of this platform is the e-PlanCheck system, which serves as a checking tool for the digital building models with building codes in the areas of building control, barrier-free access and fire safety (Solihin, 2004). The main component of this checking software is a commercial hard-coded implementation, also known as FORNAX library, which is maintained and developed by a private company. The individual processing steps of the checking process are not visible to the user so that there is no or just limited way for the users to alter it. Therefore, this approach is also considered as a "black box" implementation, which domain experts do usually not trust Nisbet *et al.* (2008).

The SMARTCodes project was led by the International Code Council (ICC) as a significant research approach. A result of this project is a two-phase process, in which domain experts enrich the regulation texts written in human language with semantic markup first. Afterwards, the enriched text is semi-automatically transferred into a computer-processable form (Nisbet *et al.*, 2008; Hjelseth and Nisbet, 2011; Hjelseth, 2012). The basis for the first step is the application of the RASE syntax, a derived form of the well-known and widely used data modeling standard Extended Markup Language (XML), which results in an easily verifiable and maintainable code representation. With the help of RASE, the text contents of a guideline are manually categorized into the four different classes: Requirement, Applicability, Selection, and Exception. A significant advantage of this method is, that even complex contents of codes and guidelines may be formalized into these essential components and that the formalized text can be read afterward by both, human and machine. A significant limitation of this approach is that only the content of a standard can be covered, but not expert knowledge, which results from the experience of the respective reviewer. Furthermore, the approach does not cover procedural knowledge; so how objects and information to which the guideline refers to, is produced to check the rule.

Solihin and Eastman (2015) break down the structure of rules into atomic elements and represent these as conceptual graphs (CG). They introduce this representation as a suitable method due to its expressiveness and conformity with first-order logic, which allows unambiguous description of the requirements that can be understood by participants in implementation efforts. Therefore, this approach is primarily used to achieve a uniform interpretation of the regulations and to implement them subsequently by software experts.

Uhm *et al.* (2015) use context-free grammar (CFG) for processing the rules defined in the natural language in a request for proposals for large buildings in South Korea. As a result, they identify essential objects and methods, as standard components of these requests, which can then be re-used in the translation and thus achieve a high coverage rate. Moreover, the translated rules are translated with the help of the computer-interpretable Semantic Web Rule Language (SWRL) rules.

Park and Lee (2016) introduce KBimCode, a standardized scripting language for the representation, definition, and evaluation of building permit rules as a part of the Korean KBim project. Software specialists can use this scripting language to translate natural language rules with a rule-based logic mechanism. Based on initial analyses of the Korean Building Act, a database has emerged as an essential result of this approach, in which specific objects and properties are stored and classified, which in turn can be used for further translations (Kim *et al.*, 2017).

As a part of the French Plan de Transition Numérique du Bâtiment (Ministère du Logement, de l'Égalité des territoires et de la Ruralité, 2017), a research project is conducted by the Centre Scientifique et Technique du Bâtiment (CSTB) which aims at proving the general feasibility of digital rule checking and developing a methodology which works under real-world practical conditions. For the initial feasibility study, five domains have been selected: Fire Safety, Accessibility, Acoustic insulation, the upcoming French thermal/sustainable regulation and the recommendation for the implementation of Plasterboards. The project is based on an active involvement of experts from the different domains and the regulation-setting committees. The methodology developed at CSTB allows to capture the domain's knowledge concerning the interpretation of the regulations and to transfer it into a representation that is processable by computers. It is based on the application of a semi-formal language and the identification of key terminology. The domain expert groups analyze the existing regulation texts and translate it into semi-formal rule definitions consisting of if-then clauses.

3. METHOD

As described in the last sections, that many approaches deal intensively with the translation of rule knowledge, but there is a lack of methods to derive necessary knowledge and information from building models for the application of sophisticated checks. A full-automated black-box derivation of such information is not recommended, as the adaptability is very limited, and the user is interested in the temporarily generated information e. g. for plausibility checks. Consequently, this process should be at least transparent in an intermediate step and adaptable with the help of parameters, so that the user can influence it and the coverage rate of the approach is increased. To tackle this problem, we propose a pre-processing system which describes the methods for the derivation of missing information transparently. The approach is primarily based on three components: the identification of information that needs to be derived, an MVD-based IFC model checking and a visual programming language for a precise and adjustable description of the information derivation processes. The individual components, as well as their composition, are described in the following sections.

3.1 Semi-Formal Language for Identification of Key Terminology

In the frame of the French feasibility study about digitalization/formalization of Construction regulation the process adopted can be decomposed into two main steps.

The first step is focusing on the production of semi-formal assertions. Domain experts with the support of a toolbox were in charge of re-writing the regulation constraints expressed in natural language into a semi-formal representation following a few set of recommendations. In order to support this production process, a toolbox has been used. This toolbox was mainly a text editor with a dedicated syntax highlighting and auto-completion capabilities. The highlighting was based on the identification and collection of a controlled construction oriented vocabulary organized into an ontology derived from the IFC4 ontology. Above the IFC4 backbone, French terms chosen to ensure as much as possible the link with the openBIM, have been added coming from existing thesauri or on expert basis. Along with this definition of a BIM/French vocabulary, a simple grammar has been decided in order to guide the re-writing of the regulation constraints expressed in natural language into a "semi-formal" representation. This approach was inspired from existing work and standards (Hjelseth and Nisbet, 2011; Object Management Group, 2017). This step is totally independent from any technical solution.

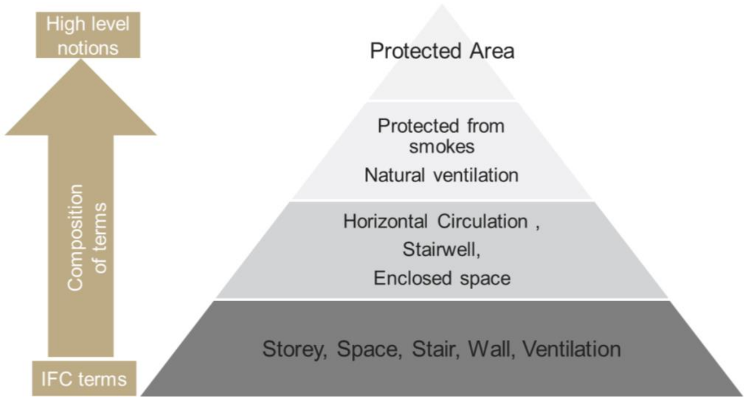

Figure 3. Composition of notions - the approach chosen to enrich the openBIM vocabulary with high level notions manipulated by regulatory documents.

The second step in the transformation (or formalization) process is the translation of the semi-formal rules into formal rules. This step is technology dependent and has required as a prerequisite a good knowledge about the capabilities and limitations of the selected technologies. Following the construction standards (buildingSMART), CSTB was used to use mvdXML technologies and related tools. This approach has several assets, it is already used and well known by our developers and it is used to check ifc files compliance with the Building Information Modeling Execution Planning. But the limitations of such an approach are also known (Fahad *et al.*, 2016). In mvdXML the presence or absence of an entity or a property can be easily tested as well as the notion of cardinality. But when it comes to the extraction and evaluation of the value of a property in order to perform extra analysis depending on this value (conditional branching) things are becoming complicated. Another limit of this approach is that we are limited to the expressiveness of IFC. It means we are not allowed to manipulate notions or concepts not present natively in IFC. This is where the second approach based on the Semantic Web technologies offers solutions to overcome the above mentioned limitations. For instance in the French Fire Safety regulation, there is a notion of "Protected Area" which as a specific meaning and aggregates several entities and specific properties (mixing zones, rooms, openings, walls and ceiling coverings, fire rating values, etc…). Such a notion has no direct (and easy to manipulate) equivalence in IFC and the use of semantic technologies appears as an enabler to manipulate such high level vocabulary/ notion closer to the content of the regulatory texts.

It is worth mentioning that as soon as new high level terms are defined, they are populating the controlled vocabulary mentioned and used in the first step.

3.2 Visual Code Compliance Checking

The Visual Code Checking Language (VCCL) is a domain-specific, strongly typed and object-oriented programming language specifically designed for the formulation of checking and verification routines (Preidel and Borrmann, 2016). The VCCL is not only intended to store the contents, which are represented by codes or guidelines, but also to cover procedural knowledge. This means, that the VCCL describes, how information must be processed to describe content, which is a part of a guideline but not of the digital building model to be tested.

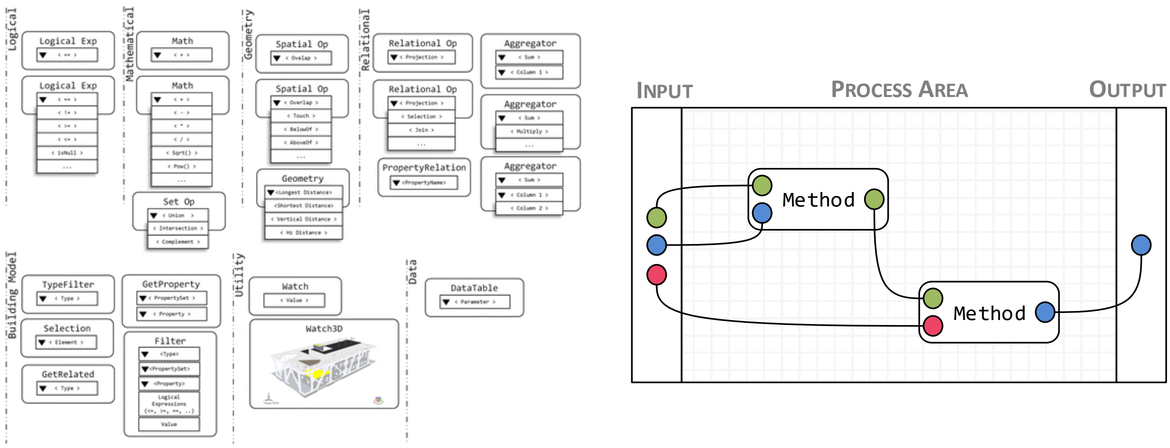

Figure 4. Left: Selected atomic methods of the VCCL, Right: Generic VCCL application environment

For the description of information processes, the VCCL provides essential elements: methods (rectangularly shaped nodes) representing well-defined operations, ports (input- and output circles forming part of the method shapes) representing the incoming and outgoing data type and directed edges, which connect the methods by linking the ports (see Figure 4). To cover procedural knowledge, the VCCL provides functionalities for extracting specific information from a building model similar to query languages. As a starting point, basic VCCL methods are provided for the user, which describe fundamental operations, whose semantics are unambiguously well defined. These methods are called atomic methods as their implementation is hidden from the user and realized by standard programming languages (see Figure 4). Atomic methods include, among others mathematical operators, geometric operators, Relational Algebra operators, and access operators for extracting information from building models. These methods represent a black-box level, from which we assume that it is acceptable for a given application area, where deeper control and insight is neither desired nor necessary, and too much effort would be created for implementing the provided functionality using VCCL instead of a standard programming language. For further information, i.e., application cases, the reader is referred to (Preidel and Borrmann, 2016).

3.3 Preprocessing of IFC Building Models for Code Compliance Checking

With the tools, as described in the previous sections, a preprocessing process for IFC models is composed as shown in Figure 5. The major purpose of this system is to derive information from the building model, which can afterward be used to verify the design planning.

The main input of the preprocessing system is an IFC model, which contains the essential architectural information representing the design planning. To meet general quality requirements, the model must be prepared in a preliminary step. Depending on the case of application, these requirements include, e.g., the compliance with specific modeling guidelines or complete attribute sets. For geometry-related checks such as accessibility regulations, a consistent geometric representation (i.e., explicit vs. implicit representation) is especially important. On the other hand, the actual rule check and thus the essence of a selected rule is extracted in a manual step by determining parameters, which enter the processing system as input. In the VCCL system, the input information comes together and is processed to information, which can afterward be merged into with the original IFC model. The prerequisite for this is that the VCCL can describe all required components and methods. With the help of the VCCL, the resulting processed information is added as new IFC objects or attached as additional attributes to existing objects. In a final step, the resulting model can be tested with the help of MVD specifications.

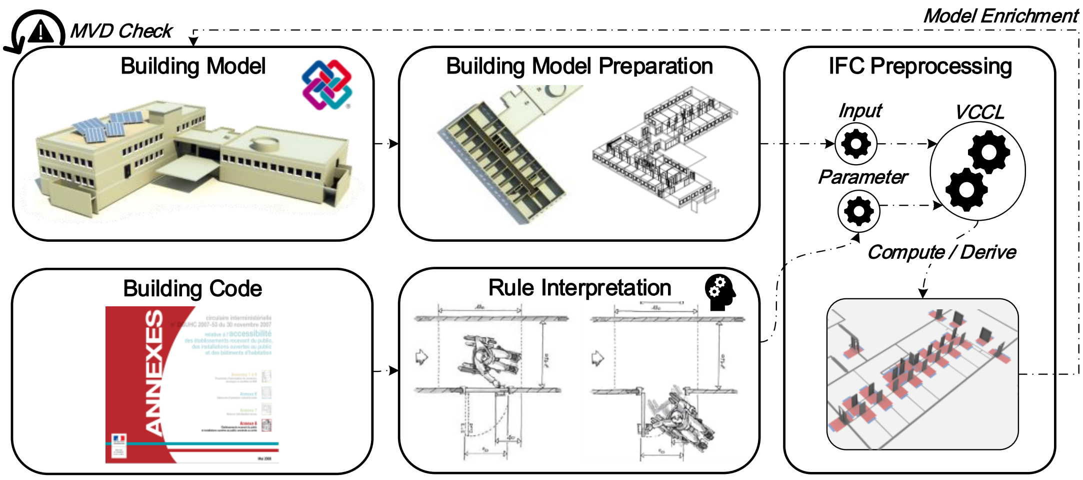

Figure 5. Schematic workflow of the IFC preprocessing

4. RESULTS

To proof our proposed workflow, we implemented the check of accessibility by identification of valid operating spaces. Since the operating spaces are not included in a building model, they must be derived. According to the selected French regulation Annexe 8 different possibilities how wheelchair users can enter a room are valid. The room to maneuver, which is described by a certain width and length, can be located either directly in front of or beside the door so that there is enough free space. The space situation also differs from which side the wheelchair user approaches and in which direction the door opens. From these circumstances, it can be deduced that a combination of spaces in front of and behind the door must be given so that the door can be considered as accessible and handicap free. The space combinations, as well as the dimension, have to be manually extracted as shown in Figure 6.

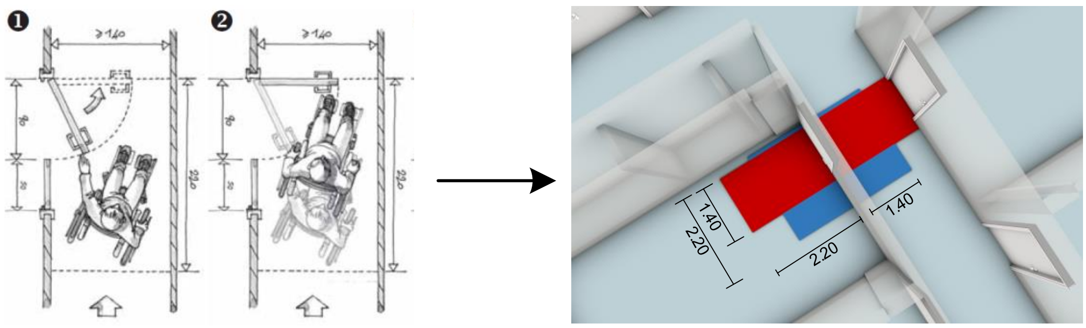

Figure 6. Operational space combinations of a door. At least one space (blue or red) in front of or behind the door must be free so that the door is considered as handicap free (Ministere du logement et de la ville, 2008).

As there is no space checking feature in the VCCL so far, a new atomic method "Create Operational Spaces" has been introduced, which can create all operational space combinations of moving building elements, such as doors or windows. This operator can adjust the dimensions of the spaces according to respective parameters. From the given requirements results in the full VCCL system as represented in Figure 7.

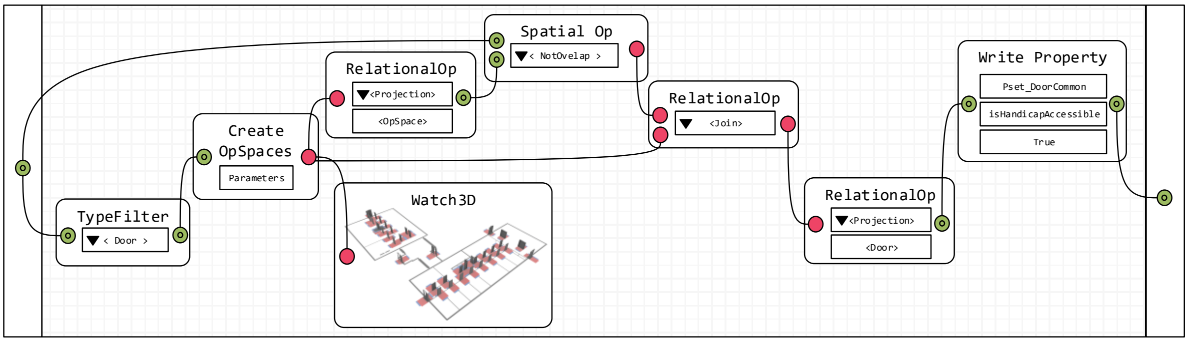

Figure 7. VCCL System for the generation of the handicap information

After creating the valid operational space combinations, in the VCCL system, the spaces are checked for overlaps/clashes with existing building elements. If at least a variant is valid for a door, it may be considered as handicap accessible. To store the result of the accessibility check, the according to IFC standard attribute, i.e., the single value *HandicapAccessible* (Boolean), can be attached to all doors and set correspondingly to *True* for the valid and *False* for the invalid elements. As a last step in the VCCL environment, the original IFC model is updated with the new information.

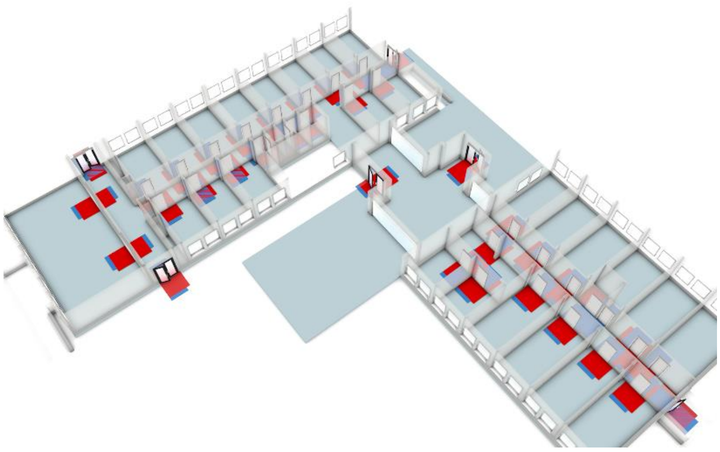

Figure 8. Resulting Operational Spaces of a selected floor in the CSTB test building

The resulting IFC model is checked finally regarding the accessibility of contained doors by an MVD specification, which contains an *HandicapAccessible* attribute definition for all *IfcDoor* instances. In this way, the actual check is described compliant with existing standards.

5. CONCLUSIONS

As shown in our paper, the automation of the design review regarding the various building regulation represents a key challenge to increase efficiency. Many research efforts focus on translating rule knowledge into a formal language but do not consider that for various regulations information must be derived from the models first. With the proposed preprocessing system, we introduce a method with which the processing of required information is described transparently, so that a user may determine specifically which information is to be generated as IFC model content. The actual checking process is carried out using MVD and thus effectively separated from the preparation of the information.

With implementing the IFC preprocessing for the example of operational spaces, we prove that the approach is applicable. As a next step further demanding checks can be implemented with the preprocessing system. A continuation of the presented example is a general accessibility check by determining accessible routes in the building. With the help of the VCCL an accessibility graph can be created, which primarily represents all possible access routes through a building by linking building elements, that connect rooms and storeys as shown in Figure 2. Based on the accessibility information, which was processed in the first example, and a selected route, it is possible to determine if a specific route is accessible as a whole. In this way, a building model can be checked regarding the overall accessibility for wheelchair users.

ACKNOWLEDGMENTS

The authors gratefully acknowledge the support by CSTB, Nemetschek Group as well as ALLPLAN GmbH for the research project presented in this paper.

REFERENCES

Eastman, C., Lee, J.-m., Jeong, Y.-s. and Lee, J.-K. (2009), “Automatic rule-based checking of building designs”, *Automation in Construction*, Vol. 18 No. 8, pp. 1011–1033.

Fahad, M., Bus, N. and Andrieux, F. (2016), “Towards Mapping Certification Rules over BIM”, *Proceesings of the CIB W78 2016, Brisbane, Australia*.

Hjelseth, E. (2012), “Converting performance based regulations into computable rules in BIM based model checking software”, in Gudnason, G. (Ed.), *eWork and eBusiness in Architecture, Engineering and Construction: ECPPM 2012*, CRC Press, Hoboken, pp. 461–469.

Hjelseth, E. and Nisbet, N. (2011), “Capturing normative constraints by use of the semantic mark-up RASE methodology”, *Proceedings of the 28th International Conference of CIB W78*, pp. 26–28.

Kim, J., Lee, J.-K., Shin, J. and Kim, H. (2017), “Classification of Objects and their Properties Defined in Korea Building Act to Translate into KBimCode and their Application”, in Cheng, M.-Y., Chen, H.-M. and Chiu, K.C. (Eds.), *Taipei, Taiwan, 28.06.2017 - 01.07.2017*, International Association for Automation and Robotics in Construction (IAARC).

McGrawHill Construction (2007), “Interoperability in the construction industry. SmartMarket Report, Interoperability Issue”, available at: http://www.1stpricing.com/pdf/MGH_Interoperability_SmartMarket_Report.pdf (accessed 7 November 2017).

Ministere du logement et de la ville (2008), *Annexe 8 : Accessibilité des établissments recevant du public, des installations ouvertes au public et des bâtiments d habitation*.

Ministère du Logement, de l'Égalité des territoires et de la Ruralité (2017), “Plan Transition Numérique dans le Bâtiment”, available at: http://www.batiment-numerique.fr/ (accessed 7 November 2017).

Nisbet, N., Wix, J. and Conover, D. (2008), “The Future of Virtual Construction and Regulation Checking”, in Brandon, P.S. and Kocatürk, T. (Eds.), *Virtual futures for design, construction & procurement*, Blackwell Pub, Oxford, Malden, MA, pp. 241–250.

Object Management Group (2017), *SBVR*, available at: http://www.omg.org/spec/SBVR (accessed 10 January 2018).

Park, S. and Lee, J.-K. (2016), “KBimCode-based Applications for the Representation, Definition and Evaluation of Building Permit Rules”, in *2016 Proceedings of the 33rd ISARC*, Auburn, USA, pp. 720–728.

Preidel, C. and Borrmann, A. (2016), “Towards code compliance checking on the basis of a visual programming language”, in *Journal of Information Technology in Construction*, Special Issue CIB W78 2015 Special track on Compliance Checking, pp. 402–421.

Solibri Inc. (2017), “Solibri Model Checker”, available at: https://www.solibri.com/ (accessed 08.032016).

Solihin, W. (2004), “Lessons learned from experience of code-checking implementation in Singapore”, available at: https://www.researchgate.net/publication/280599027_Lessons_learned_from_experience_of_code-checking_implementation_in_Singapore (accessed 6 October 2016).

Solihin, W. and Eastman, C. (2015), “Classification of rules for automated BIM rule checking development”, *Automation in Construction*, Vol. 53, pp. 69–82.

Uhm, M., Lee, G., Park, Y., Kim, S., Jung, J. and Lee, J.-K. (2015), “Requirements for computational rule checking of requests for proposals (RFPs) for building designs in South Korea”, *Advanced Engineering Informatics*, Vol. 29 No. 3, pp. 602–615.