

Automated Code Compliance Checking Model for Fire Egress Codes

Özgün Balaban¹, Elif Sezen Yağmur Kilimci², Gülen Çağdaş³

¹Istanbul Technical University, Architectural Design Computing Graduate Program, Turkey, ^{2,3}Istanbul Technical University, Faculty of Architecture, Turkey.

¹<http://www.ozgunbalaban.com>, ³<http://www.akademi.itu.edu.tr/cagdas>

¹ozgunbalaban@gmail.com, ²elif.s.yagmur@gmail.com, ³glcagdas@gmail.com

Abstract. *Architecture today has come to its most complex form. There are lots of criteria such as fire safety, structure, sustainability etc... which must be controlled by the designers. To improve the performance and accessibility of buildings, governing bodies publish different codes for each of the different criteria. Buildings must comply with these codes to get a permit for construction. The checking of the buildings according the codes is done manually by code officials. This process is time consuming, high in cost and prone to errors. To remedy this problem by using the tools like BIM and AI, systems that can automatically check the code compliance of projects are being developed. In this paper we provide an overview of the structures and capabilities of these systems and present the automated code compliance checking system that we develop for checking building models against some parts of the Turkish Fire Codes.*

Keywords. *Automated Code Compliance Checking; Fire Codes; BIM.*

INTRODUCTION

Architecture today has evolved into its most complex form. AEC companies have many criteria to check such as fire safety, acoustics, sustainability etc... during or after the design process. Buildings that are constructed for the public has to obey the legislation; and for this purpose there are different codes published depending on the type of the building or the criterion they are referring to. Thus, generally for a building project to get a building permit, a design firm must satisfy dozens of building codes. Until today regulation checks have been done manually by people. This process requires extensive manual work and time and is prone to errors. For instance, in a mass housing project done in England in 1998, the ramps for the wheel-chaired users were found to be too steep and narrow only after they were constructed. The required slope

and width for the ramps had been published in the codes but designers failed to check this information both during the early design phase and after the design check phase. The reconstruction efforts cost GBP 800,000 and it took more than eight months to solve the problem and deliver the project (Nikkhah, 2003). Such a costly mistake could have been avoided with a little effort and time with a system which can check the building projects against the codes in an automated fashion.

Manual code checking is problematic because there are vast numbers of codes and in these codes there is constant referencing between clauses, which makes locating of the information required difficult. In addition, the language used in these texts is old legal language, which is difficult to follow for designers. Moreover, there are many different

subjects of codes and these codes “create a massive volume of semi-structured documents with possible differences in formatting, terminology and context” (Lau, 2004). Furthermore, the situation becomes much more complex if the project is a product of a multinational company where the company needs to adapt to the different languages, structures and requirements of different countries. In a survey done in the industry, 85% of the architects are found to be interested in automated code compliance checking systems and in the same survey it is demonstrated in an ordinary project around 200 hours is spent for code checking (Young, Jones and Bernstein, 2009)

One more problem of the manual code checking is that it is not transparent enough. This becomes problem in the developing countries as bribery and misconduct in the approval stage is prevalent and it is difficult to recheck the given permits to test if there is corruption in the approval stage. All of the mentioned problems can be remedied with the introduction of automated code compliance checking systems.

Two advancements are progressing in support of the development of automated code checking systems. The first one is the development of BIM (Building Information Models), which is a digital building model that defines buildings with various parameters. The other one is the development of expert systems that evolved parallel with the progresses in the Artificial Intelligence. The code checking systems can be regarded as specialized expert systems.

There are ongoing studies to accomplish a fully automated code compliance checking system. The most successful one is CORENET, which has been in use and under continuous development since 1995 in Singapore. With the help of the CORENET, all code checks in Singapore are done digitally. Apart from CORENET there are other efforts in Norway, Australia, and USA but they all have concentrated on some special topics like accessibility.

The current code checking performance in Turkey has some problems. AEC companies do not pay enough attention to the regulations. This fact is

prevalent in most of the criteria of the buildings like accessibility, fire safety, earthquake safety etc... For example, most of the public space in Turkey lacks properly designed accessible structures. In addition, Turkey is a country constantly under threat from earthquakes and several major earthquakes happened in the last fifteen years. The high mortality rate after those earthquakes is found to be because of poorly designed and poorly checked structures. The reasons for this poor design and poor checking maybe lack of experts, corruption and lack of proper methods to inspect the designs. All of these problems can be improved by introducing automated code compliance checking systems aimed to work with Turkish codes.

In addition, regarding the easiness and quickness of having one project checked against the required building codes and getting an approval for one's project, Turkey is one of the worst countries in the World. According to a report by Doing Business organization, Turkey is 155th out of 183 countries with regard to the easiness of dealing with construction permits [1]. Primary reason for this poor performance is that it takes 189 days to take a permit and from these days, around 150 days go to code checking of the project. When we compare this value with Singapore's performance where it takes only 26 days [2] to get an approval thanks to CORONET, we can see the contribution of the automated code compliance checking systems to compliance checking.

In this study, we aim to develop an automated code compliance checking system for checking building models according to some clauses from the Turkish Fire Codes. Accomplishing a fully automated code checking system is a huge undertaking thus we have restricted our area to egress clauses in fire codes. This is because we have limited resources in terms of time, budget, work force etc... But according to the results of our current work, we will continue on broader spectrum. In the paper, we first present brief information about automated code compliance checking systems, and examples of them. We continue with the description of our system, which we named Fire Codes Checker (FCC). At

the end of the paper, we will discuss on our findings in this study, the strong and weak sides of FCC and we will look onto future prospects.

AUTOMATED CODE COMPLIANCE CHECKING

Automated code compliance checking is the act of checking a building model against regulations, using computerized processes. There is an ongoing effort to automatize the code compliance checking process. The earliest efforts towards automatization of code checking has started in 1960s with Fenves' (1966) effort on structuring of the codes in decision tables so that they can be resolved easily. This work was manual. Later computers started to take part in the studies, one example by Fenves and Wright (1977) was about software tools to manage regulations. With expert systems coming into the scene, these efforts in the structuring of the codes and regulations shifted into developing systems that can automatically assess some clauses from regulations. These systems used 2D CAD drawings as the source of building information. 2D CAD drawings cannot accommodate vast numbers of properties that building elements have. As a result, the studies were restricted to some parts in areas like fire safety, accessibility etc...With the advance of BIM, the amount of building information improved drastically. Therefore, systems that can check variety of different regulations became a possibility. Today there are ongoing studies in the countries like Singapore, Australia, Scandinavian countries. Further details on ongoing research will be presented in details in the next chapter.

There are some important cases to be considered while developing these systems, (Tan et. al., 2010), these are; 1) rule checking software must point out which object does not obey to the code, 2) most of the codes apply different rules for different situations so rule checking software must consider all of these situations, 3) codes can be changed frequently and the program must be adjusted accordingly, 4) rules are different on each region or country and these changes must be applied in the program,

5) finally if there is not enough conversation between the developers and the rule makers, the software can work in an erroneous way (Han, Kunz, and Law, 1997).

From the early works and the general structure of all previous code checking system examples Eastman et al. (2009) divides the code checking process into four stages. These are; 1) "Rule Interpretation" where the written rules are translated into computer recognizable forms, 2) "Building Model Preparation" where the design are transferred into digital world via BIM software, 3) "Rule Execution" where the rules are applied to the building models and 4) "Rule Reporting" where the results and the errors (if any) are displayed to the user.

To accomplish working automated code compliance checking system these four stages must be present in the systems. In the Rule Interpretation stage, there are two different ways to translate human written codes into computer interpretable rules; first one is to use a human programmer so that he will translate all definitions to digital. This method has its shortfalls; in every change in the written code the programmer must recheck the system and make the needed changes. In addition, in this method one cannot avoid human factor in code checking as errors in translation may result in corrupted system. The second method for translation is to use Natural Language Processing (NLP) systems and predicate logic, which makes the translation process automatic. The drawback for this technique is, written codes have complex language, and thus it is difficult to process this information.

In Building Model Preparation stage, the building that the rule checking will be done, must be prepared in a BIM software so that all the information about the building is accessible by the automated code checking system. However, there are dozens of different BIM software, which use different proprietary file formats thus it is difficult to prepare a system that can access all the information that is generated in different file formats. This situation is solved with the introduction of a neutral object oriented file format, Industry Foundation Classes (IFC) by a

consortium of developers of the BIM software. Nearly all BIM software has the ability to convert its proprietary file format into IFC file; therefore using IFC in automated code compliance checking systems ensures interoperability between different software.

In the Rule Execution stage, the translated rules are carried out and the checking is done. In this process it is important to check all the entities, and to not to leave any unchecked clauses in the codes. And finally in the Rule Reporting stage the results of the code checking process are reported to the user. It is important for user to know the cause of the problem if any of the clauses fail.

AUTOMATED CODE COMPLIANCE CHECKING EXAMPLES

In this section we present the automated code compliance checking examples that we reviewed to derive the features to incorporate and the strategies to be followed in developing the FCC.

CORENET - Singapore

CORENET is an acronym for Construction and Real Estate Network, and it is a project started by Singapore Ministry of National Development in 1995 to *"propel the construction and real estate sector into the new millennium by re-engineering the business processes with state-of-the-art IT to achieve a quantum leap in turnaround time, productivity and quality"* [3]. Singapore Building and Construction Authority (BCA) builds and maintains the CORENET. It is the first working system that became operational.

CORENET consists of three modules; these are CORENET e-Submission, CORENET e-PlanCheck and CORENET e-Info. CORENET e-Submission is a web-based system and it aims to collect the entire project related documents and drawings needed for the code checking process against variety of different topics in one place. This system has many benefits compared to the traditional building approval process. These are: 24/7 availability, less bureaucracy, improved transparency and speed. E-Info, is a website for presenting the entire official documents about construction and real estate in one

data format online. Project developers can access these documents anywhere, anytime. e-Info uses Extensible Markup Language (XML) for transforming written documents into machine recognizable format without losing its human readability. It is used for transferring documents for use in different applications. E-PlanCheck module is the most ambitious part of CORENET and its aim is to allow *"designs for new buildings to be digitally checked against building codes, using automated procedures, rather than manual paper based processes"* [4].

CORENET uses IFC format in checking the building models. However, IFC is focused on geometry and lacks the information needed in the code compliance process. To complement the limited capabilities of the IFC, novaCITYNETS built a Code Checking Object Model (CCOM) which is named as FORNAX. FORNAX is implemented to extend the information found in the IFC. It is a *"model representing both the building geometry models in 3D and the semantics information such as the relationships and the behaviors of the building elements"* (Xu, Solihin and Huang, 2004). The FORNAX objects are encapsulation of simple building components, by this way the programmers do not need to develop separate algorithms for all the required calculation that is needed. Instead, it is possible to use FORNAX objects and their supplemented functions and attributes to check the requirements of several codes. As a result translating a written code into computer process is straightforward.

FORNAX does not replace IFC. It takes basic object information and its associated geometry from IFC model and using some geometric operations finds out information such as spatial information, network information and design constraints and adds it into its repository. Spatial information gives relative place of other objects, network information makes drawing of paths and assessing the connection of spaces possible and design constraints is about how a certain object is defined. The FORNAX system is composed of four parts; database for storing information, ACIS and Open Cascade as geometry engines and lastly IFC.

E-PlanCheck can output the results in the popular document formats such as PDF, DOC or HTML. They are presented within a website. It supports giving reference to the written code clause while listing the elements that do not comply with the codes. The reporting module has graphical presenting capabilities.

CORENET is used efficiently for automated code compliance checking in Singapore. It is much more mature than other examples we will review here. Today thousands of engineers and architects use CORENET successfully. However, CORENET is not aimed for use during design stage; it is only used by the governmental agencies. In this aspect it differs from the other systems.

DesignCheck - Australia

The Cooperative Research Center for Construction Innovation funded the DesignCheck project and it was undertaken by University of Sydney and Commonwealth Scientific and Industrial Research Organisation (CSIRO). The aim was to develop an automated code checking system for Australia which will *"enable quick and easy compliance assessment against building codes and assist designers in finding potential problems early"* (Ding et. al., 2006). This system initially focused on Accessibility codes.

Rather than starting from scratch, DesignCheck team aimed to use the existing rule based systems and develop them further according to the requirements of the project. Thus, DesignCheck project group started with a review of two existing commercial rule based systems: Express Data Manager (EDM) and Solibri Model Checker (SMC). As a result of this review, they have decided to continue on the project using EDM primarily because it had offered the option to freely modify the rule schema which SMC lacked. (Ding, 2004)

DesignCheck uses object oriented techniques for transforming written statements from the building codes into computer interpretable structures. DesignCheck team has prepared a pre-implementation specification structure that is used before the translation process begins. This structure consists

of: Description which is the written statement in the clause, Performance requirements to satisfy the Description, Objects required in the clause, Properties and Relationships of these objects, and finally Domain specific knowledge for interpretation which consists of functions that will be used in the checking (Ding et. al., 2006). Therefore, the translation occurs in two steps, first from written structures to object based interpretation by using preimplementation specification structure and then to EDM rule schema. Building model, that will be worked by the system is modelled in ArchiCAD 9, and imported to the system in IFC format. IFC format was not adequate in supplying the necessary information for the accessibility code checking thus DesignCheck project team has devised Design Check Internal model which include the information that is not supplied by IFC. Finally EDM checks the DesignCheck model with the constructed rule schema and results are published to the users.

The difference of DesignCheck is that it gives its users the option to select the design stage for code checking, this allows DesignCheck to be used in early, detailed and documentation design stages.

Statsbygg - Norway

After the success of CORENET, European countries started to search for ways to take advantage of BIM. From those, Nordic countries like Norway, Denmark, and Finland were the leading countries in this effort. In Norway, this effort turned into a project named Byggsok. Byggsok is an e-government system, which has three modules: information, zoning, building (Rooth, 2005). The information module stores and distributes documents about zoning proposals and building application processes. It has achieved to gather all the information that is published by 433 different local government entities and made these documents accessible online. The zoning module is for applying for zoning approval process. It also makes communication between authorities and developers possible. It sends zoning approval request to the local government for checking. The building module is for submitting building

plans to the system. The submission is done by using IFC building model.

After these two experimentations of automated design checking, one major BIM project named HITOS (Tromso University College) was used to experiment interoperability of different platforms along with rule checking capabilities by Statsbygg.

Two different rule types were examined in HITOS. The first one was about spatial program validation. For this purpose, the project team used dRofus as the rule based system. In the project different teams designed different parts of the building simultaneously, thus it was required that the system allows simultaneous operation. dRofus acts as a database that allows managing of the architectural programs, technical functional requirements and equipment from early stage planning.

dRofus does not require rule interpretation as it is a dedicated application which has the rules preloaded. dRofus reports the required spatial program along with the actual space area. Designers

can see the difference and correct any problematic spaces.

The other rule checking system handles the codes for accessibility. To succeed in this, the Statsbygg team used SMC rule based system.

THE FIRE CODES CHECKER

As a result of the review, we have decided to incorporate the following features and strategies that will be followed in FCC; 1) in the first stage of the study the model would be a standalone system, which would be located in the computer that would be used in the checking process. FCC would not have online checking capability, but in the future this option will be added to allow governmental agencies to check projects from one place just like in the CORENET example; 2) three modules that are used in the CORENET, information, submission and code checking was found successful by the other efforts. In Byggsok, similar structure is used. In FCC, we have taken the two module: information and code check-

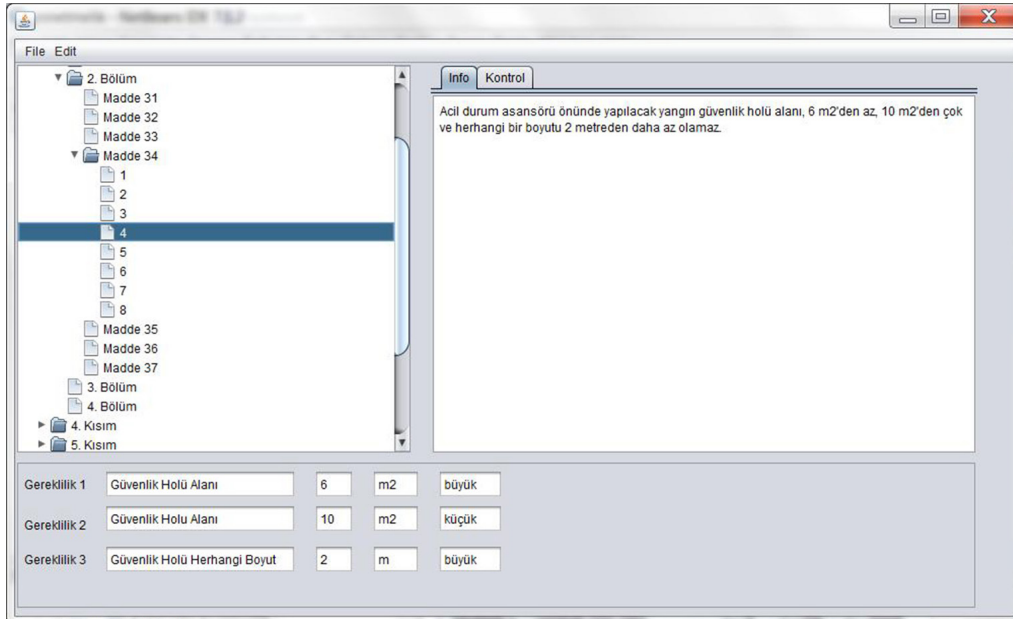


Figure 1
The interface of the Informa-tion module.

Figure 2
Turkish Fire Codes in XML file
format.

```

309 <Kisim id="3" title="Kaçış Yolları, Kaçış Merdivenleri ve Özel Durumlar">
310 <Bolum id="1" title="Genel Hükümler">
311 <Madde id="30" title="Kaçış güvenliği esasları">
312 <bent id="1" guid="3_1_30_1" stage="early">
313 İnsanlar tarafından kullanılmak üzere tasarlanan her yapı, yangın veya diğer acil
durumlarda kullanıcıların hızla kaçışlarını sağlayacak yeterli kaçış yolları ile
donatılır. Kaçış yolları ve diğer tedbirler, yangın veya diğer acil durumlarda can
güvenliğinin yalnızca tek bir tedbire dayandırılmayacağı biçimde
tasarlanıır.
314 </bent>

```

ing. However, FCC would lack the submission module as it is a standalone system which the building models are loaded into manually; 3) object based interpretation used in DesignCheck have made the translation of the written codes into computer interpretable structures easier, we have used similar approach while preparing for the rule translation phase; 4) the ability to choose the design stage when checking a project against codes that is available in DesignCheck, would also available in the FCC.

FCC consists of two modules; Information and Code Checking. Information module is developed to present the clauses of Turkish Fire Codes to the users. Designers can select part, section, article or sentence number to access the needed information.

In addition, users can search for information within the system, which speeds up information retrieval. The system also has a glossary feature, such that when a user encounters an entity that is unknown to him, he can click and learn its meaning or definition. While presenting the information in the sentence, the information module shows the requirements that need to be satisfied to comply with this clause. For instance in the example in Figure 1, the requirements for this clause are security hall must be bigger than 6 m² and less than 10m² and any side of it must be bigger than 2 m.

To develop this module we first converted the written Turkish Fire Codes into electronically readable file format and for this reason we have used extensible markup language (XML). In the XML file we preserved the hierarchical structure of the Turkish Fire Codes where the codes are divided into parts, parts into sections, sections into articles, and articles into sentences. These are separated by using tags representing these branches such as <kisim> (part), <bolum> (section) etc... The sentences have the statements; these are separated into the design stages that they apply like early, detailed or documentation stages and this information is included in the XML file with the attribute <stage="early">, <stage="detailed"> or <stage="documentation"> [Figure 2].

In second module, code checking is performed. Code checking requires rule schema and this schema is developed by the programmer by processing the written information found in the Turkish Fire Codes. In this process, object based interpretation similar to the one that is used in DesignCheck is used to ease up the translation process. For example, article 47, sentence 1 in Turkish Fire Codes dic-

Figure 3
Preimplementation specification structure used in the interpretation of the written codes to rule schemas.

Article 47 – Sentence 1
Description:
The doors found on the egress routes must have clear width not less than 80cm and height not less than 200cm. The doors must not have any thresholds. Rotating doors and turnstiles cannot be regarded as escape doors.
Performance Requirements:
All the doors that are on the egress route must have clear width more than 80cm and height more than 200cm. There must not be any thresholds on these doors. Rotating doors and turnstiles cannot be regarded as escape doors.
Objects:
Space and Door
Object Properties:
Door_type, Door_clearwidth, Door_height, Space_egress
Object Relationships:
Space contains doors
Domain-specific knowledge for interpretation:
RequiredDoorClearWidthOK
IF Door_clearwidth is more than 80 cm., THEN return TRUE.
RequiredDoorHeightOK
IF Door_height is more than 200 cm., THEN return TRUE.
DoorsNotRotating
IF Door_type is not rotating door or turnstile, THEN return TRUE.
DoorsOnEgressPath
IF Space_egress contains Door, THEN return TRUE.
DoorDoesNotHaveThreshold
IF Door does not have threshold, THEN return TRUE.

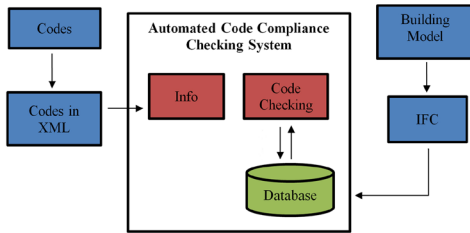


Figure 4
The overall structure of FCC.

tates “The doors found on the egress routes must have clear width not less than 80cm and height not less than 200cm. The doors must not have any thresholds. Rotating doors and turnstiles, cannot be regarded as escape doors” [5]. The object oriented interpretation that is used in this sentence can be seen in Figure 3. The functions required to check compliance of building models against the clauses are implemented in Java language by using the pseudo codes that are developed in the object based interpretation phase.

FCC uses IFC to retrieve building model data needed for code checking. Since IFC is a neutral data format supported by most of the BIM applications, use of IFC format in FCC supports retrieval of building information from all such applications. In this step, all the IFC objects and their properties are stored in FCC’s database by preserving entity hierarchy in IFC. For instance, all IFCWall objects are stored in one table, and its properties like Nominal-Length are stored in another while related to one to another.

After the database is populated, the system becomes ready for rule checking. If the user selects a sentence number from the fire codes, the system locates the corresponding function to apply the checking. If there is no special selection the system follows full check, which means checking of all sentences. If an entity complies with the sentence, the result is stored as true in the database, and if it does not comply with the clause, the result becomes false. The overall structure of the FCC can be seen in Figure 4.

After rule checking module finishes checking the building model, the results are given to the user.

The results can be PASS if all of the codes are satisfied, FAIL if one or more clauses fail and N/A if there is some missing information, which prevents the program to run. If the result is, FAIL then the program reports which clauses of the code fail and the origin of the problem.

CONCLUSION AND FUTURE PROSPECTS

From the reports of the other efforts it appears that complete automated code compliance checking systems are many years away. Our work too, is in its development stage. There are many steps to succeed to develop a fully automatic code compliance checking system.

In FCC the rule base does not cover all of the fire codes, just some parts about the egress routes are covered. First, we must cover the whole of the clauses in the fire codes. This has its difficulties, as some clauses require some intrinsic information, which is difficult if not impossible to acquire from IFC.

FCC lacks graphical reporting features which is required for easily locating the problematic building element. For now the system only supports textual reporting in which the element is referred by its number. This makes finding the element harder. One another thing to do is to build Turkish dictionary in parallel with the international effort (IFD). This is important as the tools that are used in design firms are global and it is important to map Turkish terms with the global ones.

REFERENCES

- Ding, L, Drogemuller, R, Jupp, J, Rosenman, M and Gero, J 2004, ‘Automated Code Checking’, *CRC CI International Conference*, Gold Coast, Australia.
- Ding, L, Drogemuller, R, Rosenman, M, Marchant, D and Gero, J 2006, ‘Automating code checking for building designs – Designcheck’, *Clients Driving Innovation: Moving Ideas into Practice*, March 12-14.
- Eastman, C, Lee, J-m, Jeong, Y-s, and Lee, J-k 2009, ‘Automatic rule-based checking of building designs’, *Automation in Construction*, 18(8), pp. 1011-1033.

- Fenves, SJ 1966, 'Tabular decision logic for structural design', *Journal of Structural Engineering* 92 (ST6), pp. 473–490.
- Fenves, SJ, Wright, RN 1977, 'The representation and use of design specifications', Technical Note 940, NBS, Washington, DC.
- Han, C, Kunz, J and Law, K 1997, 'Making automated building code checking a reality', *Facility Management Journal*, 22-28.
- Lau, GT 2004, *A Comparative Analysis Framework for Semi-structured Documents, with Applications to Government Regulations*, Phd. thesis, Stanford University.
- Nikkhah, R 2003, 'Laing sues over disabled access', *Building Magazine*.
- Rooth, O 2005, 'Byggsok plan - informasjon - bygning', *IAI Conference*, Oslo.
- Tan, X, Hammad, A and Fazio, P 2010, Automated Code Compliance Checking for Building Envelope Design, 24(2).
- Young, NW, Jones, SA, Berstein, HM 2007, *Interoperability in Construction Industry*, McGrawHill.
- Xu, R, Solihin, W and Huang, Z 2004, *A Code Checking Object Model for Computer Aided Building Design*.
- [1] <http://www.doingbusiness.org/data/exploreeconomies/turkey/#dealing-with-construction-permits>.
- [2] <http://www.doingbusiness.org/data/exploreeconomies/singapore/#dealing-with-construction-permits>.
- [3] <http://www.aecbytes.com/buildingthefuture/2005/CORENETePlanCheck.html>.
- [4] [http://www.civ.utoronto.ca/sect/coneng/i2c/D-Base/Onthe web-short/Case studies/Singapore-CORENET.pdf](http://www.civ.utoronto.ca/sect/coneng/i2c/D-Base/Onthe%20web-short/Case%20studies/Singapore-CORENET.pdf).
- [5] http://www.yangin.org/indir.asp?dno=701&dosya=yangin_yonetmeligi_2009.pdf.

