İzmir Municipality Housing and Zoning Code Analysis and Representation for Compliance Checking

Sibel Macit, M. Emre İlal, H. Murat Günaydın
Izmir Institute of Technology, Turkey
Georg Suter,
Vienna University of Technology, Austria
sibelmacit@iyte.edu.tr

Development of code compliance checking systems has been an area of research that aims to provide computational support for accurate compliance checking of building projects against applicable building codes in a time and cost effective way. Research in this area has focused mostly on representation of building codes in computational format, definition of the building model views, compliance checking algorithms and reporting.

The research presented in this paper focuses on computer-based representations of building codes for automated compliance checking. Although there has been much interest in this subject, impact on the AEC industry has not been as effective as expected. Literature survey reveals two main reasons. One reason is that most of the previous building code models were not comprehensive and too simple compared to the complex nature of building codes. To be comprehensive, the building code model should be able to represent all of the various types of rules in building codes. Second reason is that previous building code models were not maintainable. Building codes change continuously and the model should be able to accommodate addition of new rules and modification of existing ones. Non-programmer code authors should be able to easily carry out such model updates. In order to avoid these two shortcomings identified in previous efforts, current work focuses on the following two issues in two stages:

The first is the determination of all the different types of rules by analyzing building codes so that a building code model can be defined correctly and comprehensively. In this context, the organization of building code sections has been examined and the rules in the building code has been classified according to their similarities. Then, the types of rules that can be formalized for automated compliance checking have been determined.

The second issue is the development of a building code model based on a representation methodology that allows for easy updates by non-programmers. In this context, existent representation approaches has been evaluated in terms of maintainability criteria. Then the most convenient method has been used for representing rules in computer implementable format.

İzmir Municipality Housing and Zoning Code rules (IMHZCode) pertinent to residential buildings have been chosen as the focus of this study to understand the characteristics of building codes and to develop a building code model. Decomposed into a list of rules, IMHZCode is made up of 258 rules on residential buildings. In the first stage, to determine all the different types of rules that exist, these 258 rules have been classified according to their formalizability and whether they are self-contained or not. Results show that 58 percent of the rules are formalizable and self-contained, 21 percent of the rules are also formalizable but non self-contained, 13 percent of the rules are semi-formalizable and 8 percent of the rules are non-formalizable. It can be concluded that nearly 80 % rules of IMHZCode can be represented in a computer implementable format.

Following this in-depth analysis of IMHZCode, where rules have been all classified and the different types of rules have been identified, the second stage focused on picking the most suitable representation methodology that accommodates a high level of maintainability. An investigation of how far existing modeling approaches cover the needs for modeling IMHZCode has been carried out. Although there has been several building code modeling studies in literature, there seem to be no widely used formal methodology to represent rules in computer implementable format. The most common solution is to hard code rules into the compliance checking applications. This approach can be practical for implementing small number of simple rules but it is not generalizable and maintainable. The SMARTcode approach, in contrast, aims to enable non-programmers to define computable rules on their own, through simple tools. It is based on RASE (Requirement, Applicability, Selection, Exception) methodology where rules are broken down into four constructs. Non-programmers identify and markup these four constructs in the actual text of the code. The SMARTcode approach has been found as the most appropriate for representing IMHZCode.

This paper will first present the analysis results of the IMHZCode, then present the results of the evaluation of existing modeling approaches, and conclude with an illustrative example of the selected methodology's application within the context of IMHZCode.