文献综述

相比较于人工规范检查方式，自动规范检查(Automated compliance checking, ACC)能够减少规范检查所需的时间、金钱和精力，并能够规避在规范检查过程中可能出现的人为因素的错误。自动规范检查这一概念被提出后，相关研究人员与科研机构首先将研究的思路集中于如何将规范条文编码成计算机可执行的规则来实现规范的自动检查。Jungsik Choi【1】等人开发了一个InSightBIM-Evacuation平台用于人员疏散规范设计与检查，该平台的规范自动检查功能是通过分析人员疏散规范条文以明确检查对象与检查属性以及检查对象与检查属性与IFC实体的映射关系（如图1所示），随后将规范条文硬编码成程序逻辑（如图2）。

图1 人员疏散规范检查对象与IFC实体的映射关系图

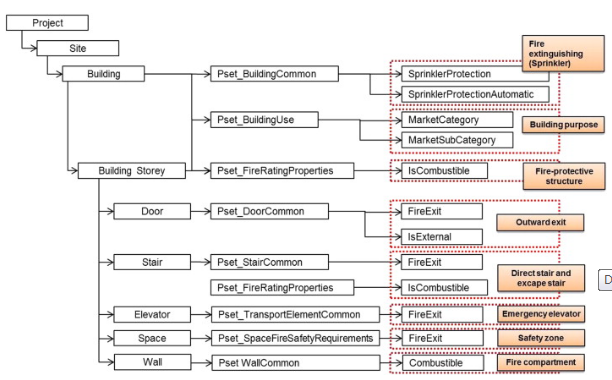
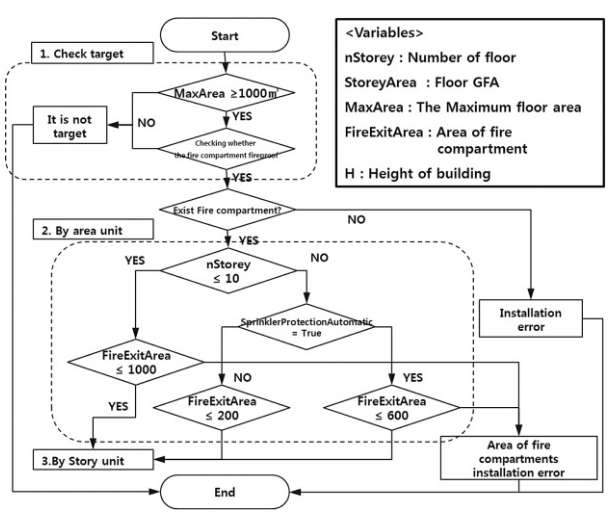
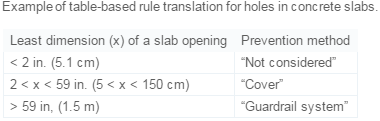


图2 规范条文硬编码逻辑图（示意）



Zhang S【2】等人着眼于施工安全规范检查，并提出可以使用如下四个步骤来实现规范的自动检查，（1）规则翻译阶段，即将规范条文映射成计算机可处理的参数化表格规则（如图3所示）；（2）模型准备阶段，使用多种MVD（Modle view definition）以便提供足够信息用于规则检查；（3）规则执行阶段，对模型实施规则检查；（4）生成报告阶段。

图3 从规范条文到参数化表格规则的映射



除了相关研究人员和机构的研究工作外，目前还有几个基于规则的系统平台或工具正在被广泛使用。其中之一是Solibri Model Checker(SMC)【3】，SMC是一个能够读取并处理IFC模型java桌面应用，其提供了一系列内建函数可供用户进行模型的预检查、基于ISO可达性规范的可达性检查，消防逃生路线距离检查等功能（如图4所示），此外，SMC还提供了API来帮助用户自定义规则来对IFC模型进行检查。Jotne EDModelChecker【4】提供了一个对象数据库并支持使用EXPRESS和EXPRESS-X语言的规则检查，该工具曾被挪威的国家建筑技术与管理办公室使用。新加坡的CORENET系统【5】使用了FORNAX平台，该平台是一个基于C++对象库建立的，其能够从IFC数据中衍生出附加数据并用于规则检查（如图5所示）。

图4 Solibri Model Checker的规则检查功能示意图

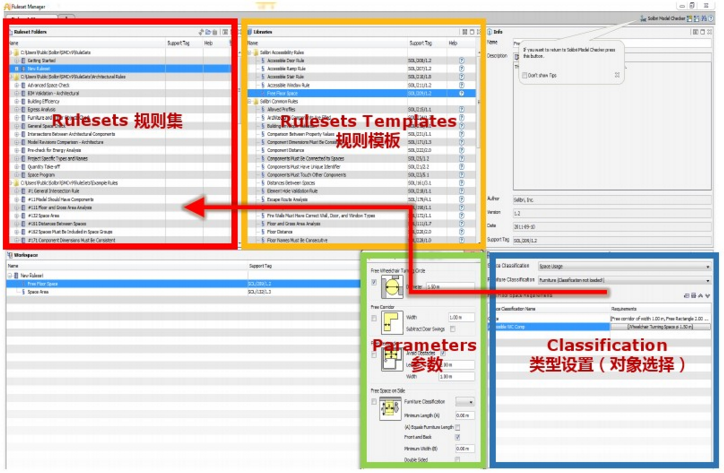
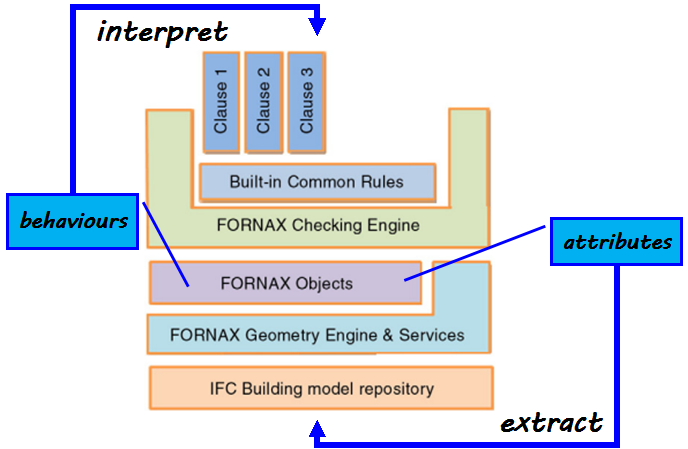


图5 新加坡CORENET系统架构示意图



上述将规范条文进行硬编码成计算机程序语言的方法相比于人工规范检查而言检查效率更高、出错可能性更小。然而，这种对规范条文进行规则硬编码的方式存在如下的不足之处：（1）需要对不同的规范文本进行规则的硬编码；（2）维护和更新硬编码后的规范条文需要大量的劳动和精力。

鉴于对规范条文进行硬编码的方式存在着以上所述的不足之处，部分研究人员开始将思路转向利用语义网技术与本体论来对规范条文进行软编码。

其中，Hjelseth E等人【6】指出为实现规范的自动检查，首先需要解决的问题是如何高效地将文本表示的规范条文捕捉并转换为可用于对模型进行检查的规则，为此，其提出了使用基于必要条件（requirement，R）、使用条件（applicabilities，A）、选择（Selection，S）和例外（exception，E）四个操作符的RASE方法。使用RASE方法首先对规范文本进行词汇标记（如图6所示），随后从文本中抽取相应实体构成检查规则（如图7所示）。但是这种方法较为适合对定量检查的规范条文进行转化，而不适宜于定性检查规范条文的转化。

图6 利用RASE方法对规范条文进行标记示意图

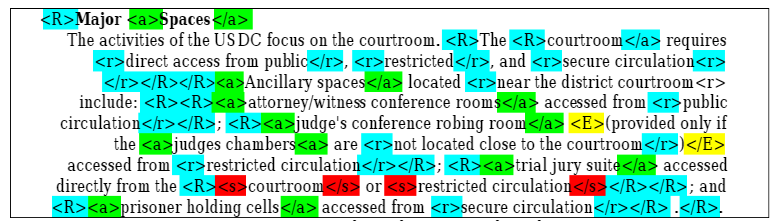
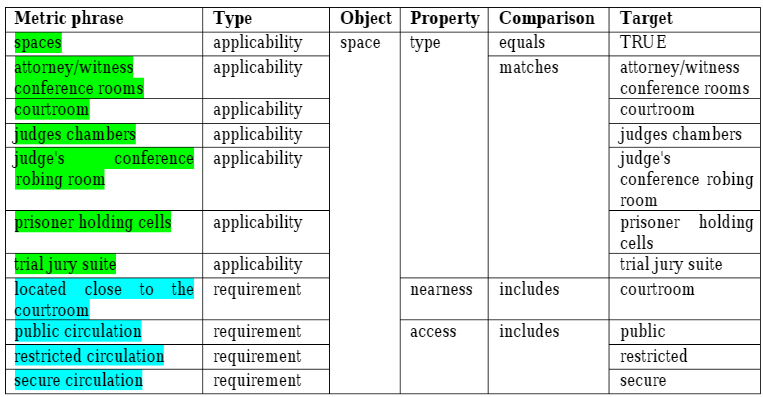


图7 利用RASE方法得到的规则



Pauwels P等人【7】在研究中意识到直接使用IFC进行规范检查存在着不足之处，因此开始尝试将IFC实体映射为语义网本体（如图8所示）以形成逻辑事实，然后利用N3Logic来对规范进行编码形成逻辑规则（如图9所示），最后利用语义网推理引擎实现规则的检查与报告生成。使用这种方式可以利用语义网的优势来弥补IFC语义的不足，但是其存在着推理引擎编写规则的能力有限、需要人工编写规则等缺点。

图8 IFC实体转化后得到的语义网本体

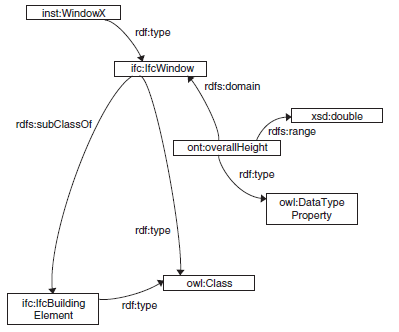
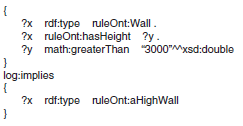
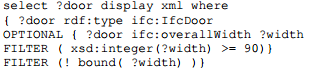


图9 使用N3Logic编写的规则示意图



Yurchyshyna A等人【8】【9】则利用语义网的SPARQL查询语言对由IFC转化而成的RDF图进行查询（如图10所示），这种做法的缺点是SPARQL本质上是一种查询语言，其对逻辑规则判断的支持先天存在不足。

图10 使用SPARQL对IFC RDF语义网进行规则检查



上述的各种利用语义网与推理机进行规范自动检查的研究中，尽管规则已不再像最开始那样需要被硬编码，但是仍然需要人参与规则的编码而无法实现完全的全自动规则检查。对此，Zhang J等人介绍一个ACC系统的架构，其中的核心功能是规范的信息抽取和设计信息的抽取和规范检查。规范信息的抽取过程分为前处理、特征生成、信息抽取和信息转化四个小阶段，前处理使用了添加记号（tokenization），语句分割（sentence splitting），形态学分析（morphological analysis）和去连接符（dehyphenation）四个NLP技术来处理自然语言文本。特征处理使用了part-of-speech (POS) tagging，phrase structure analysis和gazetteer list analysis.三个NLP技术。信息抽取利用预先定义的IE规则来匹配经过前两步处理后的文本来抽取适当的信息。信息转化将第三步抽取出来的信息利用预先定义的SMrule映射规则来匹配信息并转化为相应的Logic rules（如图11所示）。设计信息的抽取则使用了JSDAI（Java Standard Data Access Interface）工具对BIM模型进行处理和新抽取（如图12所示）。最后利用通过自然语言处理得到的逻辑规则和BIM信息进行规则检查（如图13所示）。这种方法采用了NLP来辅助规范到逻辑规则的自动转化，相比于之前的研究可以减轻规则编码的工作量，但是仍然存在操作复杂，同时仍需要定义大量匹配模板等问题。

图11 利用NLP对规范文本进行处理和逻辑规则的形成

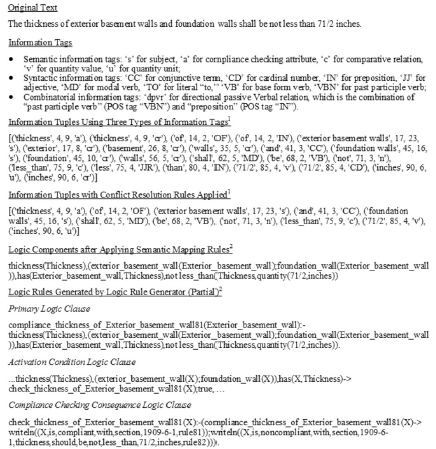


图12 利用JSDAI抽取BIM信息流程图

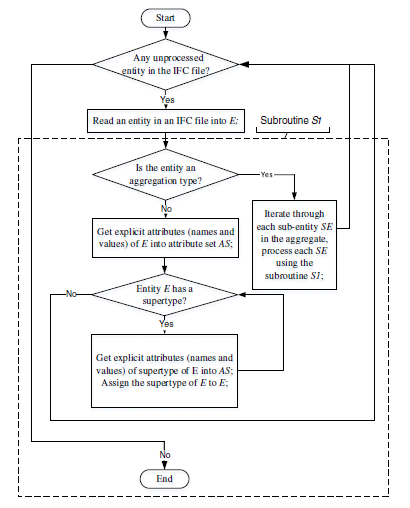
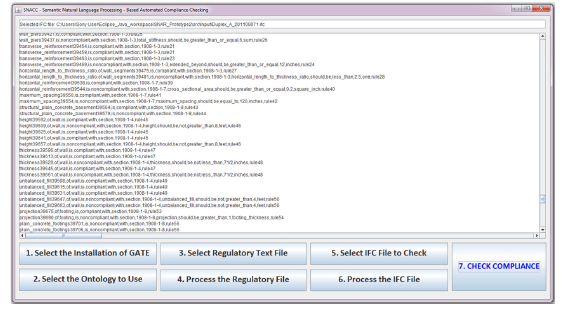


图13 规则检查界面示意图



[1] Choi J, Choi J, Kim I. Development of BIM-based evacuation regulation checking system for high-rise and complex buildings[J]. Automation in Construction, 2014, 46(10):38-49.

[2] Zhang S, Teizer J, Lee J K, et al. Building Information Modeling (BIM) and Safety: Automatic Safety Checking of Construction Models and Schedules[J]. Automation in Construction, 2013, 29(4):183-195.

[3] SMC: Solibri Model Checker, Solibri, [http://www.solibri.com](http://www.solibri.com/).

[4] EDM ModelChecker: <http://www.epmtech.jotne.com/index.php?id=512200>.

[5] novaCITYNETS Implementing IFC-Automatic code checking(e-plancheck) Pte, <http://www.novacitynets.com>.

[6] Hjelseth E. CAPTURING NORMATIVE CONSTRAINTS BY USE OF THE SEMANTIC MARK-UP RASE METHODOLOGY[J].

[7] Pauwels P, Deursen D V, Verstraeten R, et al. A semantic rule checking environment for building performance checking[J]. Automation in Construction, 2011, 20(5):506-518.

[8] Yurchyshyna A, Faron-Zucker C, Thanh N L, et al. Towards an Ontology-enabled Approach for Modeling the Process of Conformity Checking in Construction.[C]// Forum at the Caise'08 Conference, Montpellier, France, June. DBLP, 2008:21-24.

[9] Yurchyshyna A, Faron-Zucker C, Thanh N L, et al. Adaptation of the Domain Ontology for Different User Profiles: Application to Conformity Checking in Construction[M]// Web Information Systems and Technologies. Springer Berlin Heidelberg, 2009:128-141.

[10] Zhang J, El-Gohary N M. Integrating semantic NLP and logic reasoning into a unified system for fully-automated code checking[J]. Automation in Construction, 2016.