INFORMATION TO USERS

This manuscript has been reproduced from the microfilm master. UMI films the text directly from the original or copy submitted. Thus, some thesis and dissertation copies are in typewriter face, while others may be from any type of computer printer.

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleedthrough, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps.

Photographs included in the original manuscript have been reproduced xerographically in this copy. Higher quality 6" x 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

Bell & Howell Information and Learning
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA
800-521-0600

UMI®

Building Inspection with Automated Code Compliance Checking

Tang Hung Nguyen

A Thesis

at

The Centre for Building Studies

Presented in Partial Fulfillment of the Requirements
for the Degree of Master of Applied Science at
Concordia University
Montreal, Quebec, Canada

June 1996

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

0-612-44876-2

Canada

ABSTRACT

Building Inspection with Automated Code Compliance Checking

Tang Hung Nguyen

Inspecting existing buildings for conformity with current regulations is often difficult to carry out for people lacking expertise in code compliance checking. The difficulty is due to the complexity of building codes which inherently contain a large amount of regulatory information with multiple cross references. Building inspectors on site do not have much time to properly interpret and judge the building regulations. Misinterpreting and overlooking the building code information may lead to serious problems with respect to building safety due to code violations. To overcome such problems, building inspectors must be provided with a computer-based tool which facilitates the building code compliance checking process.

The present research aims at developing an automated approach for the diagnostic of existing buildings during inspection. The proposed methodology is that of an intelligent system combining current computer technologies such as expert systems, databases, and hypertext techniques. The expert system represents and reasons with specialist knowledge to diagnose problems with code compliance checking whereas the database and hypertext techniques are efficient for handling cross references among distinct building subsystems and disciplinary viewpoints in data management systems.

The development of such an automated code checking system is characterized by two tasks: the establishment of a knowledge base consisting of building code requirements in Part 3 of the National Building Code of Canada, and the incorporation of a data management module. The knowledge base is developed through three steps. First, code documents are broken into specific categories to be considered during compliance checking. Next, following the inherent logic for each compliance category, decision trees are developed in such a manner that the data required for checking is systematically collected with a minimum number of queries. The final step is to structure decision trees in a format suitable for computer-based system implementation. The data management module incorporated into the automated system allows the user to browse through documents. By using the hypertext technique, the text of the National Building Code of Canada as well as relevant case studies are stored as external databases that are dynamically linked to the automated code checking system.

The research methodology has been implemented in a software prototype known as Health and Safety Expert System (HASES). The prototype system relies on knowledge and reasoning to interpret the requirements of Part 3 of the National Building Code of Canada. HASES aims at facilitating the inspection of existing buildings by simplifying the data collection and compliance checking processes, generating reports, and providing access to texts and relevant case studies on the fly, as an inspector walks around a building.

ACKNOWLEDGMENT

I wish to express my gratitude to my research supervisors: Dr. Claude Bedard for his valuable guidance and supports; and Dr. Kinh Huy Ha for his precious insights and encouragement.

I wish to thank Dr. Paul Fazio for offering me an opportunity to get involved in a research project at SIRICON Inc, which is helpfully related to this thesis.

My sincere thanks to SIRICON Inc. for providing time and computer resources to carry out this research. I am thankful to Sebastiano Depanni, professional engineer, for his discussions and interests shown through the project implementation; and to Kathleen Rose, junior building engineer, for her proofreading of this thesis. Francesca Ottoni and Max St. Denis, Co-op students at Center for Building Studies (1995), are also thanked for their contribution to the development of decision trees and Winhelp files for the code document.

This research project was supported by a postgraduate scholarship from the Natural Sciences and Engineering Research Council of Canada, the J.W. McConnell Memorial Graduate Fellowship from Concordia University, and a grant from Centre for Building Studies. This support is most gratefully acknowledged.

TABLE OF CONTENTS

Chapter 6

List of Figures

List of Tables

Chapter 1

INTRODUCTION

1.1 General

Building code compliance checking is considered an integral part of engineering design and construction processes. At the design stage, the code checking process is often very complex to carry out. However, it becomes even more complicated to perform for existing buildings that are renovated or modified. These buildings must be inspected periodically for compliance with the current code requirements to ensure the health and safety of occupants. Usually, code violations are difficult to identify for people lacking the expertise in code compliance checking. In addition, many on-site inspectors do not have much time to properly interpret and judge the building safety regulations. They often find code provisions confusing and overwhelming because of the multiplicity of hierarchical subdivisions and cross references. Misinterpreting and overlooking the building code information may lead to serious problems with respect to building safety due to code violations. Recently, the emergence of information technologies have provided new effective approaches to automation in building code compliance checking. This makes it possible to develop a computer-based system capable of facilitating the building inspection process.

1.2 Building Inspection Issues

Building inspection is an examination of construction work completed or in progress to verify its compliance with the current building code requirements. Normally, the construction is executed only when the compliance with all applicable code provisions of the drawings and specifications have been achieved. However, numerous existing buildings have been found to violate current code requirements. In effect, a recent survey on Agriculture Canada buildings conducted by Public Works Canada revealed about 1700 occurrences of code violations in 19 buildings due to occupancy changes, renovations, and building code revisions [PWC 1991]. It has also been reported that most violations fell in categories of fire, occupancy and building hazards which correspond to the code requirements found in Part 3, Use and Occupancy, of the National Building Code of Canada [NBCC 1990].

One of the main reasons for the regulation violations is the lack of proper in-progress building inspection in which inspectors play an essential role. The main duties of a building inspector can be summarized as follows [Alhussayni 1996, Personal Communication]:

- Carry out technical and detailed inspection from the beginning to the end of the construction work to ensure conformance with construction permits.
- Ensure compliance with applicable code provisions by determining the symptoms of diagnostic problems in code violations, documenting the code violations (if any), and providing suitable remedial measures to meet the code requirements.
- Monitor and make sure that the project progresses according to schedule.
- Reject or stop construction work, modification or demolition carried out without permits, violating building code requirements or exposing danger to public safety.

Basically, the building inspection process includes three sequential sub tasks: first, information is gathered about various aspects of the building; second, the information is compared with applicable code provisions to verify for conformance; and finally, compliance results are documented as inspection reports which include explanations for code violations, if they exist, as well as suitable recommendations to meet the code requirements. To successfully accomplish this process, the building inspector must be knowledgeable about the building codes, consistent in compliance checking, and methodical in report writing. However, the on-site inspector normally may not have enough time and expertise to interpret and identify correctly the building codes which contain massive, wide-ranging, and very complex regulations of a broad scope [Nguyen 1996, Personal Communication]. Although the field inspector is trained in basic code usage and interpretation as well as techniques of inspection, the quality of his performance may be limited. Therefore, it is proposed to develop a computerized system which can assist on-site inspectors in facilitating the code compliance checking process.

1.3 Scope and Objectives

Checking a design or an existing building for conformance with applicable building codes is a tedious, laborious, and complicated task. Misinterpreting or overlooking provisions of the building codes, which is often the case with inexperienced users, may lead to serious consequences. An expert system can be developed to effectively simulate human competence in code compliance checking because the knowledge contained in codes and standards is largely in the form of rules, and is generally used with a systematic and sequential process to verify the compliance of buildings [Frye et al. 1992].

The objective of this research is to develop a systematic methodology to obtain a computer-based system which will assist code practitioners in automating the compliance checking process in building inspection. The proposed methodology is that of an intelligent system combining current computer technologies such as expert systems, databases, and hypertext techniques. The regulatory source used to extract knowledge for the automated compliance checking system includes code regulations found in Part 3 of the National Building Code of Canada [NBCC 1990]. The knowledge and information contained in this part are structured and represented by a knowledge-based system approach. An external database is used to store the general building data which can be extracted as needed during the code checking process. Users can view, edit, update and delete or save this data in the database. Also users can access to the NBCC text and relevant case studies which are, by means of databases and hypertext techniques, incorporated in the system to assist in identifying the applicable code provisions and providing suitable solutions for problems with code violations. Once a compliance check is completed, its final result is automatically displayed as an inspection report including building information, code requirements, NBCC references, and check results. The report then can be sent to an external text file that can be viewed and edited from any text editor.

A research prototype named Health and Safety Expert System (HASES) was implemented in an attempt to demonstrate the practicality and feasibility of the proposed methodology. The HASES implementation includes the development of a knowledge base, a data query user-interface, a compliance checking mechanism, and a data management module. HASES has been tested to validate the knowledge base in building inspection.

1.4 Organization of the Thesis

The following chapter presents general characteristics of the NBCC and a review of literature on electronic building codes.

Chapter 3 discusses the critical issues in developing an automated code compliance checking system, identifies some information technology applications as alternative solutions, and presents the proposed methodology for the development of an intelligent system to assist in building inspection.

Chapter 4 describes the basic components of an intelligent system including the expert system and the data management module. A description of the knowledge representations of the expert system is also provided in this chapter.

Chapter 5 provides the details of the HASES implementation, the main characteristics of the research project, and the description of a commercial tool to be selected for developing the prototype system. The validation of HASES as well as comments on the prototype system are also summarized.

Chapter 6, as a conclusion, identifies the contributions of this research project and points out the feasibility of the study outcome. Several recommendations for extending the present work are also presented.

Chapter 2

BUILDING CODES AND ELECTRONIC CODES

2.1 Introduction

Building codes are generally difficult to learn, hard to use, and onerous to apply [Rosenman and Gero 1985]. "Information technologies have made it possible to provide instant and accurate access to the vast amount of information contained in building codes and standards" [Vanier et al. 1994]. Hence, numerous research efforts are in progress around the world to facilitate the building code compliance checking process through computer-based approaches.

This chapter provides the contextual background material for the knowledge of automated code checking systems. The building codes in general and the NBCC in particular are briefly described to indicate the research direction and to define the extent of the domain. Some previous work done in the field of electronic codes are summarized to form part of the basis for the methodology to develop an automated code checking system: the subject of the study.

2.2 Overview of Building Codes

Building codes play an essential role in construction industry: they are indexes for constructing safe, durable and reliable structures. Most building codes are developed by the national authorities of each country; and they are mandatory at a national, regional or local level. In Canada, the NBCC is a legal document that is adopted by all provinces. It is basically a set of uniform building regulations and standards for the safety of buildings with reference to public health, fire protection and structural sufficiency [NBCC 1990]. The NBCC provisions are intended to be applied to the construction of buildings, the extensions or modifications due to a change in occupancy and the improvement of buildings [NBCC 1990].

The following subsections provide a brief description and structure of the NBCC as well as its use in the construction industry. Some Information Technology applications to the field of building codes, including NBCC electronic prototypes and projects, are also presented.

2.2.1 General Description of Building Codes

Every country around the world has different national building codes. Each is generally intended to provide overall regulations for life, health and safety protection. In the United States of America (USA), building codes are developed by independent private organizations: the Uniform Building Code of the International Conference of Building Officials (CBO), the Standard Building Code of the Southern Building Code Congress International, Inc. (SBCCI), and the National Building Code of Building Officials and Code Administrators International, Inc. [BOCA 1993]. These three model building codes are currently adopted by the various regions, states or municipalities of the USA.

The National Building Code of Canada is drafted in such a way that it may be adopted or enacted for legal use by any jurisdictional authority in Canada [NBCC 1990]. With the exception of some provinces that use their own regional codes by modifying the NBCC to suit their provincial requirements, most provinces have adopted the original form of the NBCC. Differences among the national and provincial building codes have resulted in some of the difficulties in establishing a comprehensive computer-based system to assist in automatic code checking in Canada. The NBCC includes not only building standards and regulations for life, health and safety protection; but also other areas such as environmental protection, barrier-free design [NBCC 1990] and energy conservation. Any requirements extending the scope of the Code must undergo a thorough consultation with regulatory authorities before taking effect [NBCC 1990].

2.2.2 Hierarchical Structure of the NBCC

Similar to most national building codes, the NBCC is structured in a well-defined format as shown in Table 2.1. Its hierarchical structure consists of nine main subdivisions named NBCC Parts, each concerning general regulation areas. The next level of subdivision called NBCC Sections, deals with more specific regulations with different Headings, such as 'Requirements for Fire Safety', 'Safety Requirements within Floor Areas', 'Requirements for Exits', 'Service Facilities', etc. An NBCC section is subdivided into a number of subsections which address very specific subjects such as 'Exits through Lobbies', 'Exits Signage', etc. The lowest subdivision having a Heading is NBCC Article that contains multiple NBCC Sentences stating NBCC regulations. Each sentence may contain a list of the conditions or requirements for conformance, called NBCC Clauses and Subclauses.

All NBCC Parts have the same format of hierarchical structure. The following example demonstrates the format of the NBCC:

Part 3 Use and Occupancy
Section 3.4 Requirements for Exits
3.4.3. Width and Height of Exits
3.4.4.5. Exit Capacity

\- - - - - - - - -

(3) The required width of means of egress serving a Group A, Division 4 Occupancy shall be determined by multiplying the occupant load of the area served by

(a) 1.8 m per person for
(i) aisles, . . .

Table 2.1. **NBCC Hierarchical Structure**

Number	Subdivision	Heading
3	Part	Use and Occupancy
3.5	Section	Service Facilities
3.5.2	Subsection	Service Rooms
3.5.2.1	Article	Fire Separations around Service Rooms
3.5.2.1.(2)	Sentence	N/A
3.5.2.1.(2).(a)	Clause	N/A
3.5.2.1.(2).(a).(ii)	Subclause	N/A

In general, each of the lowest subdivisions in the NBCC states a specific requirement. However, one statement may refer to another statement, building standards or appendices resulting in multi cross-references throughout the building code document. The following example illustrates the cross-references in the NBCC:

> 3.2.4.16 Sprinklers in Lieu of Heat Detectors
> (1) Heat detectors in Articles 3.2.4.10 and 3.2.4.11 need not be provided where an automatic sprinkler system conforming to sentences (2) to (5) and Article 3.2.5.13 is installed throughout the floor area (NBCC, 1990).
>
> \- - - -
>
> 3.2.4.17. Manual Pull Stations
> A manual pull station shall be installed in every floor area near every required exit (See Appendix A).

2.2.3 Use of Building Codes

The building codes are used not only within the design stages but also during the construction process including extension or renovation of existing buildings. In all design phases, due to its generally complex nature, the building code is rarely regarded as a design index or a design guideline and the designers have tendency to avoid official code literature [Liebing 1982]. Most design professionals do not want to take time to explore and understand the details of the code. Many find the building regulations to be restrictive or imposing and, therefore, impede the development of the basic design concept. In the meantime, others use buildings codes to extract the design constraints for building design [Fazio et al 1990]. In the construction process, the standards and codes are used to regulate the construction. Building inspectors use the building codes to verify the construction for conformance with the applicable regulatory requirements. In addition to building inspectors, other participants such as owners, architects, engineers, contractors, etc. must share the responsibility for the code compliance of construction drawings and specifications [Liebing 1982, Alhussayni 1996 Personal Communication]. The Code must take into account a wide variety of buildings (institutional, industrial, or residential) and building situations (extension, modification, or renovation). For example, the NBCC Sections 3.3.4. and 3.3.5. apply to floor areas or parts thereof used as residential and industrial occupancy respectively.

2.3 Electronic Building Codes

Recently, computer applications to the management of regulatory information have attracted considerable attention in construction industry. Previous studies on the application of information technology (IT) to building codes [Frye et al. 1992, Kahkonen et al. 1992, Kumar et al. 1995] confirmed that electronic codes are in demand. Also, these surveys indicate that building codes have been viewed as a promising area for automation for a long time. In Canada, a survey on the NBCC usage conducted by the Institute for Research in Construction (IRC) has concluded that the majority of the building code users have a keen interest in electronic code products [Vanier 1989, 1994]. However, there are still many obstacles to overcome in the development of an NBCC comprehensive electronic code, which are identified in the following subsection. Several available electronic code applications are also described.

2.3.1 Difficulties in Developing Electronic Codes

The emergence of advanced computer technologies such as hypertext, expert systems, and expertext systems which are presented in the following chapter, are providing alternative approaches to the development of electronic building codes. However, the progress of producing electronic retrieval systems for the NBCC is impeded due to the complexity of user demands, provincial and national code differences, and classification of building code information.

Building code users include interdisciplinary professionals such as architects, engineers, building inspectors, etc. as well as non-professionals such as building owners and manufacturers. The code practitioners have different levels of expertise in code usage and various needs for the application of code provisions. For example, architects use building codes to verify the compliance of their architectural designs, whereas structural engineers regard building codes and standards as indices for safe designs. In addition, the rapid evolution of information technologies leads to confusion in selecting a software application most suitable for electronic codes. The variety of building code users and the numerous potential applications make it difficult to provide a comprehensive or integrated software development program for electronic codes in order to meet all their requirements.

To add to the complexity, the NBCC application varies from one region to another since it is modified to suit individual local requirements. Apart from the provincial and national differences, the NBCC document is required to have minor corrections every year [NBCC 1990]. Therefore, the building code information in electronic code applications must be updated to meet new requirements, as well as correct deficiencies and omissions. The other Canadian Codes, such as the National Fire Code of Canada, the Canadian Plumbing Code and the Canadian Farm Building Code, as well as provincial codes also encounter the same problems.

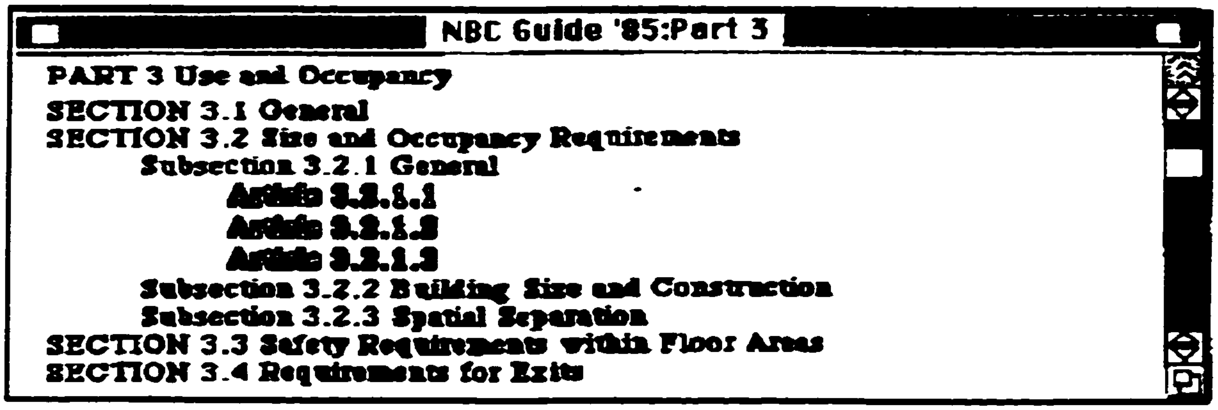
Another major obstacle to the production of workable electronic codes is the fact that the information content within the building codes is not classified or structured for electronic application, as presented in subsection 2.2. In most research prototypes, the information contained in building codes is formalized on the basis of the procedural logic

[Rosenman et al. 1986, De Waard 1992 , Frye et al. 1992] rather than the classification from the code itself. Although these prototypes provide users with access to electronic information, they are still limited in providing 'intelligent' electronic codes and standards [Vanier et al. 1994].

2.3.2 Electronic NBCC

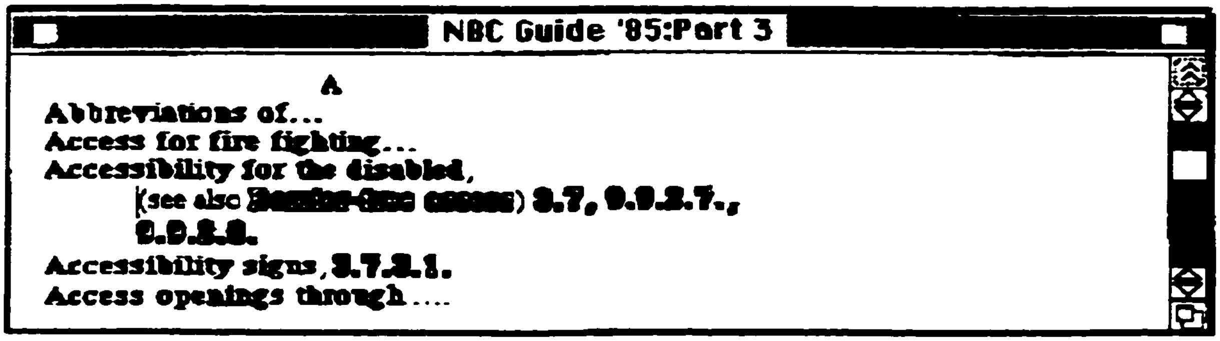
Electronic NBCC prototypes and projects have been developed at IRC since the mid-eighties. These electronic NBCC versions were aimed at enhancing the existing building code text with intelligent user interfaces and simplifying the contents using IT [Vanier et al. 1994].

The first prototype electronic version of the NBCC was the NBCC Retriever which was made available to 20 architects and engineers for evaluation [Vanier et al. 1994]. This electronic NBCC prototype is workable on both the Apple Macintosh and MS-DOS personal computers. It provided the functionality required to access the NBCC text quickly and efficiently, included a friendly user-interface, electronic tables of contents and checklists. This prototype was later revised and named the NBCC Guide' 85 with the same features as the previous version. The NBCC Guide' 85 ran only on the Macintosh environment using a commercial hypertext shell to store and browse the NBCC text. It provided three alternative access methods into the NBCC. First, the electronic table of contents which is a duplication of the information at the front of the NBCC allows the user to search for the desired code provisions. Figure 2.1 is an example of the electronic table of contents with Section 3.2 partially expanded. Clicking on the bold text reveals further information of the code text of interest, or clicking on the outlined text brings the user to that location within the NBCC document. The second access tool, namely hypertext index, in the NBCC Guide' 85 includes a list of subjects identical to the index at the back of the hard copy NBCC. The information can be accessed by browsing through the text, or using a keyword search in the index. The features of this access method are illustrated by Figure 2.2: by clicking on the bold line the user locates the desired Article, and clicking on the outlined text (e.g. Barrier-Free Access) moves the user to that NBCC provision.


NBC Guide '85:Part 3

PART 3 Use and Occupancy
SECTION 3.1 General
SECTION 3.2 Size and Occupancy Requirements
    Subsection 3.2.1 General
        Article 3.2.1.1
        Article 3.2.1.2
        Article 3.2.1.3
    Subsection 3.2.2 Building Size and Construction
    Subsection 3.2.3 Spatial Separation
SECTION 3.3 Safety Requirements within Floor Areas
SECTION 3.4 Requirements for Exits

Figure 2.1. **Example of Electronic Table of Contents** [Vanier et al. 1994]

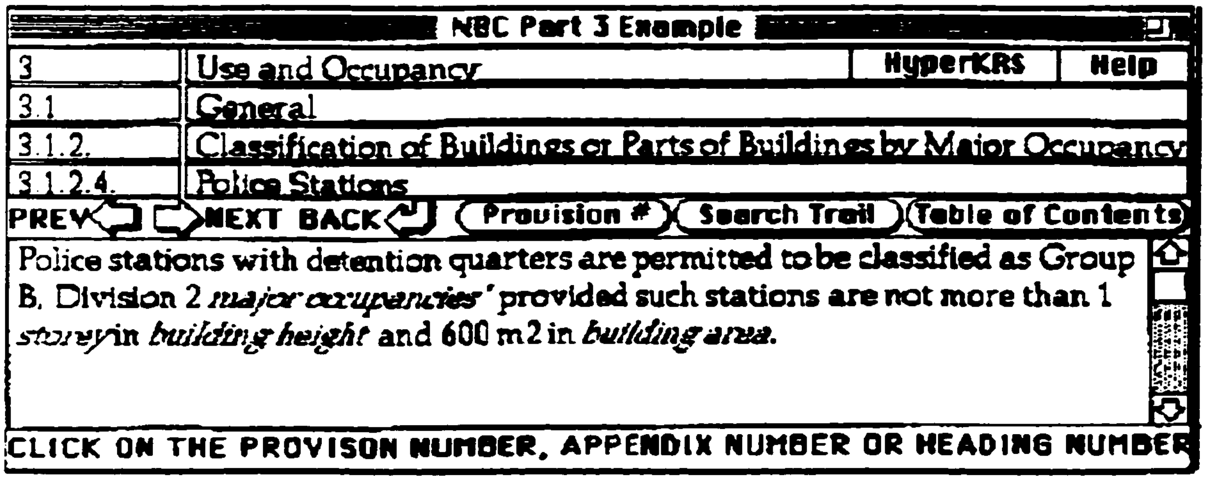

NBC Guide '85:Part 3

A
Abbreviations of...
Access for fire fighting...
Accessibility for the disabled,
    (see also Barrier-free access) 3.7, 9.9.2.7.,
    9.9.2.2.
Accessibility signs, 3.7.2.1.
Access openings through....

Figure 2.2. **Example of Hypertext Index** [Vanier et al. 1994]

The last alternative access tool of the NBCC Guide' 85 is the electronic checklist. This access method allows users such as building plan examiners and inspectors to work on a code compliance checklist to verify buildings for conformance with the regulation, as well as to access the relevant NBCC provisions. The detail of this access method is described in subsection 3.3.1.

Another prototype electronic version of the NBCC was NBCCard developed in the late 1980's. [Vanier 1991a]. This NBCC hypertext prototype was structured as a card-based sequential database in which each NBCC subdivision is represented by one card or database record. An example of a typical NBCCard Record is demonstrated in Figure 2.3 including the NBCC Article 3.1.2.4 Police Stations and its corresponding references such as Part, Section, and Subsection numbers. The NBCCard allows the user to browse and navigate within the NBCC document. Its functionality includes cross-reference linking, search trails, NBCC Definitions (Figure 2.4) and keyword searching (Figure 2.5). Although both the NBCC Guide'85 and the NBCCard are sequential hypertext applications, the database structure of the latter provides a more comprehensive and efficient access tool into the NBCC document.

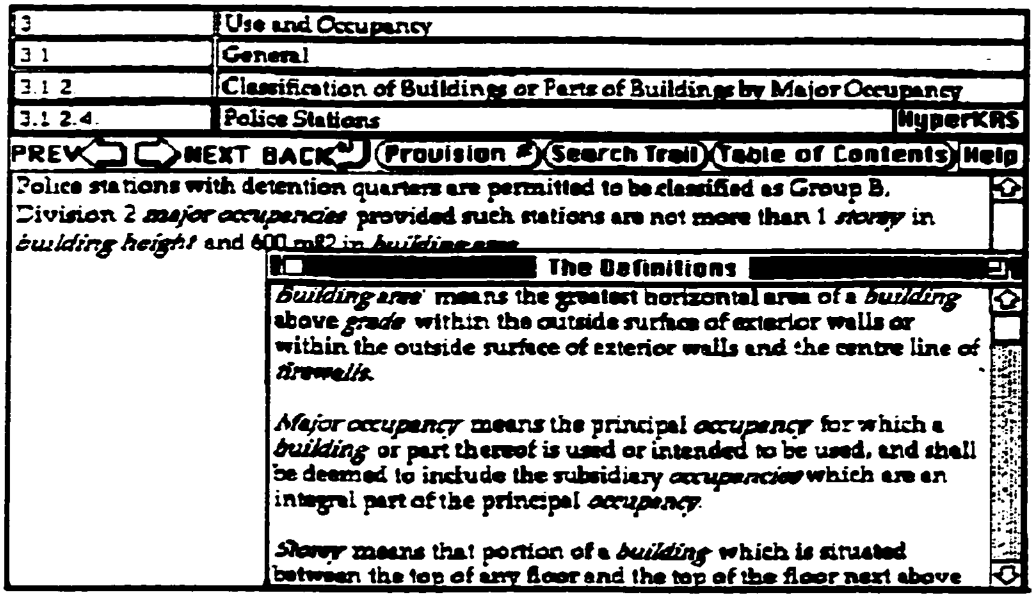


3 Use and Occupancy
3.1 General
3.1.2. Classification of Buildings or Parts of Buildings by Major Occupancy
3.1.2.4. Police Stations

Police stations with detention quarters are permitted to be classified as Group B, Division 2 *major occupancies*' provided such stations are not more than 1 *storey* in *building height* and 600 m2 in *building area*.

CLICK ON THE PROVISON NUMBER, APPENDIX NUMBER OR HEADING NUMBER

Figure 2.3. **A Typical NBCCard Record for NBCC Article 3.1.2.4** [Vanier et al. 1994]



3 Use and Occupancy
3.1 General
3.1.2. Classification of Buildings or Parts of Buildings by Major Occupancy
3.1.2.4. Police Stations

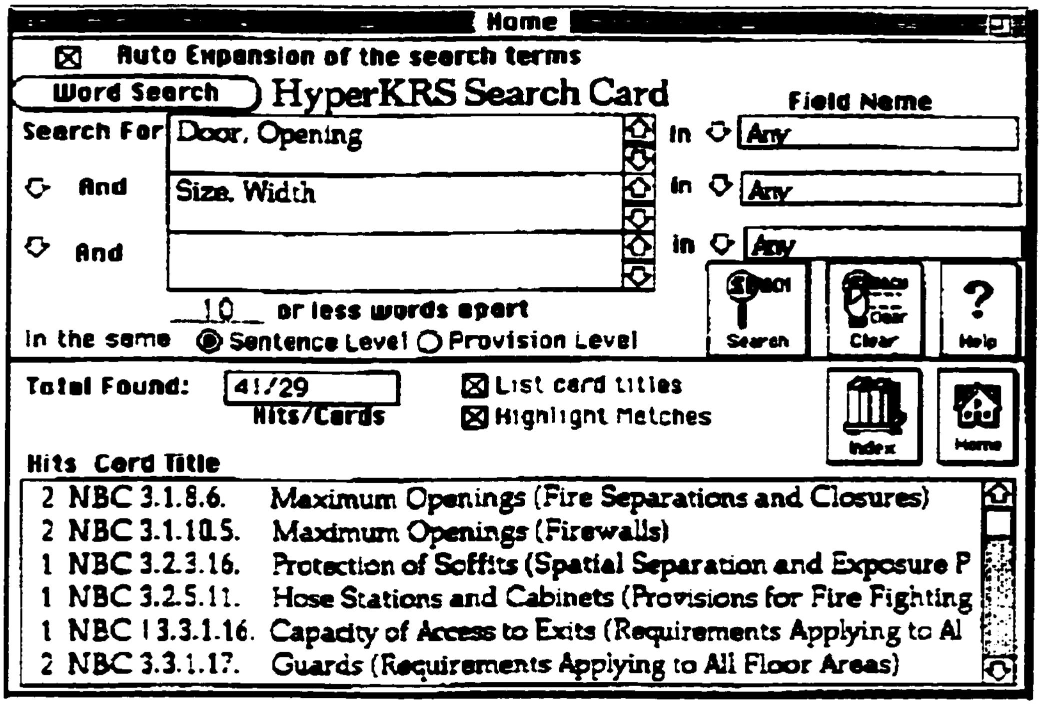
Police stations with detention quarters are permitted to be classified as Group B, Division 2 *major occupancies* provided such stations are not more than 1 *storey* in *building height* and 600 m82 in *building area*

Building area means the greatest horizontal area of a *building* above *grade* within the outside surface of exterior walls or within the outside surface of exterior walls and the centre line of *firewalls*.

Major occupancy means the principal *occupancy* for which a *building* or part thereof is used or intended to be used, and shall be deemed to include the subsidiary *occupancies* which are an integral part of the principal *occupancy*.

Storey means that portion of a *building* which is situated between the top of any floor and the top of the floor next above

Figure 2.4. **NBCC Article Record with Definition** [Vanier et al. 1994]



☒ Auto Expansion of the search terms

Search For: Door, Opening in Any
And: Size, Width in Any
And: in Any

10 or less words apart
In the same ● Sentence Level ○ Provision Level

Total Found: 41/29 Hits/Cards
☒ List card titles
☒ Highlight Matches

Hits	Card Title	
2	NBC 3.1.8.6.	Maximum Openings (Fire Separations and Closures)
2	NBC 3.1.10.5.	Maximum Openings (Firewalls)
1	NBC 3.2.3.16.	Protection of Soffits (Spatial Separation and Exposure P
1	NBC 3.2.5.11.	Hose Stations and Cabinets (Provisions for Fire Fighting
1	NBC 13.3.1.16.	Capacity of Access to Exits (Requirements Applying to Al
2	NBC 3.3.1.17.	Guards (Requirements Applying to All Floor Areas)

Figure 2.5. **NBCCard Keyword Searching Interface** [Vanier et al. 1994]

2.4 Building Inspection

Inspecting existing buildings for compliance with code requirements of the NBCC is generally difficult to perform for inexperienced or infrequent code users. The difficulty results mainly from the complexity of the code document containing numerous subdivisions and cross references which may cause confusion in interpreting and identifying texts. In effect, many building code users often find NBCC confusing and overwhelming because of the number of requirements which apply or seem to apply to a given building [Frye et al. 1992]. The major problems to overcome in the use of the hard copy edition of the building codes include finding the appropriate section, and correctly interpreting and understanding NBCC provisions [Alhussayni, Nguyen 1996].

Some research efforts to develop electronic codes to facilitate the inspection process have been paid significant attention. These efforts include the development of the electronic versions of NBCC such as NBCC Retriever, Guide' 85, and NBCCard, as presented in the previous subsection, in which code practitioners can search and retrieve with ease specific information within the document. In general, the positive features of these NBCC electronic versions are the hypertext links and friendly user interface. Despite the availability of these software products, most inspectors on site are rarely equipped with electronic codes and still use the hard copy of the NBCC when carrying out code compliance checking during building inspection [Nguyen 1996]. The latest electronic version of the NBCC is a CD-ROM (Compact Disk - Read Only Memory) developed by Institute for Research in Construction [IRC 1995]. This CD-ROM contains both 1990 and 1995 code documents. In this electronic NBCC, the user has access to all information required for compliance checking. Cross-referenced articles are easy to follow by simply clicking on them. The software also identifies related references in other code documents.

Although these electronic code products are useful to code practitioners in general, and building inspectors in particular, as browsing and searching tools, they fall short of providing automated verification for code compliance. As a result, a number of computer-based systems were initiated to address this deficiency. Some are presented in the following chapter.

Chapter 3

AUTOMATION OF CODE COMPLIANCE CHECKING PROCESS

3.1 Introduction

Checking a design or an existing building for conformance with applicable code requirements is a difficult and time-consuming task for inexperienced or infrequent users of building codes. The availability of information technologies such as hypertext, knowledge-based expert systems (KBES), intelligent systems are providing new environments as well as new approaches to automation of the code checking process.

This chapter is divided as follows: first, the background in developing an automated code checking system is presented, including difficulties in interpreting building codes, roles of case studies, the specific requirements for an automated code checking system, and IT applications to be described as possible alternatives; second an overview of various approaches to the field of building codes automation is summarized; and finally the automation approach selected is explained.

3.2 Background in Developing an Automated Code Checking System

3.2.1. Interpreting Building Codes

The building code document contains a large amount of complex regulations with numerous subsections, articles, sentences, and cross references to address various code requirements. Despite much effort of the code writers to organize the document in a clear fashion, the knowledge contained in building codes tends to be unstructured and difficult to interpret [Maher 1987, Kumar et al. 1995]. The difficulty may cause confusion in interpreting texts as well as identifying specific provisions from the general document [Frye et al. 1992].

To develop an automated code checking system, the regulatory requirements in the building codes are used as the main source of information to establish its knowledge base. However, code texts are written in a natural language format which is not able to be encoded directly to the computer-based system. Thus, the procedural logic of the code should be restructured and formalized into a suitable format for the implementation of the computer application. In the case of the NBCC, differences between national and provincial codes make it difficult to construct a comprehensive knowledge base to be valid for all code requirements. Expertise in the code, especially Part 3, has been slow to develop, though a lot of time, effort and money have been spent in trying to make the code document more understandable and easier to use [Frye et al. 1992].

3.2.2. Roles of Case Studies

Case studies, in the context of code checking, represent a piece of information related to various aspects of code violations found previously in existing buildings. Each of the cases describes building components non-compliant to specific regulatory requirements and proposes solutions for the violation problem. Roles of case studies are identified below.

Providing reminders to code practitioners in problem solving

Case studies can be incorporated into databases that are dynamically linked to the automated code checking system. During the checking process, cases related to a particular code provision may be called from the database for providing valuable reminders to past problems and solutions for code violations. In solving diagnostic problems in code checking of existing buildings, the information available from these case studies can be consulted for remedial measures for the code violation problems.

Enriching the knowledge base [Raphael et al. 1995]

Knowledge acquisition is considered to be a difficult task in the development of a knowledge-based system (to be presented at the end of this chapter). The difficulty is due in part to the fact that rules to represent the knowledge, which are precise and general enough is very difficult to obtain from an expert. It is, however, found that obtaining information from past relevant cases is easier than dealing with a current case. Case studies can be used as sources of rules. In effect, it is theoretically possible to elicit rules from cases by generalizing from the specific situations. For instance, using a large number of code conformance and violation cases, rules can be created about the features that make a compliance or non-compliance example.

The present study takes into account case studies which have been extracted from 1700 occurrences of code violations of existing buildings contained in the recent survey conducted by Public Works Canada, as mentioned in the beginning of Chapter 1. The survey represents a large amount of relevant information which may be extremely valuable for future inspections of other buildings These case studies can be incorporated into an automated code checking system as reference materials for consultation during building inspections. Such an incorporation is discussed in Chapter 5.

3.2.3. Specific Requirements for an Automated System

The automated system is designed to improve performance in building code compliance checking. The design objectives which were established to successfully develop such a system include efficiency, consistency, and convenience. The first objective translates into specific requirements as follows:

- Provide easily understood questions for eliciting information and templates for presenting compliance results.
- Operate fast and give reliable conclusions. To meet this requirement, the number of questions required for checking must be minimized, and all compliance categories extracted from the code document must be structured and formulated in a systematic fashion.
- Ensure automation in storing/updating building data, displaying check results, and generating reports.
- Provide access to relevant code provisions and related case studies.

Second, consistency in compliance checking as well as high quality results must be ensured. This is done through available rules representing the information of building codes. Finally, the system should be convenient to use, i.e. a friendly user interface is needed to satisfy the requirements of various code users. Another element of convenience is to ensure that the software product is a portable tool which enables the on-site inspector, while walking around the building, to carry out the building inspection process.

3.2.4 Information Technology Applications

Recently, the applications of advanced IT to the development of computer tools which assist in accessing, interpreting and applying regulatory information have been given particular attention [Kumar et al. 1995]. Several IT applications such as hypertext, expert systems, expertext and intelligent systems are briefly described below in an attempt to provide an overview of the possibilities in developing automated code checking systems.

Hypertext can be defined simply as the creation and representation of interlinked discrete piece of text [Kumar et al. 1995]. In hypertext systems, each discrete piece of text is represented by a node which is linked to other nodes allowing the user to navigate between nodes. According to [Yabuki and Law 1993], there are two types of links: navigation links and organization links. The former connects a document to other referencing documents and the latter connects the table of contents and indices of a book such as the NBCC to its subdivisions (Parts, Sections, Subsections, etc.). A navigation system systematically and efficiently provides access to information . [Kumar et al. 1995] summarized the advantages in using the hypertext approach for codes and standards processing as follows:

- It is suitable for representing codes and standards information.
- The user can access easily desired information without encountering unnecessary information.
- Cross-references are incorporated as 'hotspots' to enable the user to move to the relevant location simply by clicking on the reference.
- The code information contained in a hypertext environment can be easily updated or modified.
- Tables, graphs and figures can be attached to provisions texts.
- The hypertext system can be integrated with external processing software such as spreadsheets, thus assisting in design and conformance checking.

However, this approach still has disadvantages [Yabuki and Law 1993]: the hypertext version of design standards is limited by the development platforms (software and hardware) and the representation form of multiple nodes results in a complicated network.

'Expert systems' is defined as 'an intelligent interactive computer program that can play the role of a human expert by using heuristic knowledge or rules of thumb' [Adeli 1988]. It may function as a human expert or as a decision-maker assistant in solving problems or giving advice. With the assistance from the expert system, the level of performance of a novice decision maker may be raised to the level of an expert. This improvement in performance was illustrated in Figure 3.1 [Vadas 1992].

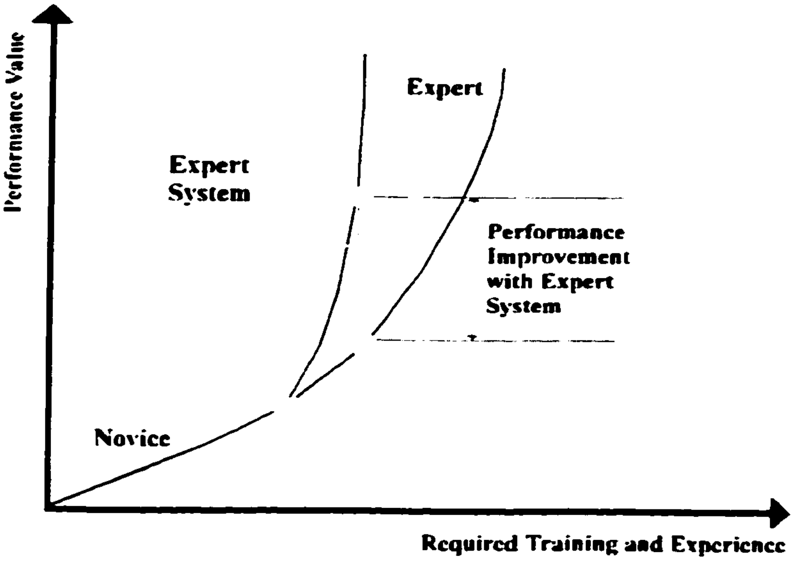


Figure 3.1. **Performance Improvement with Expert System** [Vadas 1992]

The differences between conventional computer programs and expert systems are summarized in Table 3.1. Many of the expert system applications to building codes were developed around the world to date for the construction industry [Rosenman and Gero 1985, Dym et al. 1988, Bedard and Ravi 1991, Sharpe 1991, Moulin 1992, Frye et al. 1992, Heikkila and Blewett 1992, Delis and Delis 1995]. Some of these research are described in the subsection 3.3.

'Expertext' represents a combination of expert systems and hypertext [Casson and Stone 1992]. This approach combines the best features of expert systems and hypertext to develop computer-based systems having the semantically rich nodes of hypertext and the well-specified, computable links of expert systems. [Vanier et al. 1994] stated that expert systems require too much work to develop, whereas hypertext systems fall short of providing an 'intelligent' interface to assist in searching for building codes and standards.

The requirement for an efficient control mechanism may be satisfied in a system integrating both hypertext and expert systems.

Table 3.1. **Characteristics of Conventional Programs and Expert Systems** [Allen 1992]

Conventional Programs	Expert Systems
- Representation and use of data	- Representation and use of knowledge
- Knowledge and control integrated	- Knowledge and control separated
- Algorithmic (repetitive) process	- Heuristic (inferential) process
- Effective manipulation of large database	- Effective manipulation of large knowledge base
- Programmer must ensure uniqueness and completeness	- Knowledge engineer inevitably relaxes uniqueness and completeness restraint
- Mid-run explanation impossible	- Mid-run explanation desirable and achievable
- Oriented toward numerical processing	- Oriented toward symbolic processing

'Intelligent systems' is a term used in this thesis as a combination of computer technologies including expert systems, hypertext and databases. Chapter 4 is dedicated to describing in detail this system.

3.3 Approaches to Automated Code Checking

This section presents the review of implementations regarding automated code checking systems, which is intended to identify the representative cases of what has been done around the world. This literature survey will hopefully lead to the selection of an appropriate approach. The research implementations to be described can be classified into three groups: Hypertext Systems, Expert Systems, and Integrated Systems.

3.3.1 Hypertext System Applications

Hypertext systems with a full text database can serve as replacements for hard copy publications. As described in subsection 3.2.4, this approach provides a new software paradigm for managing text-based information such as building codes. In addition to the NBCC electronic versions developed by using the hypertext approach, as presented in subsection 2.3.2, a number of projects were initiated to address alternative approaches to providing more 'intelligent' access to building codes.

The International Organization for Standardization [ISO 1986] described the Standard Generalized Markup Language (SGML) as one of the major international standards for the interchange of text-based information. It is a generalized markup language for encoding document structure information as part of the document, and for tagging the structural elements of documents. The Institute of Research for Construction has adopted this approach to encode the NBCC documents, in which SGML is used to denote special notation for the various subdivisions such as Parts, Sections, and Subsections, or for special annotation of cross-references, glossary, or dimension descriptors. The following example demonstrates how an NBCC Article (e.g. Article 3.1.2.4) could appear in an SGML application:

> <NBCC art> NBCC num> **3.1.2.4.** </ NBCC num><NBCC art head> **Police Stations** </ NBCC art head>
> <art body> **Police stations with detention quarters are permitted to be classified as Group B, Division 2**< NBCC def > **major occupancies** </ NBCC def > **provided such stations are not more than 1** < NBCC def >

Storey </ NBCC def > **in** < NBCC def > **building height** </ NBCC def ></ art body ></ NBCC art >

In the above example, the start and the end of the structural element are indicated by angle brackets (< --- >) and (</ --- >) respectively. This SGML demonstration consists of two portions: the first identifies the 'NBCC Article 3.1.2.4' with Heading 'Police Stations' and the second contains the body of the NBCC Article. In addition, the < NBCC def > in the second portion refers to the NBCC Definitions. An IT tool using an SGML shell, named MiniCode, has been developed at the IRC [Vanier et al. 1994] to assist the design community in performing building code compliance checking [Thomas et al. 1992]. The characteristics of the MiniCode system are summarized as follows [Vanier et al. 1994]:

- Assist the user in classifying the type of building.
- Identify the NBCC provisions relevant to the user's classification selection, while excluding those that do not pertain to the specific building projects.
- Provide a Microsoft Window Help interface to the full-text of the NBCC, including hypertext cross-references, history trails and interactive NBCC definitions.

The MiniCode system initialized the idea of intelligent searching in electronic codes. However, it is not specific enough to identify all detailed information contained in the NBCC document. This deficiency lead to the development of a Designer's MiniCode system, that is a logical extension of the MiniCode Generator [Vanier et al. 1994] with a number of enhancements such as additional classification trees and building attributes.

Another IT tool to support building code users that has been developed by IRC using the SGML-encoded documents in a hypertext environment is CD-ROM Based Code [CD-ROM 1993, IRC 1995]. This product allows users to search for full-text of all documents on the CD-ROM, including tables and equations. In addition, the other features of the CD-ROM Based Code such as interactive side-by-side tables of contents and full-text, dynamic cross-references, indexed searching and interactive NBCC definitions are also integrated into the IT tool.

3.3.2 Expert System Applications

[Rosenman and Gero 1985] were among the first research to explore expert systems technology to represent building codes. According to the authors, the user can 'communicate' with the system by answering questions or replying with 'how' (how do I answer this?) or 'why' (why do you want to know?). The reply 'why' prompts the system to provide an answer in terms of rules. An example of a rule is:

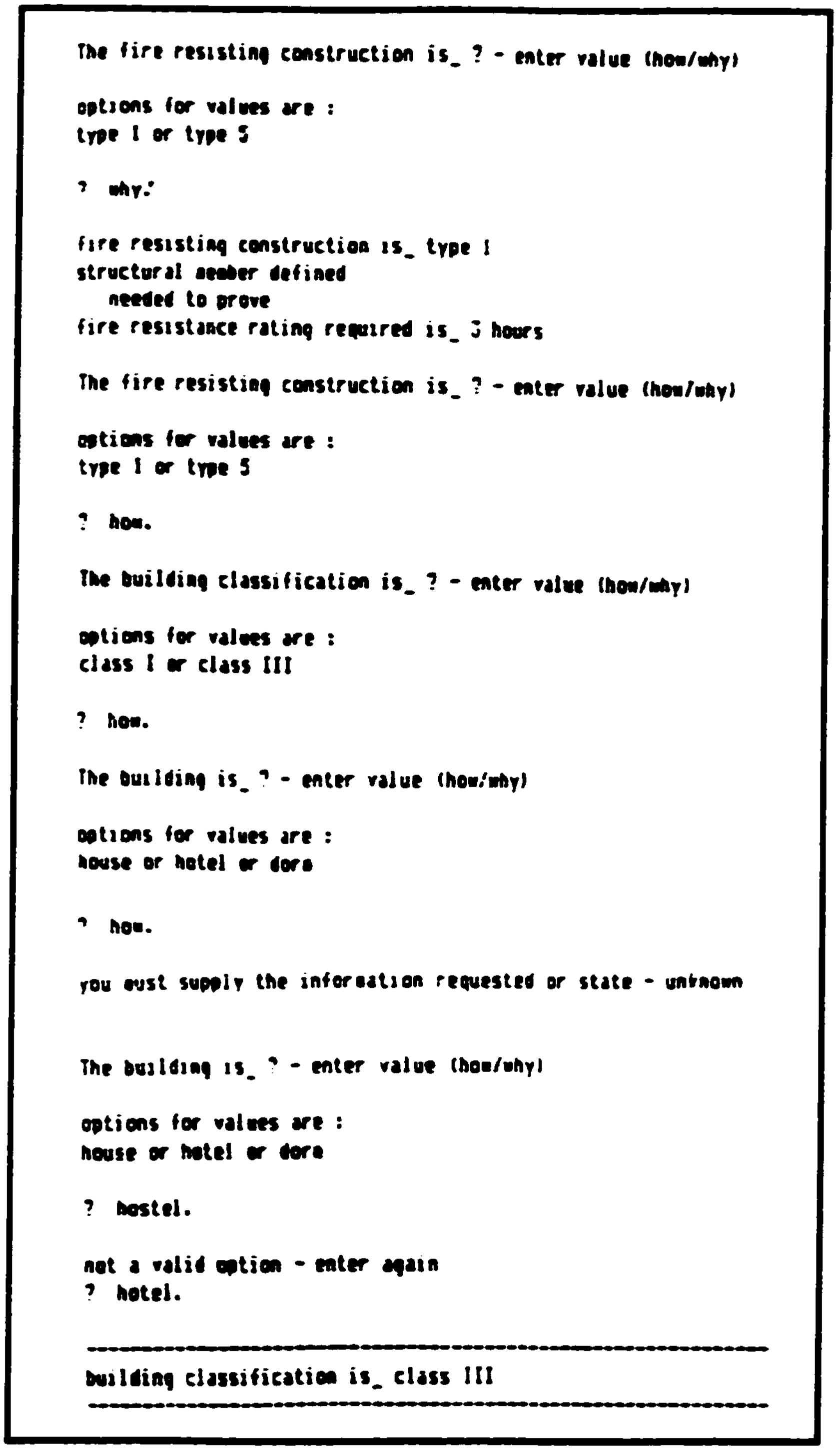
If 'the building' is_a hotel or dormitory
Then classification is 'Class III'

The reply 'how' directs the system to search for any knowledge it may have to satisfy this request. The search procedure is made more efficient by the system's indexing of the rules with regards to the objects. Figure 3.2 is a sample dialogue between the user and the system from the initial command to the first conclusion reached. The conclusions of the research indicate that expert systems could be a suitable tool for providing a variety of users the means of accessing the knowledge contained in building codes in an efficient, complete and consistent manner. The authors also emphasized the enormous amount and complexity of information in the building code document.

[Frye et al. 1992] developed an expert system for the Fire Protection Requirements of the National Building Code of Canada. The initial knowledge base of the system is limited to Section 3.2 - Size and Occupancy Requirements for Fire Safety of the NBCC-

Part 3. The expert system provides the user with a simple query mode to input the information required for compliance checking. Rules are used to represent the input information based on the structure of decision trees for each independent protection topic (Figure 3.3). Once the input data have been completed and their pre-conditions to the rules have been satisfied, these rules will fire to give a compliance result with absolute certainty. Furthermore, a <HELP> option is provided in case that assistance in responding to a prompt is required. Both text and graphic modules are available in most of the HELP screens.

Some researchers have evaluated expert systems as available tools for automating building design process [Rosenman et al. 1986] in which compliance code checking is also integrated [Bedard and Gowri 1990, Bedard and Ravi 1991, Kumar and Topping 1988]. Others [Fazio et al. 1989] found that this approach does not provide a complete remedy for the modeling of building codes due to the complexity in the representation of knowledge contained in such a large regulatory document. However, [Fazio et al. 1989] believed that the combination of frame-based and rule-based systems may be a desirable and helpful approach to engineering design. A survey on expert systems for structural design codes by [Kumar and Topping 1988] indicates that it is easy and efficient to represent the building code information and the encoded standards information could be used to facilitate the design of structural members as well as to provide solutions for design problems.



```
The fire resisting construction is_ ? - enter value (how/why)

options for values are :
type 1 or type 5

? why.

fire resisting construction is_ type 1
structural member defined
   needed to prove
fire resistance rating required is_ 3 hours

The fire resisting construction is_ ? - enter value (how/why)

options for values are :
type 1 or type 5

? how.

The building classification is_ ? - enter value (how/why)

options for values are :
class I or class III

? how.

The building is_ ? - enter value (how/why)

options for values are :
house or hotel or dorm

? how.

you must supply the information requested or state - unknown


The building is_ ? - enter value (how/why)

options for values are :
house or hotel or dorm

? hostel.

not a valid option - enter again
? hotel.

-------------------------------------------------------
building classification is_ class III
-------------------------------------------------------
```

Figure 3.2. **Sample Dialogue Between User and System From Start to First Conclusion** [Rosenman and Gero 1985]

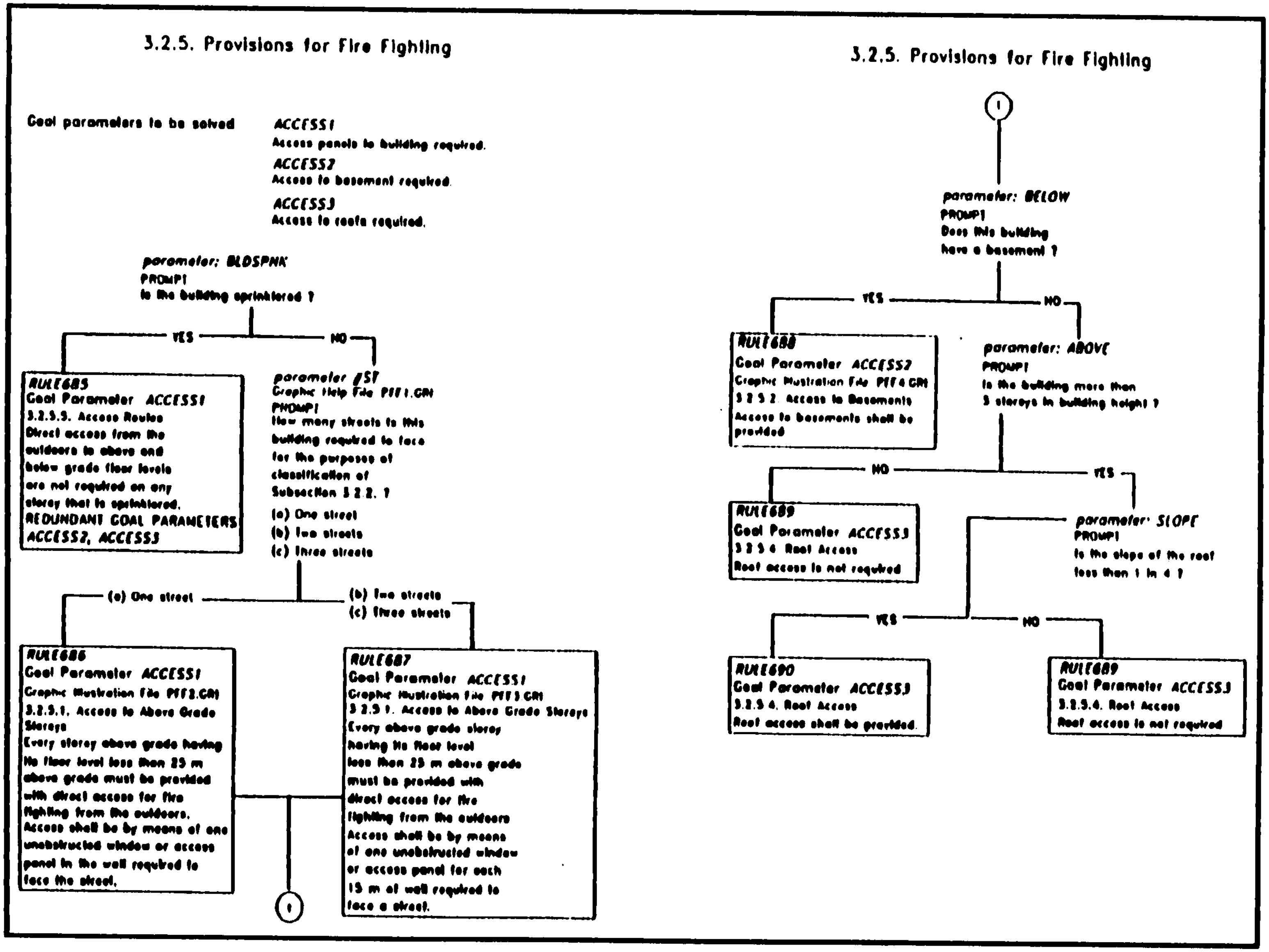


Figure 3.3. **Decision Tree, Provisions for Fire Fighting** [Frye et al. 1992]

A commercial expert system product currently available is BCAider [Sharpe 1991, Blackmore et al. 1994]. This is a computerized version of the Australian Building Code using an expert system to represent the information knowledge. The BCAider user can access the interested code provisions by entering the values for the desired category of building, and then investigate detailed information by answering TRUE/FALSE questions to identify the conditions under which the provision is applicable. In spite of its limitation in Australian building plan examination, BCAider demonstrates the capabilities of expert systems in assisting building code users.

3.3.3 Integrated System Applications

Some researchers have developed comprehensive systems for the domain of automated code checking that integrate codes and standards information in building design systems.

[Dym et al. 1988] described an expert system embedded in a Computer-Aided Design and Drafting (CADD) system to assist in automating architectural design process. The prototype system, named Life Safety Code (LSC) Advisor, links the CADD system to a type of building code to allow users to review CADD drawings and ensure compliance to the LSC of the National Fire Prevention Association. The authors stress the rationale for rule-based and frame-based representations used in LSC Advisor. Rule-based representations are used to encode the text of the LSC, whereas frame-based representations are used to represent floor plans of building. LSC Advisor users input a file (knowledge base) that contains information describing a floor of a building. The CADD information is captured and verified for code conformance. If a fire code requirement is not satisfied, LSC Advisor provides the information extracted from the rule base. Figure 3.4 is an example of the primary output of the LSC Advisor system.

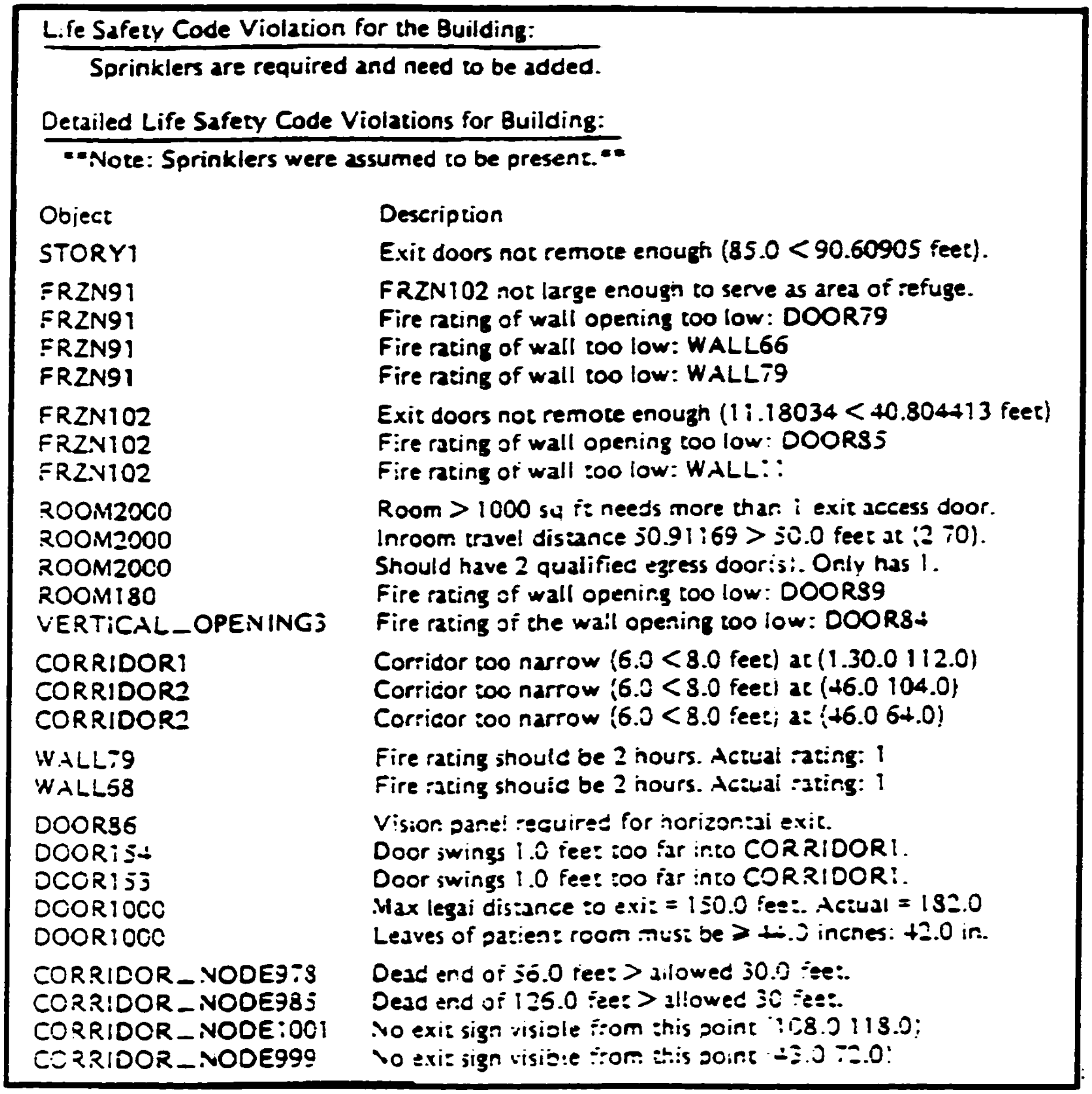
An extension to the LSC Advisor is Fire Code Analyzer (FCA) developed by [Delis and Delis 1995]. This is an automatic Fire-Code checking System using expert system technology. 'The system consists of a rule-based system that encapsulates the various code requirements, a frame-based system for building representations, and a set of geometric algorithms' to facilitate knowledge inference. The FCA is used to assist in the review of building designs. The enhancements of FCA include the following:

- Provide a more versatile frame-based representation of floor plans.
- Create a more elaborate rule-based system to handle three dimensional issues.
- Utilize a set of geometric algorithms to facilitate knowledge inference.
- Provide a more user-friendly interface.

Figure 3.5 shows the elements of the interface: the upper-left grid is used to review architectural layouts, the middle panel allows users to input graphical information of building drawings, and the upper-right window provides access to the rules representing the LSC (e.g. the paragraph 12-1.6.2.c of the LSC are displayed in screen). These rules fire when all preconditions are satisfied.

The authors also indicated some limitations of the FCA prototype:

- Only architectural layouts from the domain of hospital buildings are examined.
- Interface with a CAD system is not available.
- Only a fraction of the LSC is encoded in the system.

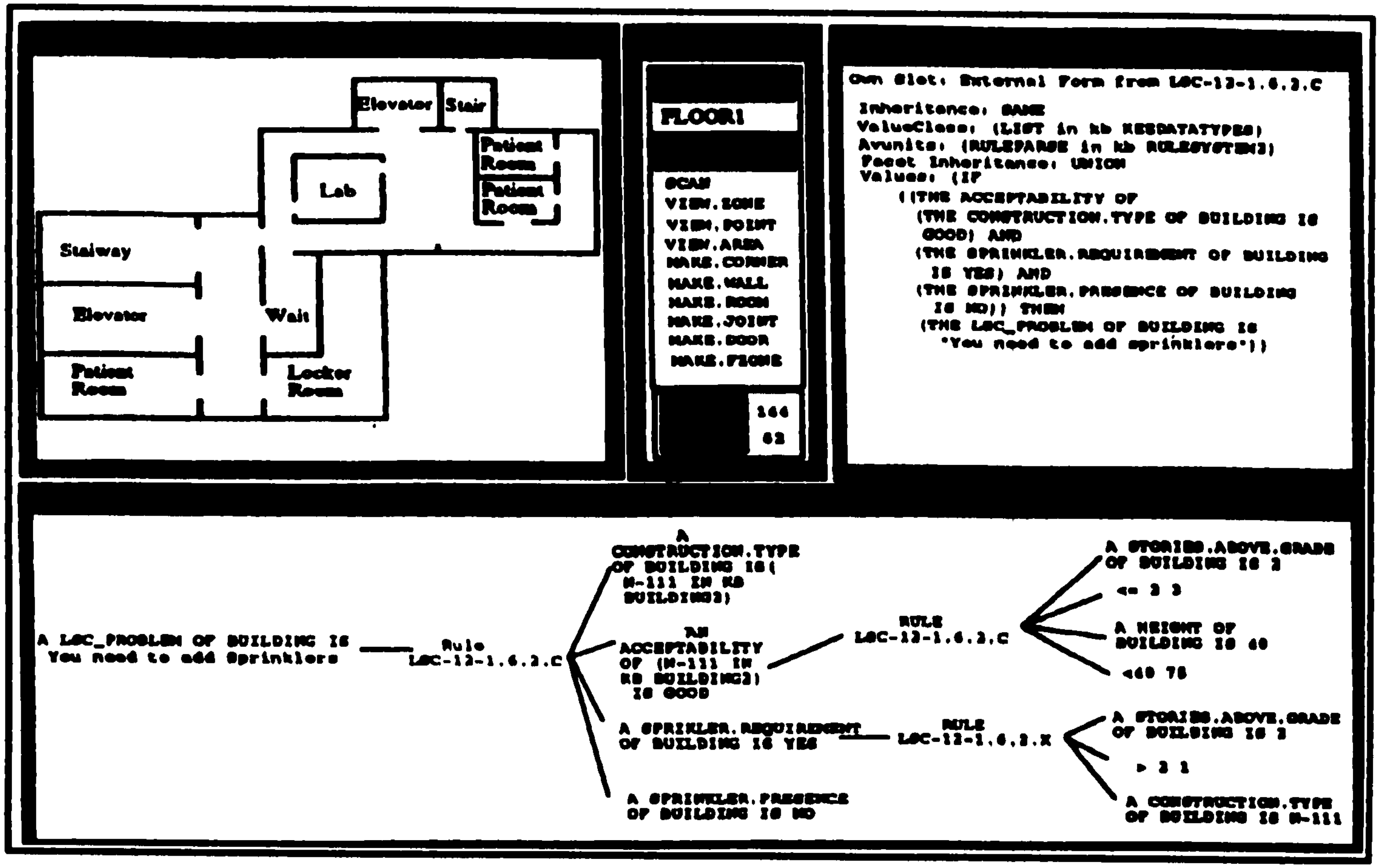


```
Life Safety Code Violation for the Building:
    Sprinklers are required and need to be added.

Detailed Life Safety Code Violations for Building:
  **Note: Sprinklers were assumed to be present.**

Object                  Description
STORY1                  Exit doors not remote enough (85.0 < 90.60905 feet).
FRZN91                  FRZN102 not large enough to serve as area of refuge.
FRZN91                  Fire rating of wall opening too low: DOOR79
FRZN91                  Fire rating of wall too low: WALL66
FRZN91                  Fire rating of wall too low: WALL79
FRZN102                 Exit doors not remote enough (11.18034 < 40.804413 feet)
FRZN102                 Fire rating of wall opening too low: DOOR85
FRZN102                 Fire rating of wall too low: WALL[illegible]
ROOM2000                Room > 1000 sq ft needs more than 1 exit access door.
ROOM2000                Inroom travel distance 50.91169 > 50.0 feet at (2 70).
ROOM2000                Should have 2 qualified egress door(s). Only has 1.
ROOM180                 Fire rating of wall opening too low: DOOR89
VERTICAL_OPENING3       Fire rating of the wall opening too low: DOOR84
CORRIDOR1               Corridor too narrow (6.0 < 8.0 feet) at (1.30.0 112.0)
CORRIDOR2               Corridor too narrow (6.0 < 8.0 feet) at (46.0 104.0)
CORRIDOR2               Corridor too narrow (6.0 < 8.0 feet) at (46.0 64.0)
WALL79                  Fire rating should be 2 hours. Actual rating: 1
WALL68                  Fire rating should be 2 hours. Actual rating: 1
DOOR86                  Vision panel required for horizontal exit.
DOOR154                 Door swings 1.0 feet too far into CORRIDOR1.
DOOR153                 Door swings 1.0 feet too far into CORRIDOR1.
DOOR1000                Max legal distance to exit = 150.0 feet. Actual = 182.0
DOOR1000                Leaves of patient room must be ≥ 44.0 inches: 42.0 in.
CORRIDOR_NODE978        Dead end of 56.0 feet > allowed 30.0 feet.
CORRIDOR_NODE985        Dead end of 126.0 feet > allowed 30 feet.
CORRIDOR_NODE1001       No exit sign visible from this point (108.0 118.0)
CORRIDOR_NODE999        No exit sign visible from this point (43.0 72.0)
```

Figure 3.4. **Typical Floor Plan Conformance Report** [Dym et al. 1988]



```
Own Slot: External Form from LSC-12-1.6.2.C
 Inheritance: SAME
 ValueClass: (LIST in kb KESDATATYPES)
 Avunits: (RULEPARSE in kb RULESYSTEM2)
 Facet Inheritance: UNION
 Values: (IF
     ((THE ACCEPTABILITY OF
       (THE CONSTRUCTION.TYPE OF BUILDING IS
        GOOD) AND
       (THE SPRINKLER.REQUIREMENT OF BUILDING
        IS YES) AND
       (THE SPRINKLER.PRESENCE OF BUILDING
        IS NO)) THEN
      (THE LSC_PROBLEM OF BUILDING IS
        "You need to add sprinklers"))
```

Figure 3.5. **FCA Analyzer Interface** [Delis and Delis 1995]

[Garrett et al. 1995] investigate a computer-based model integrating a standards processing system named SSE with a building design system i.e. SEED (Software Environment to support the Early phases in building Design) which is based on a division of the preliminary design process into phases, each of which addresses a specific task [Kilicotte et al. 1995]. The SSE represents a Standard Support Environment for processing building standards during the stages of design supported in the SEED system. The overall architecture of the SSE consists of two major components: the Standard Usage Module (SUM) and the Standard Processing Framework (SPF). The SUM that encapsulates the functionality of the SPF is used to support communication, data exchange and constraint management between the SEED modules and the SPF. The SPF is a multi-module framework designed for representing and processing standards. The authors stress the complexity of the dynamic nature of building design standards and indicate that the integrated model is a viable approach to providing support for both standard compliance checking and compliant generation.

3.4. The Approach Selected

3.4.1. Proposed Methodology

The review of research on automated code compliance checking demonstrates that expert systems are capable of representing and reasoning with specialist knowledge to diagnose problems with code conformance checking, whereas databases and hypertext systems are efficient for handling cross references among distinct building subsystems and disciplinary viewpoints in data management systems. The proposed methodology is that of an intelligent system combining these computer technologies. Such a combination is believed to meet the specific requirements for an automated code compliance checking system and to overcome the problematic characteristics of the information contained in the building code document, as presented at the beginning of this chapter.

To develop such an automated system, the process of mapping the knowledge available in building codes into the system's knowledge base, known as knowledge engineering, must be carried out. The development of knowledge acquisition for building codes is also required. The knowledge acquisition process basically consists of three steps:

First, code documents are classified into specific categories to be considered during the building compliance checking. As described in subsection 2.2.2, the NBCC document is divided into multi subdivisions such as Parts, Sections, Subsections, Articles, etc.; the information under this hierarchical structure can not be incorporated directly into the computer-based system. It must be reorganized in such a way that all compliance topics contained in building codes can be verified in a systematic fashion. The list, as shown below, represents all compliance categories to be checked. Each category may be divided into several compliance subcategories, depending on complexity and nature. The hierarchical structure of these categories has been extracted from the information available in Part 3 of the NBCC. (Details of these compliance categories and their corresponding subcategories are presented in Table 5.1 in subsection 5.3.1)

1. Fire Detection and Alarm Systems
2. Fire Separations
3. Fire Escapes
4. Structural Fire Protection
5. Exit Requirements
6. Means of Egress
7. Interior Finish and Insulation Protection

8. Storage and Repair Garages
9. Emergency Lighting
10. Stair Requirements
11. Service Facilities
12. High Building Requirements
13. Miscellaneous Requirements

Second, based on the procedural logic for each specific compliance category, decision trees are developed in such a manner that the data required for checking is systematically collected whilst unnecessary queries are avoided. The major problem appears to be that the existing information contained in the NBCC is not always explicit; thus, it can not be interpreted in a proper fashion without using appropriate analysis capability. The interpretation of some code provisions can be a difficult task because of multiple cross references and the qualitative nature of information.

The final step in the knowledge acquisition process is to structure the decision trees in a format that can be encoded into the knowledge based expert system. The development of these formats requires suitable knowledge representation forms. Rule-based and frame-based approaches are the two techniques to be selected for representing the knowledge. They both are presented in the following chapter.

The three stages to develop the knowledge acquisition for building codes are demonstrated by the following figure for the case of Article 3.4.2.3. Distance between Exits. (Decision trees for other code requirements of Part 3 are also available in Appendix)

Code Text

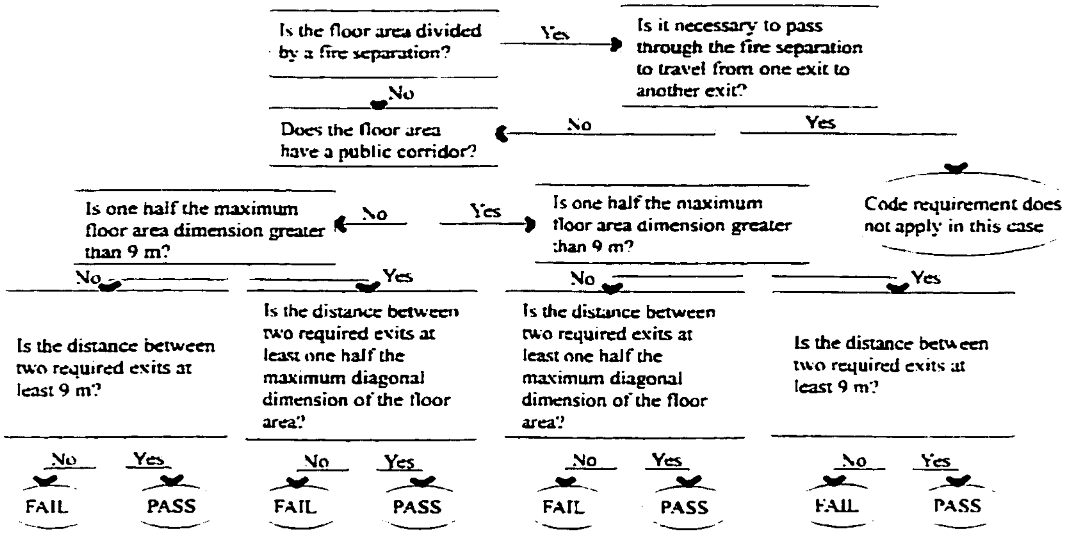
3.4.2.3. Distance between Exits

1) Except where a floor area is divided by a fire separation so that it is necessary to pass through it to travel from one exit to another exit, the least distance between 2 required exits from a floor area shall be

a) one half the maximum diagonal dimension of the floor area, but need not be more than 9m for a floor area having a public corridor, or

b) one half the maximum diagonal dimension of the floor area, but not less than 9m for all other floor areas. (See Appendix A)

Decision Tree



Rule-Base

```
IF  Floor divided by a fire separation IS Yes = TRUE  AND Pass through FireSeparation IS YES THEN
    CheckResult := " Not Applicable"
ELSE
   BEGIN
   IF  Floor has a public corridor IS Yes = TRUE THEN
       IF HalfMaxDimension of FloorArea > 9 THEN
          IF Distance between 2 exits >= HalfMaxDimen THEN
             CheckResult := "PASS"
          ELSE
             CheckResult := "FAIL"
       ELSE
          IF Distance between 2 exits >= 9 THEN
             CheckResult := "PASS"
          ELSE
             CheckResult := "FAIL"
   END
```

Figure 3.7. **Example of Knowledge Acquisition Process for the Least Distance between Exits**

3.4.2. Scope and Limitations

The proposed approach aims at interpreting, formulating, and structuring the knowledge contained in building codes into the knowledge-based expert system in order to create a practical computer-based tool that can assist an inspector in verifying existing buildings for code compliance in a consistent and efficient fashion.

The present research intends to develop a systematic approach to automating the code compliance checking process in building inspection. The compliance categories to be checked are extracted from the information available in the Part 3 of the NBCC addressing Fire Safety, Exit Requirements, Emergency Lighting, etc. However, the proposed methodology can be applicable to other building code documents using the same approach as described.

HASES is a prototype implementation of the proposed methodology for inspecting existing buildings for code conformance. HASES provides a user-friendly interface using query languages to allow users to easily input the data required for the building inspection process, to generate reports, and to access building code texts as well as relevant case studies. The two characteristic features of the proposed methodology is to minimize the number of questions needed for compliance checking and to supply relevant case studies for consultation on suitable solutions for problems in code violations.

The HASES knowledge base is developed from the information contained in Part 3 of the NBCC alone. The present work focuses only on two Building Occupancy Classifications: Group D (Business and personal services) and Group F - Division 3 (Low hazard industrial occupancies). Although the knowledge base of the prototype system actually represents all the information required for verifying the compliance of the code provisions in Part 3, it is certainly not complete and needs more knowledge acquisition to account for provisions addressing buildings classified in other occupancy groups, which are not incorporated in the system.

Chapter 4

INTELLIGENT SYSTEMS FOR AUTOMATED CODE CHECKING

4.1 Introduction

The building inspection process is a complex activity that requires various information for diagnosing problems with code compliance checking. The proposed approach combines expert systems, databases and hypertext techniques to provide an automatic system feasible for facilitating such a process. In effect, this approach benefits from specialist knowledge in the form of an expert system capable of giving advice with diagnostic problems in code checking, whereas database and hypertext techniques are exploited for their ability to handle efficiently cross references and multiple viewpoints in data management systems. This chapter presents in detail the main components of the intelligent system and suitable representation techniques to be employed for describing the building objects as well as encapsulating the knowledge contained in building codes.

4.2 Basic Components of the Intelligent System

The intelligent system's architecture basically consists of two components: an expert system and a data management module (see Figure 4.1). This section presents the general architecture of the expert system in an attempt to briefly describe the functionality of its three main facilities and the reader is referred to books such as [Maher 1987, Harmon and Sawyer 1990, Allen 1992] for a more extensive account of expert systems.

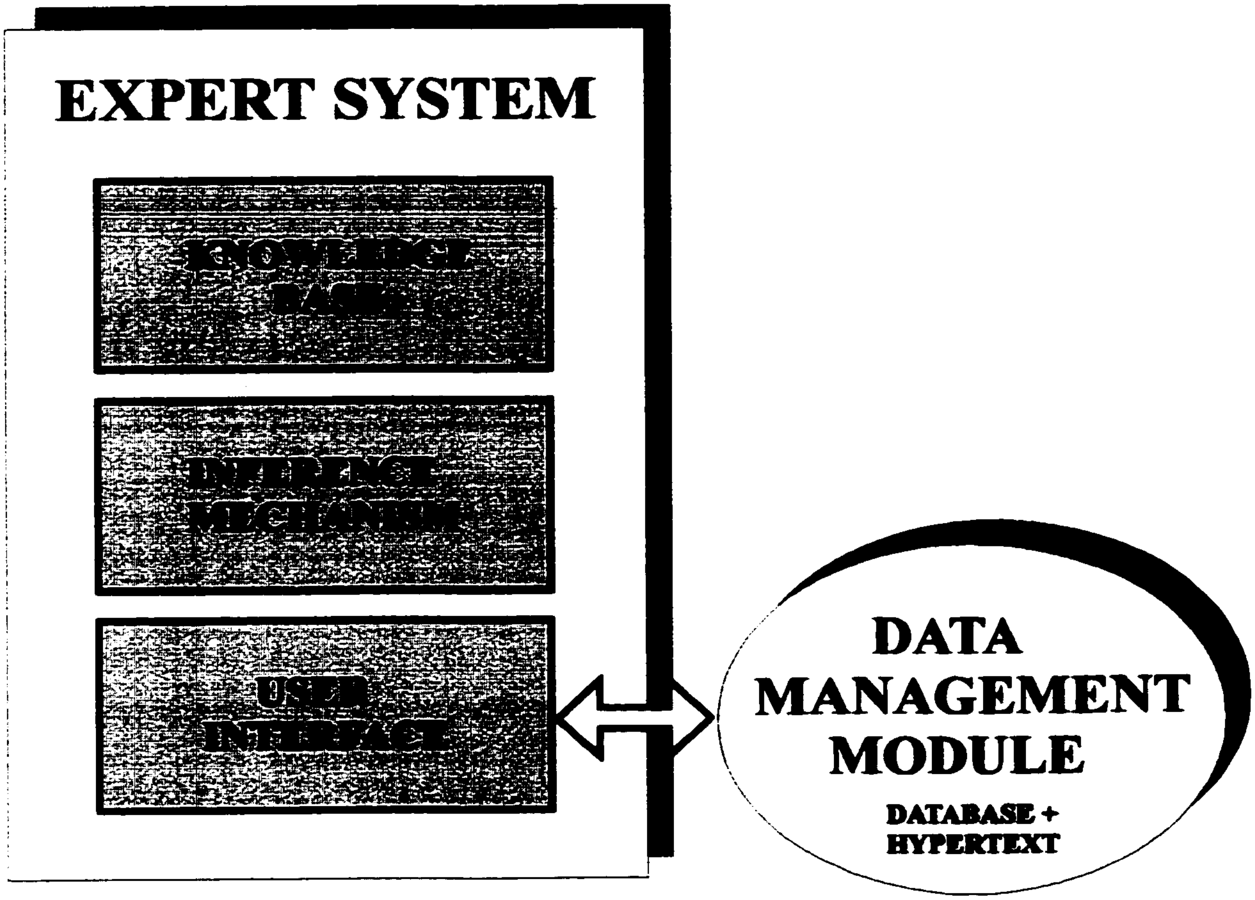


Figure 4.1. **Basic Components of An Intelligent System**

4.2.1. The Expert System

The main characteristics of expert systems is a separation of domain knowledge, control knowledge, and knowledge about the specific problem currently being solved

[Allen 1992]. This leads to the identification of three basic components of an expert system: the knowledge base, the inference mechanism, and the user interface.

The knowledge base contains the facts and the heuristics associated with the domain of building codes [Frye et al. 1992]. One example of a fact, or object, is a floor. A floor could be represented in the knowledge base as a compliance item to be checked with attributes including "floor area", "number of exits", "divided by a fire separation", etc. Heuristics in building code information are typically represented by rules. It is required that the knowledge base be transparent enough so that it can be modified or expanded since the knowledge in a building code such as the NBCC is continually changing and updating. This statement is supported by the general description of building codes in subsection 2.2.1. Rule-based and frame-based techniques are the representation forms suitable for making this knowledge as transparent as possible. These representations are presented in the following section of this chapter.

The second component of the expert system is the inference mechanism. It controls the reasoning process. An example of the inference mechanism to determine if the building requires the installation of a fire alarm system is one in which the expert system for a given occupancy classification group (e.g. F-Division 3) reasons back to known facts (e.g. Does the building contain a contained use area? An impeded egress zone? An interconnected floor space? More than 3 storeys? - as required in NBCC Sentence 3.2.4.1.(1)) to verify whether the building should be provided with a fire alarm system (see Figure 4.2). This approach is referred to as backward chaining in which the system assumes a hypothesis and reasons back to known data or facts to support or discount the assumed hypothesis [Harmon and Sawyer 1990].

Another control strategy is forward chaining. A system uses this strategy " if it works from an initial state of known facts to a goal state (conclusion(s)) " [Maher 1987]. Figure 4.3 shows an example of forward chaining to be used for determining the required exit width found in NBCC Article 3.4.3.1. Exit Width. Several hypotheses for the exit types (e.g. corridors, passageways, stairs, ramps, or doorways) are given. Using the data input from the user, the system verifies the actual condition of the exit and determines the minimum required exit width.

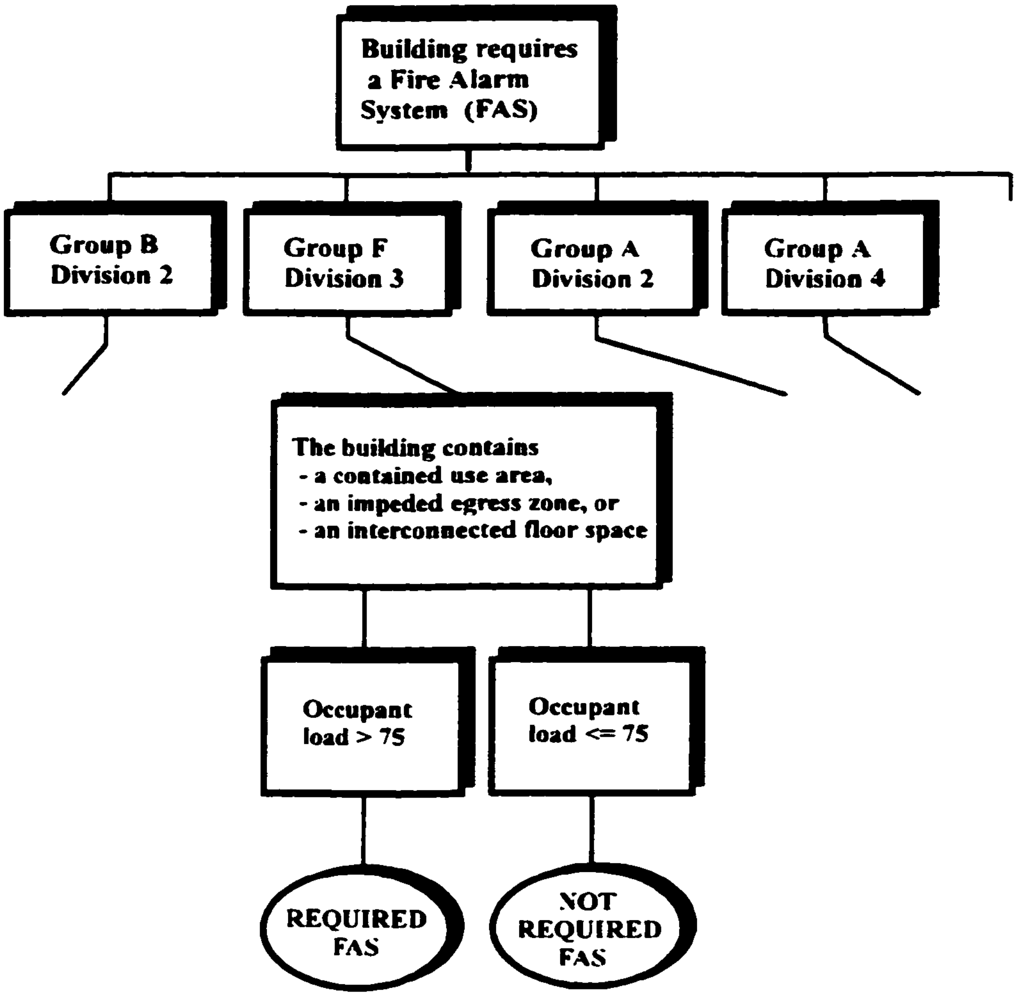


Figure 4.2. **An Example of Backward Chaining Inference**

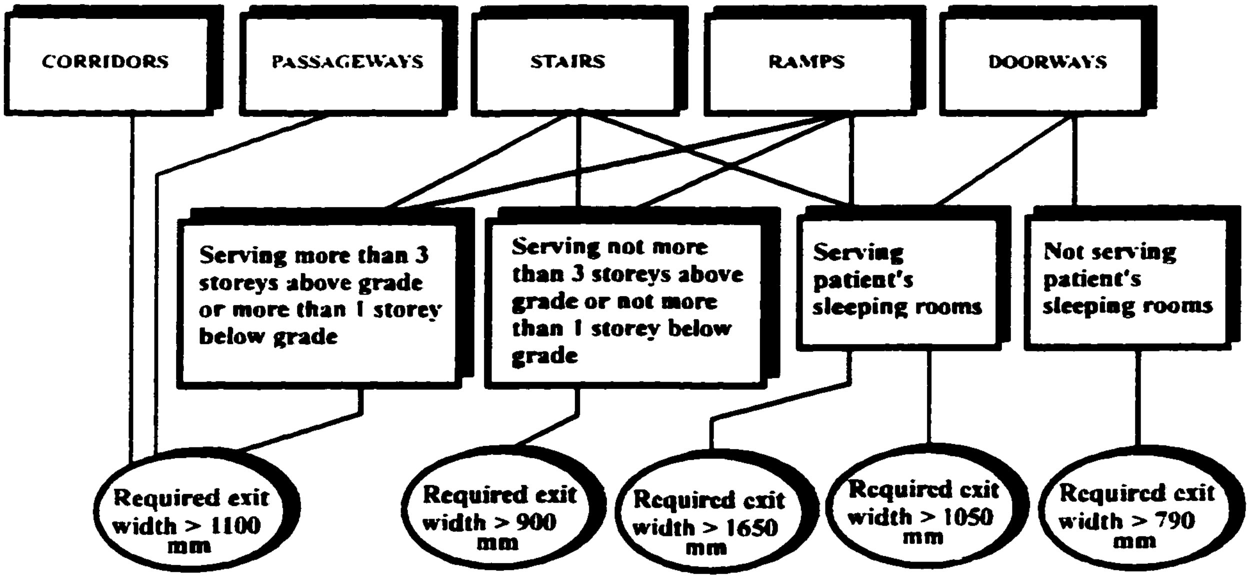


Figure 4.3. **An Example of Forward Chaining Inference**

Another basic component of the expert system is the user interface. Its role is to facilitate the dialogue between the user and the computer. It should accommodate various levels of user expertise [Frye et al. 1995]. The user interface consists of such elements as a menu generator, a query module, and a result layout module. The menu generator represents the main menu and submenus displaying the list of compliance categories to be checked. Certain compliance categories may be further sub-divided, depending on their complexity and nature. The query module elicits information by asking the user to respond to a direct question. In addition, data to be input may be obtained by means of a database file. The result layout module presents the compliance conclusion including the input information of building and code requirements. It also provides access to reference materials such as code documents and relevant case studies.

4.2.2. The Information Management Module

The large amount of various information required during the building code compliance checking process should be systematically stored and accessed in a convenient manner. The information management module which is a partial component of the proposed intelligent system is designed to perform such a task. This module employs both hypertext and database techniques to maintain the required information as well as enable the user to access relevant code provisions and pertinent case studies. Hypertext is able to assist the user in browsing easily through the information in building code documents, viewing only the parts that are of interest while avoiding unnecessary information. Moreover, the information contained in the hypertext environment can be easily modified, an important requirement for building regulations. Whereas the hypertext is exploited for its ability to create and represent interlinked discrete pieces of information text, the database system is used to stored the context or the information about code provisions, case studies, and building description required for the code checking process. The context initially contains the information that defines the parameters of the code checking problem and, as the intelligent system reasons about the given problem, the context expands and contains the information or results of the compliance checking process generated by the intelligent system to solve it. For example, a context in the intelligent system to verify whether the size of an opening in an interior fire separation meets code requirements initially contains information about the measured maximum dimension and size of the opening to be stored. The context would expand as the compliance checking process progresses to include information about sprinkling of the fire compartment and code provisions associated with required maximum dimension/size of the opening.

4.3 Knowledge Representation

Knowledge representation refers to the method used to represent the knowledge contained in building codes in a way that is recognizable to the computer in the intelligent system. As was identified in subsection 3.2.1, the regulatory texts in the code documents are written in a natural language format which can not be processed directly by the computer based system. Thus, these texts must be represented in such a format that the code information can be entered in the knowledge-based system. The establishment of such a knowledge base requires appropriate representation techniques. Two forms of knowledge representation to be used in the proposed system are presented in the following subsections.

4.3.1 Rule-Based Representation

Similar to other building code documents, most of the provisions in the NBCC are of the normative propositions which describe prescriptions and contain expressions indicating permission, obligation, or interdiction associated with the modal verbs such as "shall be", "shall have", " shall provide", "shall consist of", "shall comply with" [Moulin 1992]. This type of code provision consists of a number of conditional statements which, when satisfied, lead to a set of requirements or consequences. Hence, a rule-based technique seems the most natural for representing the knowledge in building codes as the form seems to match that of the code provisions. The following example demonstrates the interpretation of a code text into a rule, which is a part of the code representation in the automatic compliance checking system.

The National Building Code of Canada [NBCC 1990] Article 3.1.8.6 concerning Maximum Openings reads:

> (1) The size of an opening in an interior fire separation required to be protected with a closure shall be not more than 11 m^2, with no dimension more than 3.7 m, when the fire compartment on both sides of the fire separation are not sprinkled.
> (2) The size of an opening in an interior fire separation required to be protected with a closure shall be not more than 22 m^2, with no dimension more than 6 m, when the fire compartment on both sides of the fire separation are sprinkled.

Based on the procedural logic found in the requirements of the NBCC Article, the decision trees have been developed and correspondingly, the Article is translated into the following rule:

```
IF FireCompartments are sprinkled IS Yes THEN
        ReqdOpeningSize := 22
        ReqdMaxDimension := 6
ELSE
        ReqdOpening := 11
        ReqdMaxDimension := 3.7
END
IF      OpeningSize <= ReqdOpeningSize
AND     MaxDimension <= ReqdMaxDimension
THEN
        Compliance := "PASSED"
ELSE    Compliance := "FAILED"
END
```

Such a rule-based representation proves to have several advantages [Delis and Delis 1995]:

- able to represent any particular requirement at the same high level of abstraction that appears in the original text;
- easy to understand both by developers and users due to its natural way of expressing knowledge,

- possible to modularize blocks of knowledge like those that appear in building codes. Thus, new rules (knowledge) created due to code revisions may be added or deleted independently of other rules, and
- suitable for a gradual development of the proposed system since each modular rule can be developed separately.

Major disadvantages that accrue when a rule-based representation is used in an application domain include: difficulties in specifying certain control features such as sequences and complex loops; for example, it is difficult to use rules to calculate a square root, and restrictions in localizing control, as the system become more complex, due to undesirable interactions among rules [Allen 1992].

4.3.2 Frame-Based Representation

A frame is similar to a form of data-structure for representing stereotypical (and hierarchical) information of a building [Delis and Delis 1995]. In object-oriented programming, each frame contains a number of slots which are used to store the description of the attributes of an object or a class of objects. For example, an object 'building data' may have the following attributes: name, address, construction type, occupancy classification, fire safety system, number of facing streets, number of storeys, and building area. Frames may inherit information from other frames in a similar manner to that of semantic networks. In the other words, in defining any specific building (e.g. Center for Building Studies: CBS), a frame representation allows CBS to inherit the properties of the generic class of buildings. "Information hiding (of attributes and their values) is useful when visualizing a large knowledge base" [Allen 1992] since it minimizes redundant input data. When new objects are created, only specific information associated with them is required. An example of the frame BUILDINGDATA is given in Figure 4.4. When a building (e.g. CBS) is inspected, an object BUILDINGDATA_1 is created and the user is asked for eliciting general information of the building (i.e. name, address, occupancy classification, etc.), which inherits from attributes of the frame BUILDINGDATA

Frames can be organized into taxonomies using predefined links that represent relationships between various frames [Allen 1992]. Typical predefined links include class/subclass, instance-of/instance, and part-of/part. An example hierarchy of building frames illustrating the use of the class, instance, and part-of relationships is shown in Figure 4.5, the straight lines and black dots represent the parent-child relationship where inheritance occur from parent to child. Due to these inheritance characteristics, descriptive information of the building can be shared among multiple frames.

Frame-based systems provide methods for attaching procedural information, which are known as demons. These methods become activated when they are attached to or removed from slots or when slot values are accessed or changed. For example, the calculation of the CBS building area can be represented as a demon that is invoked whenever the value of the Building area slot in the frame BUILDINGDATA (Figure 4.4) is required.

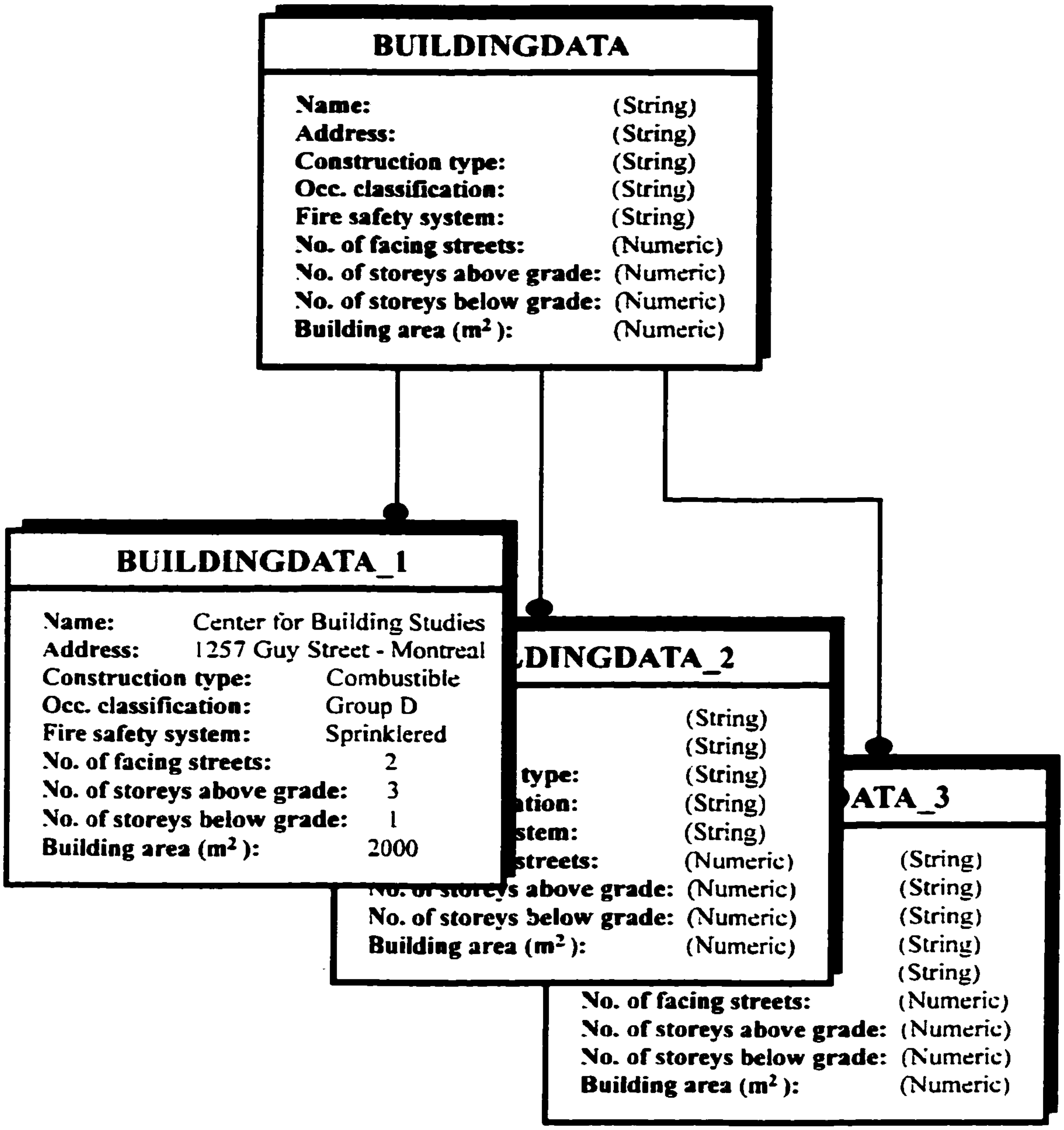


Figure 4.4. **Example of the Frame Representation**

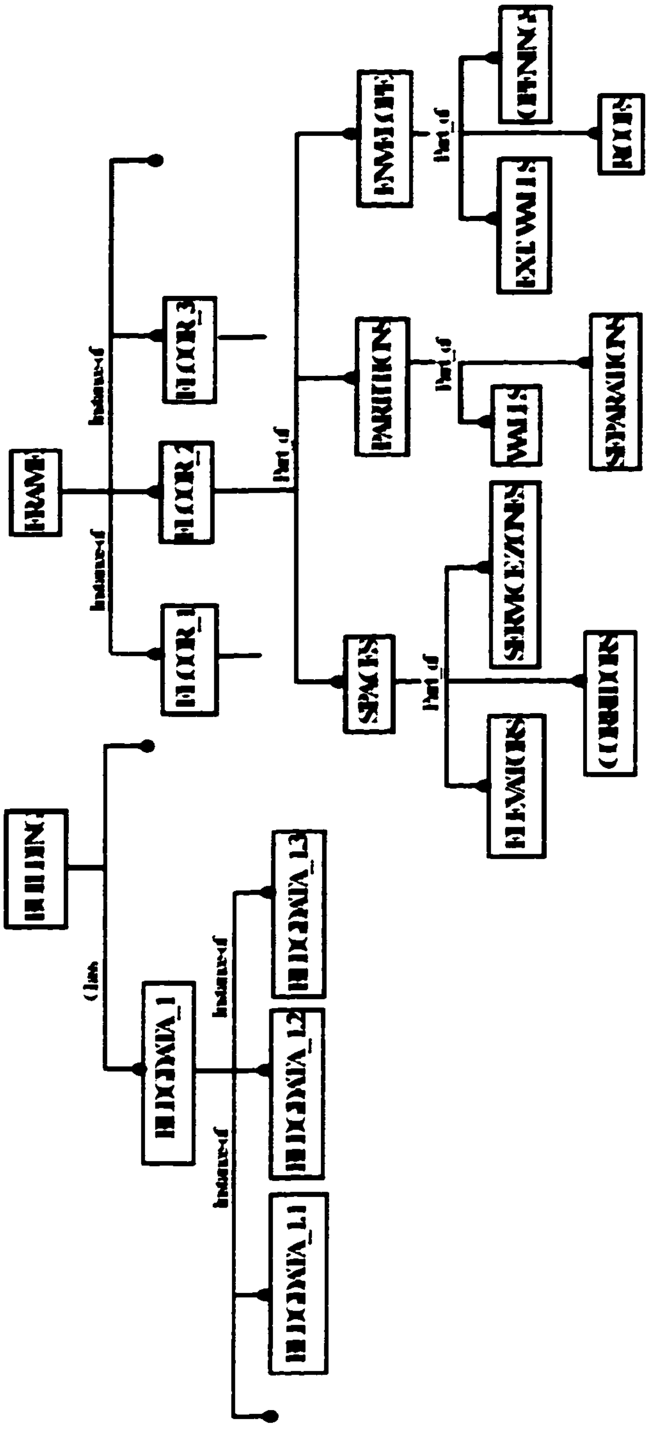


Figure 4.5. **Example Hierarchy of Building Frames**

Chapter 5

IMPLEMENTATION AND VALIDATION

5.1 Introduction

The research methodology has been implemented in a prototype system named HASES (Health And Safety Expert System). The HASES implementation aims at the development of an automated code compliance checking system for the diagnostic of existing buildings during the inspection process. HASES relies on knowledge and reasoning to interpret the regulatory requirements found in Part 3 - Use and Occupancy of the NBCC [1990]. This chapter is divided as follows: first, the software architecture HASES is presented; second, the main characteristics of the research project are identified; third, the selected development tool Level5-Object is described; and finally, the validation of the prototype system is summarized. Comments on HASES as well as its limitations are also presented.

5.2 Software Architecture of HASES

The software architecture of HASES consists of three main components: knowledge base, inference mechanism, and user interface (see Figure 5.1). In addition to these components, a data management module is incorporated in the prototype system to handle the information management.

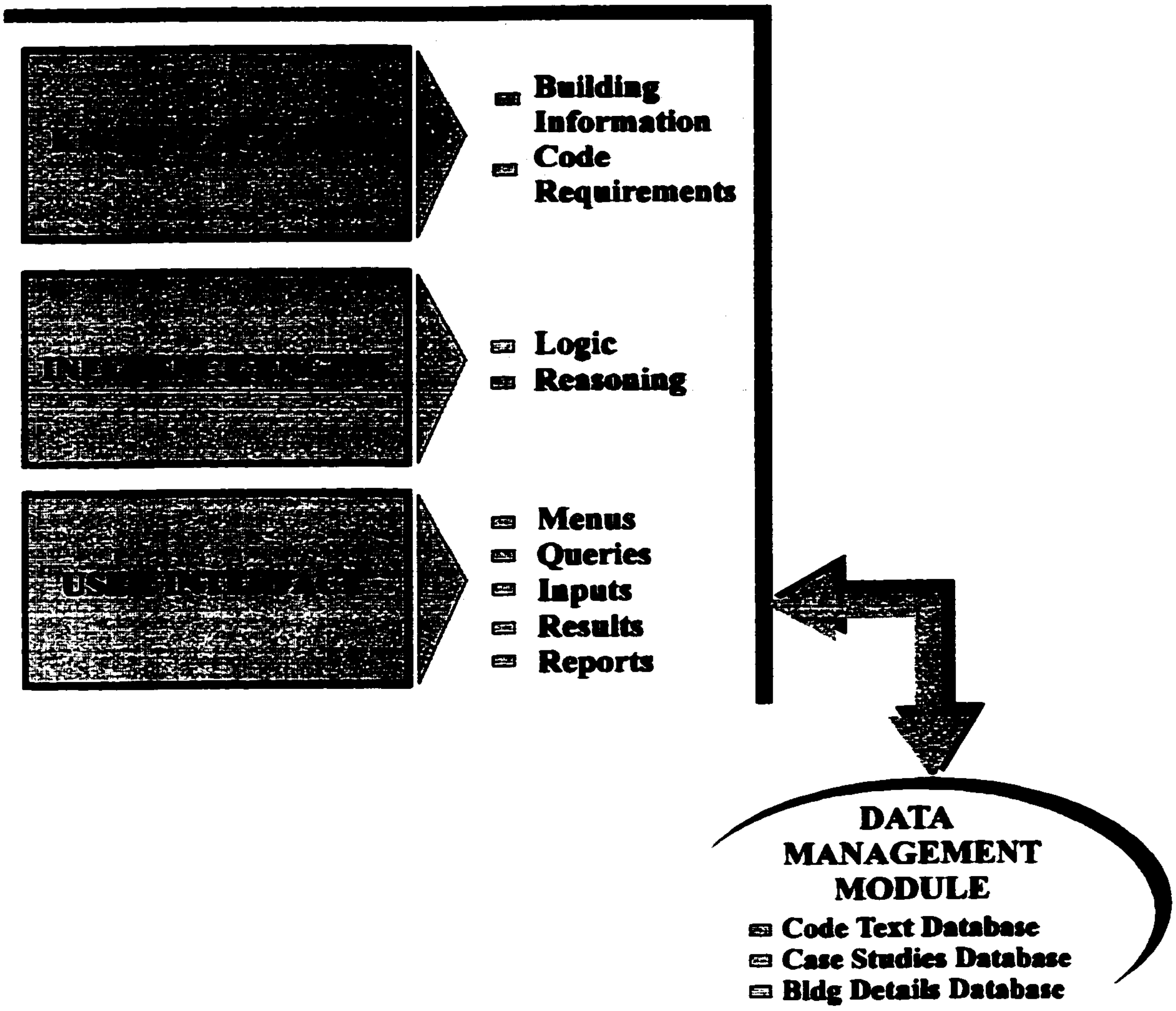


Figure 5.1 **Software Architecture of HASES**

5.2.1 The Knowledge Base

The knowledge base contains information on code requirements as well as general building data (i.e. name, address, occupancy classification, construction type, fire safety system, number of facing streets, etc.). The development of the knowledge base involves basically three steps. First, code information contained in Part 3 of the NBCC is broken into specific categories to be considered during compliance checking. Second, following the inherent logic for each compliance category, decision trees are developed in such a manner that the data required for each checking is systematically collected with a minimum number of queries. The final step is to structure decision trees in a format, such as rule-based representation, suitable for knowledge-based expert system implementation. The process of this development has been illustrated by Figure 3.7 in Chapter 3. The various types of information present in the knowledge base are implemented in the form of production rules and frames. The former are used to represent code provisions, whereas the latter are used to describe information of the objects of a building to be inspected. The details of knowledge representation have been described in section 4.3.

5.2.2 The Inference Mechanism

The HASES inferencing process applies both backward and forward chaining inference strategies, which have been discussed in the previous chapter. Mostly, the forward chaining is used to verify code compliance. The process is performed through two steps. First, the mechanism obtains information input from the user through queries corresponding to the logic for each compliance category to be checked. Once the required information is completely obtained, the reasoning is carried out through available rules representing the information of building codes. The frame-based representation described in subsection 4.3.2 shows various object attributes. The relationships between the object attributes are defined by rules to ensure the dynamic and logically dependent nature of the attributes. Whenever their conditions are satisfied, these rules fire to carry out the compliance checking reasoning. Typical examples of backward and forward chaining inference strategies have been illustrated by Figures 4.2 and 4.3, respectively, in the previous chapter.

5.2.3 The User Interface

The user interface provides prompts for answering queries, supplying information, receiving results, and producing reports. The HASES user interface is designed to be friendly since it gives attractive and easily understood prompt formats for eliciting information, which satisfy the requirements of various code practitioners. The information is collected in such a fashion that only the number of questions needed to reach a conclusion are prompted, whilst avoiding unnecessary queries. This approach allows the system to operate fast and provide reliable results. Another element of friendliness of the user interface is the automation in displaying compliance results and generating reports, hence, saving much time for building inspection. The accessibility of code texts and relevant case studies provided by the user interface makes inspectors more confident in carrying out the building inspection process as well as in defining solutions for specific problems in code violations.

5.2.4 The Data Management Module

The data management module is integrated in HASES to provide users with access to external databases in which basic building details, code texts and case studies are stored.

Basic building details represent general information not specific to any code regulation, such as name, address, occupancy classification, construction type, fire safety system, number of storeys, number of facing streets, and building area. This data is required to be input at the beginning of the inspection process and stored in an external database as a building record. During the code compliance checking process, whenever a compliance category to be checked requires information from this building record, this data is automatically extracted from the database, thus, avoiding redundant data input. This feature of HASES enables the number of queries to be minimized since several compliance categories to be checked can share the same data from the building record. In addition, this building record can be deleted or updated from the data management module.

Besides the code checking process, the HASES user can query the external database for relevant NBCC texts as well as pertinent case studies which are integrated in HASES by using Windows HLP (WinHelp) files that are dynamically accessed from the expert system. The WinHelp provides custom footnotes where a context string is assigned to identify the relevant document materials. For example, a context string "3.4.4.4" is created in the code text file as the identifier of Article 3.4.4.4 Integrity of Exits. Whenever a code compliance verification concerning this article is carried out, the system calls the context string for identifying all pertinent code provisions to be imported to the user interface for display. In addition, "hotspots" is a feature available in WinHelp to allow the user to navigate between code provisions in the NBCC. Figure 5.2a represent a table of contents consisting of all code requirements in Part 3 of the NBCC incorporated as "hotspots", from which the user can browse any specific code provisions by clicking the desired Article. For example, by clicking the underlined Article 3.5.2.1 Fire Separation around Service Rooms, the user will move to the relevant location (Figure 5.2b). Also, this feature is very useful for displaying cross-references in code documents in a different window. Figure 5.2b is an example of the use of "hotspots" to display cross-reference: clicking the underlined text "Article 3.5.2.2" will reveal another window containing the requirements of this Article. The integration of case studies into the system which follows a similar technique is described in detail in the following section.

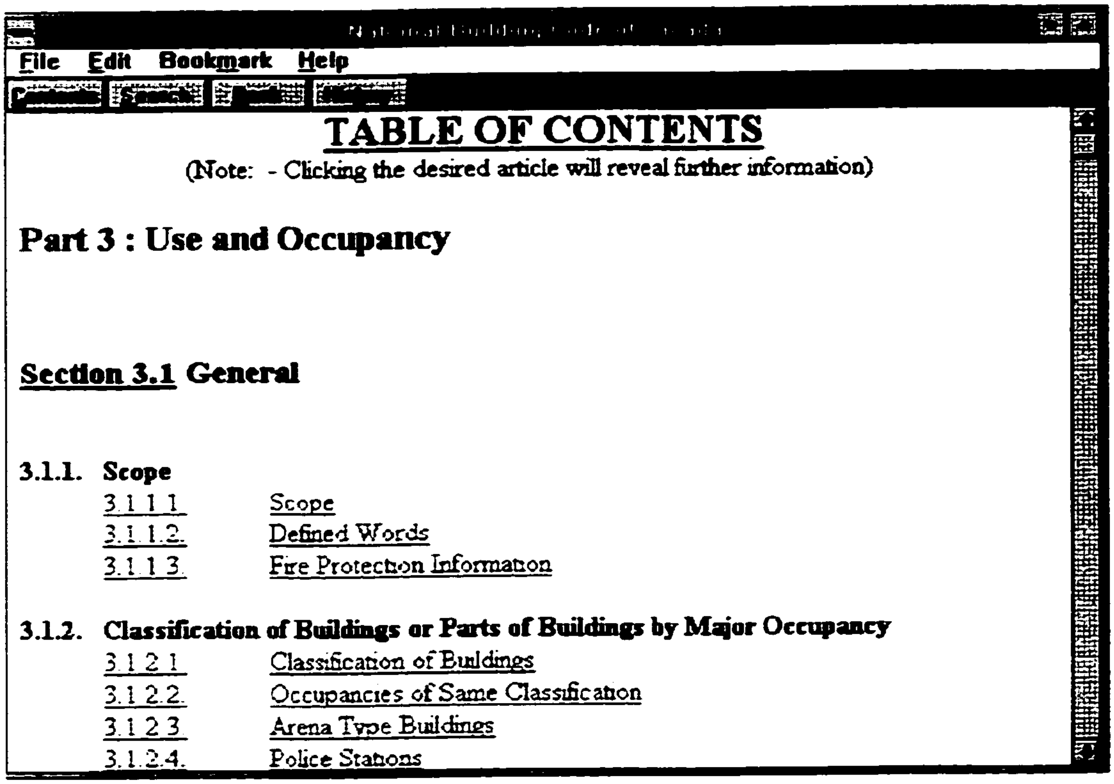


TABLE OF CONTENTS

(Note: - Clicking the desired article will reveal further information)

Part 3 : Use and Occupancy

Section 3.1 General

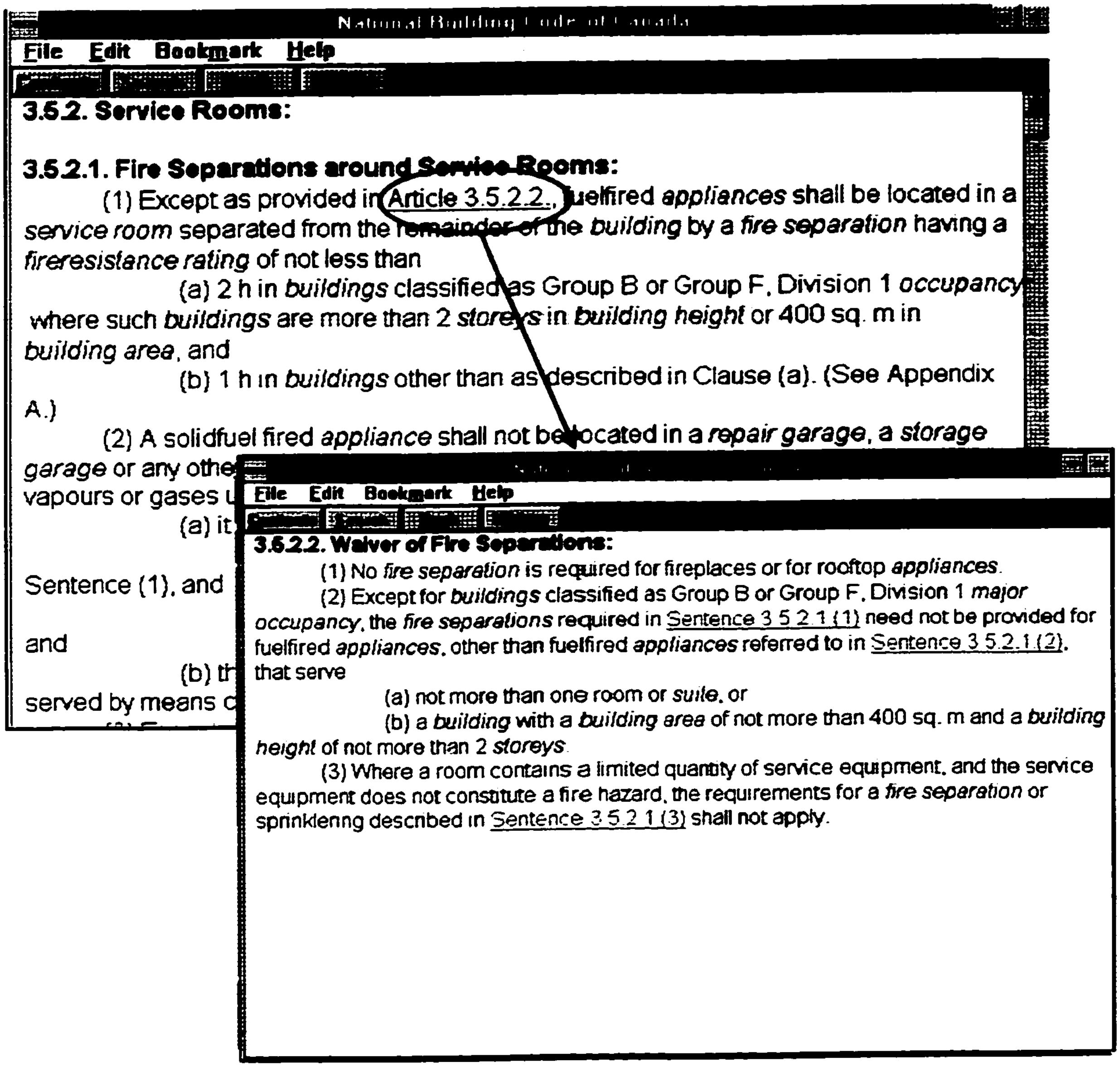
3.1.1. Scope

3.1.1.1.	Scope
3.1.1.2.	Defined Words
3.1.1.3.	Fire Protection Information

3.1.2. Classification of Buildings or Parts of Buildings by Major Occupancy

3.1.2.1.	Classification of Buildings
3.1.2.2.	Occupancies of Same Classification
3.1.2.3.	Arena Type Buildings
3.1.2.4.	Police Stations

Figure 5.2a. **Content Table Containing Part 3 of the NBCC Stored in External Database**



3.5.2. Service Rooms:

3.5.2.1. Fire Separations around Service Rooms:

(1) Except as provided in Article 3.5.2.2., uelfired *appliances* shall be located in a *service room* separated from the remainder of the *building* by a *fire separation* having a *fireresistance rating* of not less than

(a) 2 h in *buildings* classified as Group B or Group F, Division 1 *occupancy* where such *buildings* are more than 2 *storeys* in *building height* or 400 sq. m in *building area*, and

(b) 1 h in *buildings* other than as described in Clause (a). (See Appendix A.)

(2) A solidfuel fired *appliance* shall not be ocated in a *repair garage*, a *storage garage* or any othe

vapours or gases u

(a) it

Sentence (1), and

and

(b) th

served by means c

3.5.2.2. Waiver of Fire Separations:

(1) No *fire separation* is required for fireplaces or for rooftop *appliances*.

(2) Except for *buildings* classified as Group B or Group F, Division 1 *major occupancy*, the *fire separations* required in Sentence 3.5.2.1.(1) need not be provided for fuelfired *appliances*, other than fuelfired *appliances* referred to in Sentence 3.5.2.1.(2), that serve

(a) not more than one room or *suite*, or

(b) a *building* with a *building area* of not more than 400 sq. m and a *building height* of not more than 2 *storeys*.

(3) Where a room contains a limited quantity of service equipment, and the service equipment does not constitute a fire hazard, the requirements for a *fire separation* or sprinklering described in Sentence 3.5.2.1.(3) shall not apply.

Figure 5.2b. **Example of the Use of "Hotspots" to Display Cross-Reference in Code Text**

5.3 Main Characteristics of the HASES Project

5.3 Main Characteristics of the HASES Project

The research project demonstrates the feasibility of an automated code compliance checking system for building inspection. The main characteristics of such a project are identified below.

5.3.1 Interpretation of Part 3 of the NBCC

The National Building Code of Canada [NBCC 1990] is generally a very large and complex regulation document with multiple parts, sections, subsections, and cross-references. Efforts to make this code document more understandable and easier to use have been spent over the last decade, such as training through formal education programs, seminars, and workshops [Frye et al. 1992]. In spite of these efforts, the development of expertise in the NBCC, especially Part 3 Use and Occupancy, has been slow. The code requirements found in Part 3 of the NBCC constitute the main source of information for establishing the knowledge base for the HASES system. Major issues in building the knowledge base include:

- NBCC-Part 3 contains a large amount of complex regulations with numerous subsections, articles, sentences, and cross references to address fire safety, means of egress, exit requirements etc. The complexity is due in part to the qualitative nature of the information which may cause confusion in interpreting texts;
- Code texts are written in a natural language format which requires formalizing and extracting the procedural logic of the code for the establishment of the knowledge base; and

- Differences between national and provincial codes make it difficult to construct a comprehensive knowledge base to be valid for all code requirements.

The complexity due to the qualitative nature of regulatory information found in Part 3 has been a major obstacle for researchers to develop a complete knowledge base containing all code provisions from this part. Some researchers have made efforts to develop computer-based systems for facilitating the code checking process, however, the knowledge base of these systems was mostly limited to certain sections of the code document. For example, the knowledge base of an expert system for building code checking developed by [Frye et al. 1992], as mentioned in subsection 3.3.2, was narrowed only to sections of Part 3 concerning fire protection requirements.

The knowledge base of HASES contains regulatory information addressing most code provisions in Part 3 of the NBCC. To achieve the successful development of such a knowledge base, an incremental approach has been adopted as a suitable technique for gradually encoding the code information in the computer-based system. This code information is initially broken into a set of specific categories to be verified for the compliance during inspection. The set includes, as mentioned in subsection 3.4.1, thirteen compliance categories whose regulatory information is entered in the HASES knowledge base. Each compliance item may be divided into several subcategories, depending on its complexity and nature. In addition, these compliance categories are mostly independent of each other, thus, facilitating the incremental approach to the development of the knowledge base. Table 5.1 represents the list of thirteen compliance categories with their corresponding subcategories. The hierarchical structure of these categories has been established on the basis of the building code regulations discussed in [NRCC 1993]. The problem domain is initially narrowed to one compliance category extracted from several related code provisions, then it is expanded to other categories .

Table 5.1. **List of Compliance Categories and Subcategories**

1	FIRE DETECTION AND ALARM SYSTEMS **Fire Alarm Systems** - Requirements - Restrictions - Emergency Power Supply **Fire Detection Systems** - Smoke Detector Requirements - Smoke Detector in Air Handling Ducts - Fire Detector Requirements - Supervision of Sprinkler Systems
2	FIRE SEPARATIONS **Opening Sizes** **Opening Protection** - Rating of Closures - Positive Door Latches - Hold Open Device Release - Door Closing Devices
3	FIRE ESCAPES **Fire Escape Provisions** **Fire Escape Protection** **Guards and Railings**
4	STRUCTURAL FIRE PROTECTION **Rating of Supporting Construction** **Mandatory Sprinklering of Basements** **Conditions to Permit Heavy Timber Roofs**
5	EXIT REQUIREMENTS **Minimum Number of Exits** **Exit Width** - Protection into Exit Width - Cumulative Exit Width - Required Exit Width **Exit Capacity** **Distance between Exits** **Exit Headroom Clearance** **Location of Exits** **Horizontal Exits** - Gradient in Horizontal Exits - Clear Width on Horizontal Exits - Floor on Sides of Horizontal Exits - Stairs/Steps in Horizontal Exits - Doors in Horizontal Exits **Exit Doors** - Direction of Exit Doors - Width of Exit Door Leaves - Opening of Exit Doors - Stair Riser - Exit Door Distance
6	MEANS OF EGRESS **Barrier Free Access** **General Requirements** - Height of Guards - Door Leaves - Requirements for Exits - Access to Exits from Roofs **Egress from Suites or Rooms** **Means of Egress from Roofs** **Public Corridors** - Minimum Width - Fire Separation **Underground Walkways**

Table 5.1 (Continued)

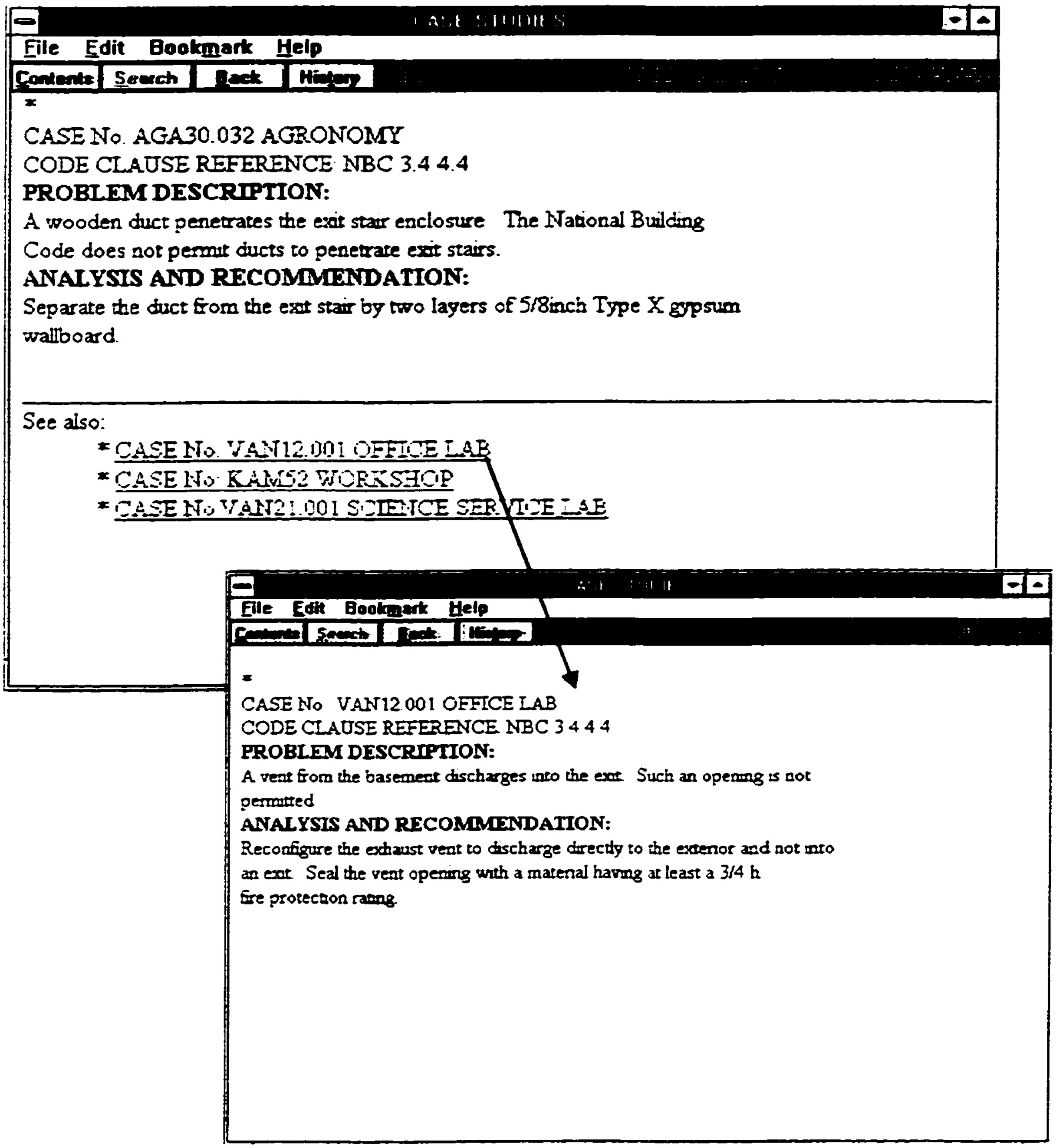
7	INTERIOR FINISH AND INSULATION PROTECTION **Flame-Spreading Rating of Interior Finish** **Protection of Foamed Plastic Insulation** **Protection of Insulation Including Foamed Plastic** **Protection of Foamed Plastic Factory Panels**
8	STORAGE AND REPAIR GARAGES **Interior Stairs Extending to the Roof of a Storage Garage** **Separation of Garages** **Ventilation for Garages** **Sprinklering for Garages** **Vestibules to Stairs in Storage Garages** **Clear Height in Storage Garages**
9	EMERGENCY LIGHTING **Minimum Lighting Levels in Corridors and Exits** **Emergency Lighting for Corridors and Exits** **Provisions of Emergency Lighting**
10	STAIR REQUIREMENTS **Curved Stair Requirements** **Slip Resistance of Stairs and Ramps** **Smoke Tightness of Scissor Stairs** **Landings and Maximum Vertical Rise of Stair Flights** - Limits on Vertical Rise of Exit Stairs - Dimensions of Landings - Clearances on Door Leading onto Ramps - Level Area on Ramp beside Doorways **Handrails Requirements for Stairs and Ramps** - Handrail Requirements - Handrail Height **Guard Requirements for Stairs, Ramps, and Passageways** **Stair Treads and Risers in Exit Stairs**
11	SERVICE FACILITIES **Service Rooms** - Fuel-Fired Appliances - Fire Separation - Hazardous Service Rooms **Electrical Equipment Vaults** - Fire Separation - Protective Measures - Ventilation - Sprinklering - Liquid Tightness of Floors **Fire Separation of Vertical Service Spaces** - Vertical Service Spaces - Top of Vertical Service Spaces - Bottom of Vertical Service Spaces
12	HIGH BUILDING REQUIREMENTS **Smoke Control Systems** **Central Alarm and Control Facility** **Voice Communication Systems** **Protection of Electrical Conductors** **Mandatory Sprinklering of Defined Spaces** **Elevators for Use by Fire Fighters**
13	MISCELLANEOUS REQUIREMENTS **Underground Walkways** - Smoke Barrier Doors - Non-combustible Finishs **Interconnected Floor Spaces** **Mezzanine Egress Requirements**

5.3.2 Incorporation of Case Studies

In a recent study conducted by Public Works Canada [PWC 1991], 1700 occurrences of code violations have been identified in 19 Agriculture Canada buildings. This represents a large amount of relevant information which is extremely valuable towards future inspection of other buildings. A number of code violations were analyzed to extract rules [Gowri and Depanni 1994]. This analysis of case studies indicated that the extraction of generic rules to diagnose specific problems would not be useful in practice since no pattern could be established to classify these rules. However, it was proposed to incorporate these case studies in HASES as reference materials for consultation during building inspection. Hypertext technique available in WinHelp, as mentioned in subsection 5.2.4, has been used to carry out this task. The incorporation of case studies in HASES can be characterized by the following.

Classifying Case Studies

The amount of relevant information containing those 1700 occurrences of code violations has been analyzed and classified into 60 cases based on the compliance categories extracted from code provisions (e.g. articles, sentences, and clauses) in NBCC Part 3. Each case study format includes the descriptions of regulatory violations as well as suitable recommendations to meet the code requirement. Figure 5.3 represent a typical case (i.e. CASE No: AGA30.032 AGRONOMY) which concerns code requirements of Article 3.4.4.4 Integrity of Exits. In this case, besides the availability of problem description and recommendation, other case studies addressing code violation regarding regulatory provisions in the same article are integrated as "hotspots" and can be accessed by clicking the underlined case (e.g. CASE No: VAN12.001 OFFICE LAB, CASE No: KAM52. WORKSHOP, or CASE No: VAN21.001 SCIENCE SERVICE LAB).



*

CASE No. AGA30.032 AGRONOMY
CODE CLAUSE REFERENCE: NBC 3.4.4.4
PROBLEM DESCRIPTION:
A wooden duct penetrates the exit stair enclosure. The National Building Code does not permit ducts to penetrate exit stairs.
ANALYSIS AND RECOMMENDATION:
Separate the duct from the exit stair by two layers of 5/8inch Type X gypsum wallboard.

See also:
- * CASE No. VAN12.001 OFFICE LAB
- * CASE No. KAM52 WORKSHOP
- * CASE No VAN21.001 SCIENCE SERVICE LAB

*

CASE No VAN12.001 OFFICE LAB
CODE CLAUSE REFERENCE NBC 3 4 4 4
PROBLEM DESCRIPTION:
A vent from the basement discharges into the exit. Such an opening is not permitted
ANALYSIS AND RECOMMENDATION:
Reconfigure the exhaust vent to discharge directly to the exterior and not into an exit. Seal the vent opening with a material having at least a 3/4 h fire protection rating.

Figure 5.3. **Example of Case Studies**

The information available from these case studies can be consulted for remedial measures for the diagnostic problems in code checking of other buildings whose situations are similar to those described in the case studies. It is noted that in current automated code checking systems, the use of case studies has not been exploited much so far in assisting code practitioners to develop solutions for problems in code violations; except in some building design systems, case studies have been incorporated to support "the rapid generation of design representations" [Flemming et al. 1994].

Indexing and Retrieval

The sixty relevant case studies have been integrated in HASES by using Windows HLP (WinHelp) files that are dynamically accessed from the expert system. Each case is indexed and attached to an appropriate code provision. Once the code compliance checking process has been completed, all cases related to the particular code provision are imported to the user interface for display. Similar to the technique used to identify relevant code provisions from the code document stored in database, as discussed in subsection 5.2.4, custom footnotes in WinHelp are used to create a context string as an identifier for locating the relevant document materials. For example, a context string "3444" is created in the case studies text file as the identifier of a case addressing code violation of Article 3.4.4.4 Integrity of Exits. Whenever a code compliance verification concerning this article is carried out, the system calls the context string for identifying all relevant case studies to be imported to the user interface for display. In addition, the "hotspots" feature which initiates a jump to another topic enables the user to navigate between case studies stored in the database. The user can also access a content table (Figure 5.4) including all case studies incorporated as "hotspots" which allow the user to browse any pertinent case by clicking the desired one. For example, clicking the underlined CASE No: DEL34.HEADERHOUSE will move the user to the relevant location (Figure 5.4). Also, the use of "hotspots" allows the user to move from one specific case to other relevant case studies regarding code requirements of the same article, as illustrated in Figure 5.3.

Requirements for Case Studies in HASES

The incorporation of case studies in HASES aims at assisting building inspectors in two critical ways. First, it provides access to a large memory of previous solutions for code violation problems that contain examples inspectors on site may not remember immediately on their own, since either the amount of case studies are quite numerous or they were generated by different inspectors. A second aim is to quickly provide inspectors with examples of solutions for code violation problems, which may be similar to the current case under investigation.

5.3.3 Portability of the HASES tool

The HASES tool is designed to improve performance in building inspection. The portability is a necessary design objective for the development of such a computerized code checking system. It is noted that no portable computer-based tools are currently available to assist code practitioners in verifying existing buildings for code conformance. In effect, most field inspectors carry out the code compliance checking process in a manual manner by using code documents in a text-based format, which are preferably utilized in offices rather on site [Alhussayni 1996, Nguyen 1996, Personal Communication]. This situation causes time-consuming problems and reduces productivity as well as the quality of performance in building inspection. To overcome such problems, the HASES tool has been designed to be capable of automating the code compliance checking process. Besides the automatic nature of such a computer-based tool to facilitate the building inspection process, it is a portable software product, which enable the on-site inspector, while walking around a building, to carry out the building code compliance checking process, as well as, to quickly generate reports on compliance results.

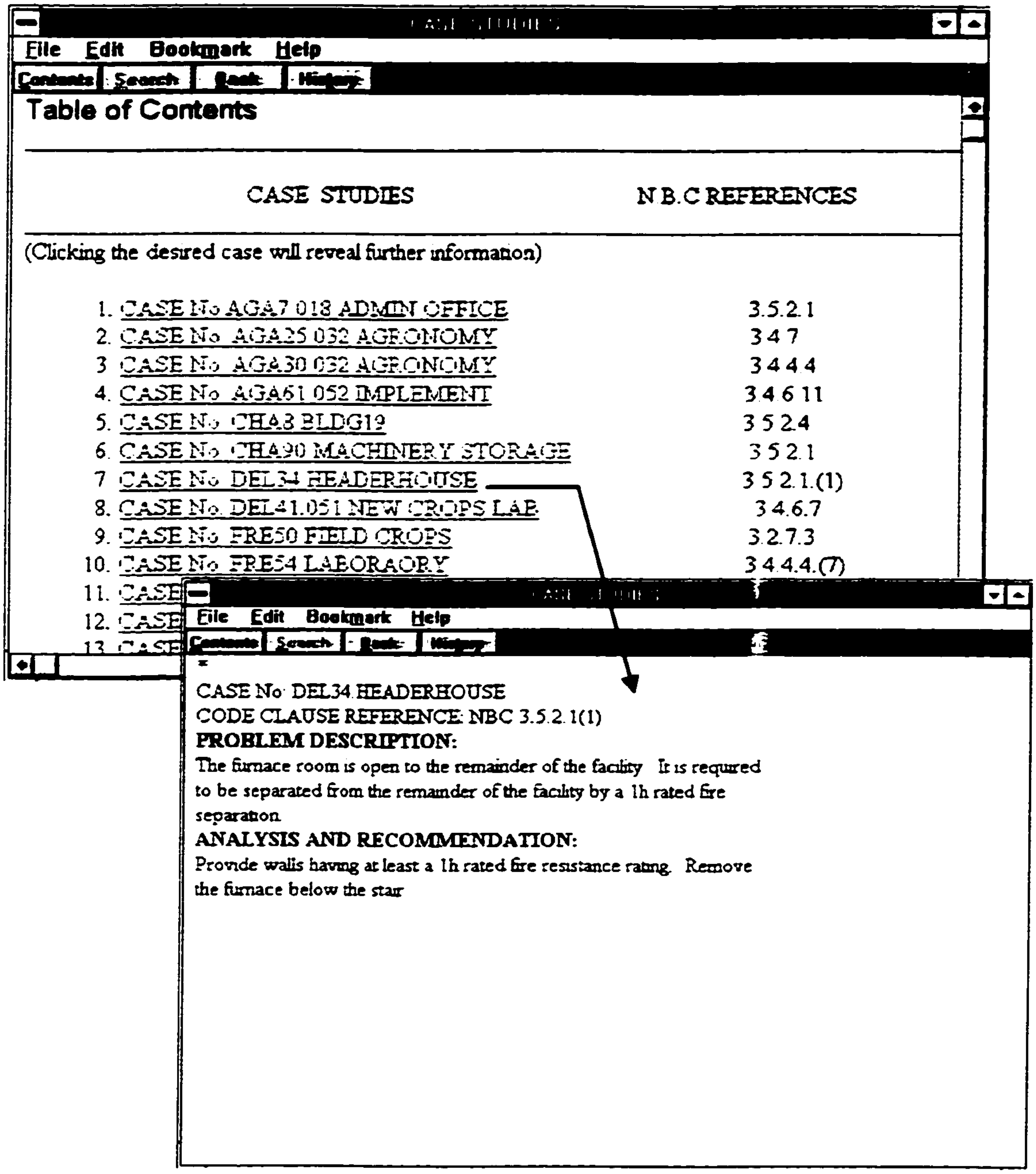


Table of Contents

CASE STUDIES	N.B.C REFERENCES
(Clicking the desired case will reveal further information)	
1. CASE No AGA7 018 ADMIN OFFICE	3.5.2.1
2. CASE No AGA25 032 AGRONOMY	3 4 7
3 CASE No AGA30 032 AGRONOMY	3 4 4 4
4. CASE No AGA61 052 IMPLEMENT	3 4 6 11
5. CASE No CHA3 BLDG19	3 5 2.4
6. CASE No CHA90 MACHINERY STORAGE	3 5 2.1
7 CASE No DEL34 HEADERHOUSE	3 5 2.1.(1)
8. CASE No. DEL41.051 NEW CROPS LAB	3 4.6.7
9. CASE No FRE50 FIELD CROPS	3.2.7.3
10. CASE No FRE54 LABORAORY	3 4.4.4.(7)
11. CASE	
12. CASE	
13. CASE	

CASE No: DEL34 HEADERHOUSE
CODE CLAUSE REFERENCE: NBC 3.5.2.1(1)

PROBLEM DESCRIPTION:

The furnace room is open to the remainder of the facility It is required to be separated from the remainder of the facility by a 1h rated fire separation.

ANALYSIS AND RECOMMENDATION:

Provide walls having at least a 1h rated fire resistance rating. Remove the furnace below the stair

Figure 5.4. **Content Table Containing Case Studies Stored in Database**

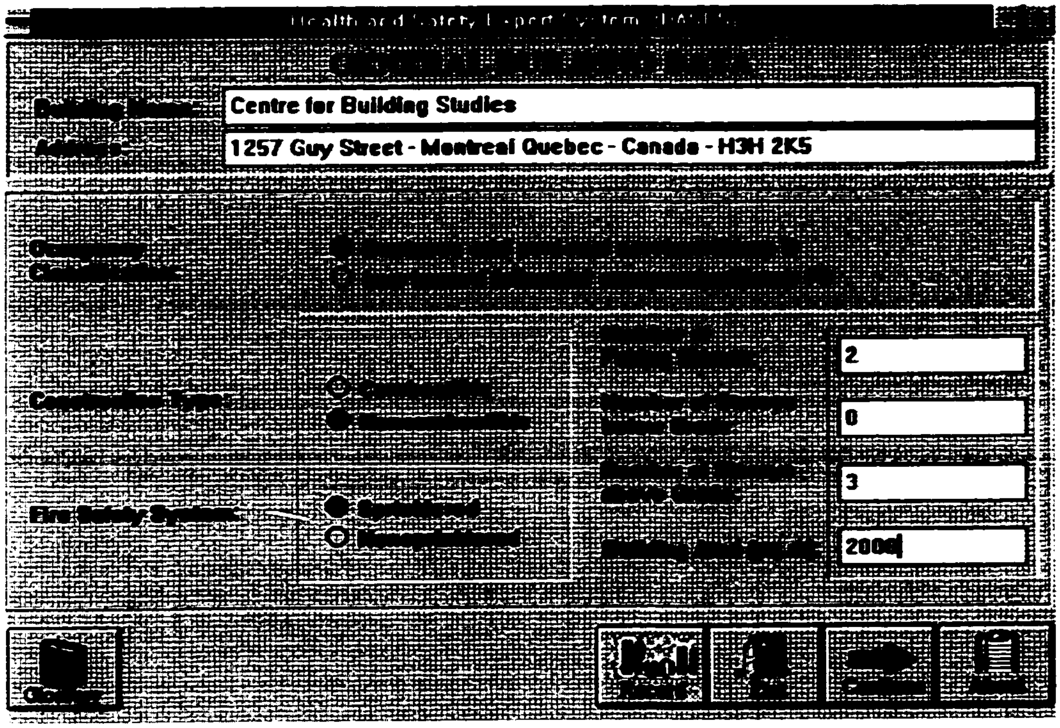
5.3.4 Main Operations of HASES

The main operations that are performed by HASES include:

- Store general building data,
- Select Compliance Items - specific queries, and
- Display compliance checking results and reports.

Store general building data

Figure 5.5 presents the general building data screen from which the user can view, edit, and update the basic data not specific to any code regulation such as the name, address, occupancy classification, construction type, number of storeys, building areas, number of facing streets. This data is stored in an external database as a building record and can be automatically extracted as needed during the code compliance checking process, hence, avoiding redundant queries. In addition, from this general building data screen the user can select, delete, or add a record to the database by activating the "Record" button. In addition, the user is usually provided with a "Glossary" button in case that definition of technical terms in responding to a prompt is required.



Centre for Building Studies

1257 Guy Street - Montreal Quebec - Canada - H3H 2K5

2

0

3

2000

Figure 5.5. **Data Stored in Database**

Select compliance items - specific queries

The user next selects the specific compliance items to be verified from the list of compliance categories in the Main Menu screen (see Figure 5.6). This menu is established on the basis of the building code requirements discussed in the document "Guidelines for the Application of Part 3 of the National Building Code of Canada to Existing Buildings" [NRCC 1993]. Corresponding to each of these categories, submenus list the specific items to be checked. For example, if the user selects the "Exit Requirements" category, a submenu appears and presents eight compliance items (see Figure 5.7). Some compliance items are further subdivided, depending on complexity and nature. Through the HASES user interface, specific data required for compliance checking is finally collected by querying the user. The queries depend on the initial building data stored in the database and the answers to any previous queries. For example, if the user select the pushbutton "Exit Width", HASES will query the user for additional information (see Figures 5.8a and 5.8b).

Display compliance checking results and reports

The compliance check result screen presents a summary of the data collected and of the code requirements used, the reference code text, as well as the final pass/fail result for the compliance check (see Figure 5.9). The "Code Text" button allows the user to view the code texts relevant to the compliance check item. Also, the user can access previous case studies corresponding to the specific code reference by activating the "Case Studies" button. The "Code Text" and "Case Studies" buttons are attached to Code and Case identifiers, respectively. Once the verification of a specific compliance category has been completed, the system calls these identifiers to import code provisions and case studies relevant to the compliance category to the user interface for display. For example, as the result screen for compliance verification of Exit Width appears, Code string "3.4.3.1" and Case string "3431" are attached to the "Code Text" and "Case Studies" buttons, respectively, to identify relevant code texts and case studies regarding the regulatory requirements of Article 3.4.3.1 Exit Width. Activating the "Code Text" button or "Case Studies" button will reveal another window displaying relevant code provisions (Figure 5.10) or pertinent case studies (Figure 5.11). From this window, users can move to a content table of the reference materials by pressing the "Contents" button, where they can browse any specific topics through the reference document, as illustrated by Figures 5.2a and 5.4). In addition, the results of compliance checks can be stored in text files as HASES reports. As a report screen appears, the user can view all the data in the current report, as well as add comments or supplementary information if necessary, before sending it to the external text file (see Figure 5.12).

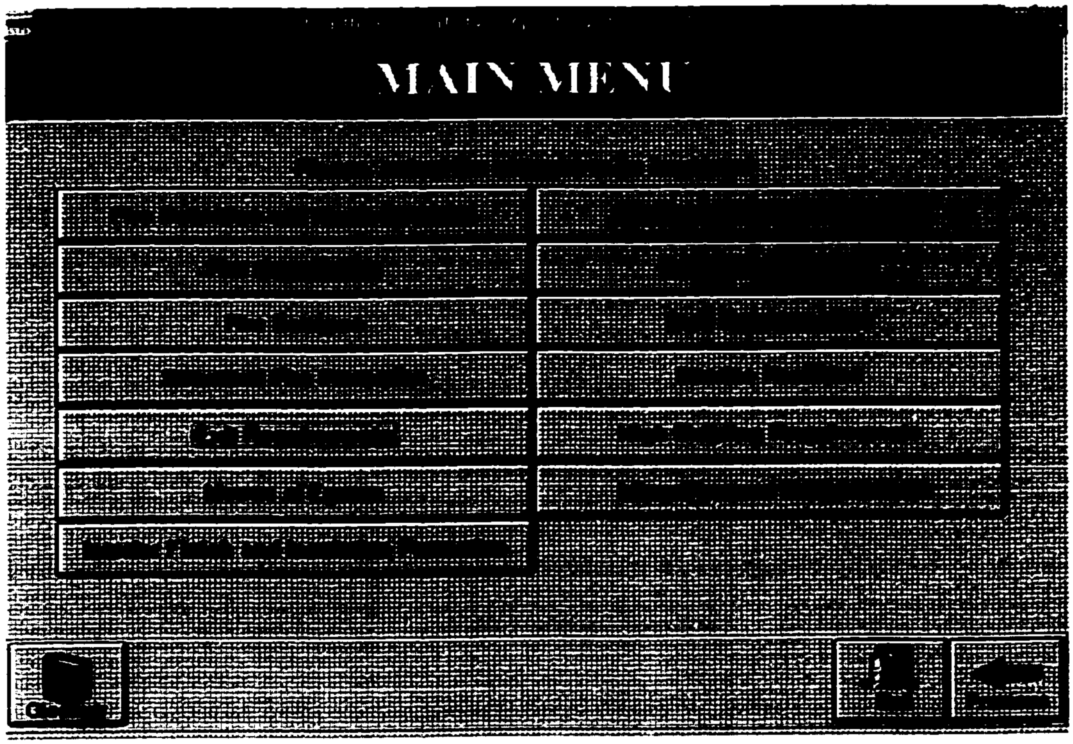
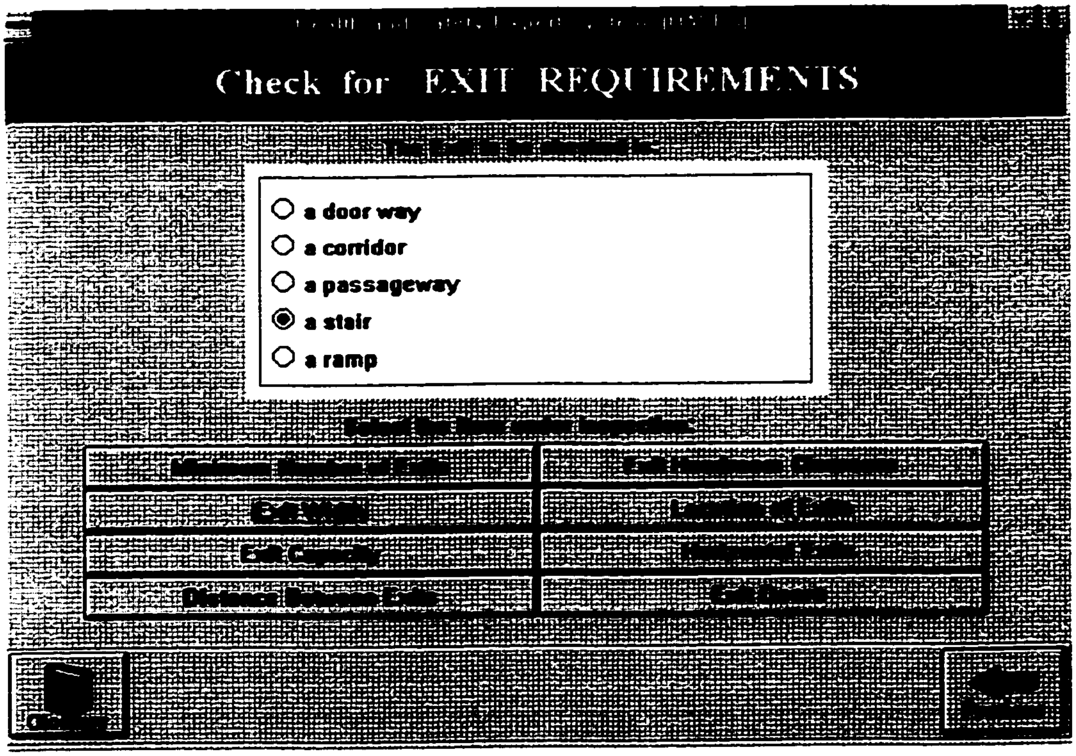


Figure 5.6. **Main Menu**



○ a door way
○ a corridor
○ a passageway
● a stair
○ a ramp

Figure 5.7. **Example of a Sub-menu Screen**

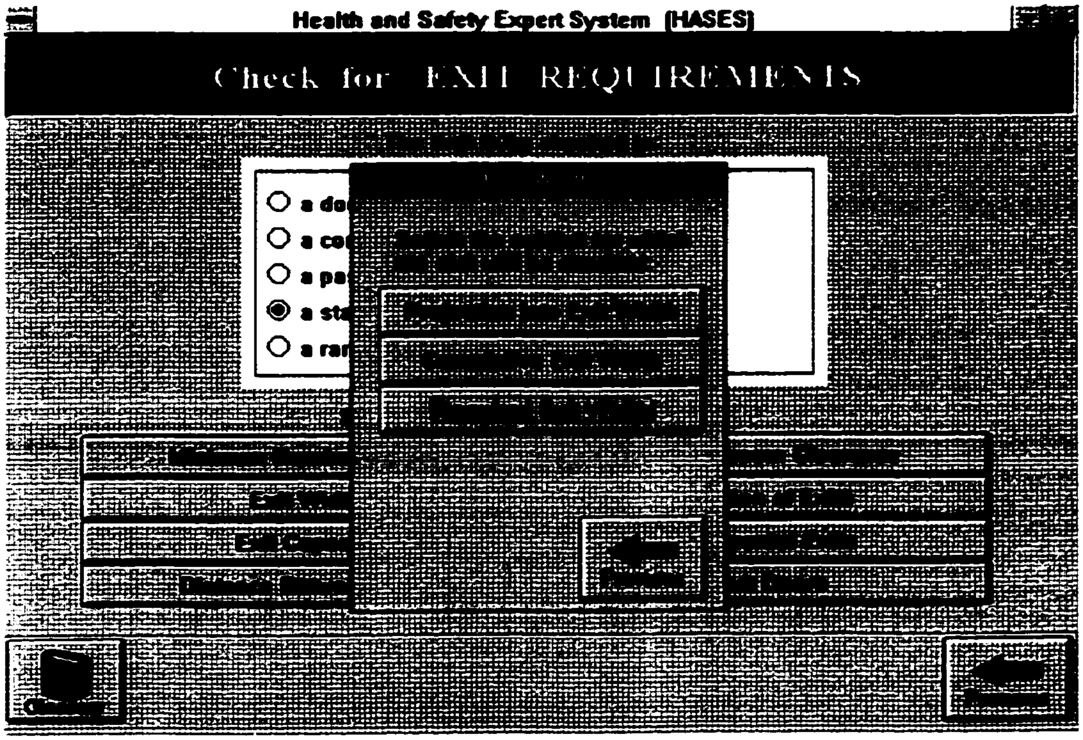


Figure 5.8a. **Another Example of Sub-menu Screens**

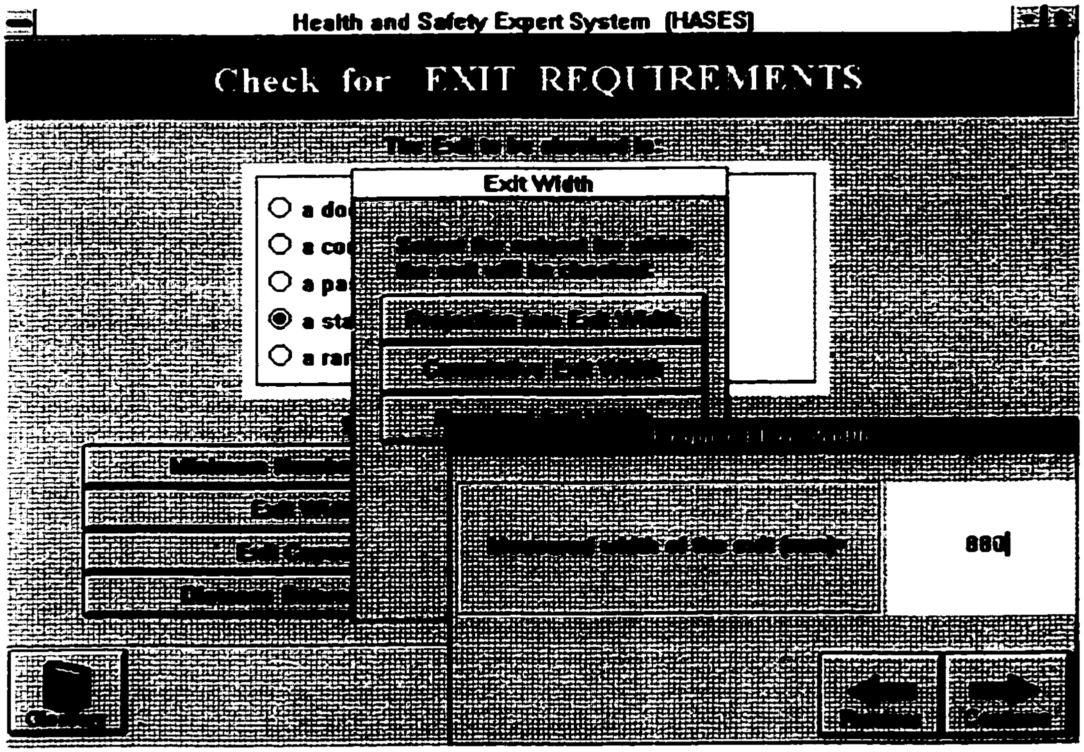


Figure 5.8b. **Example of HASES Query**

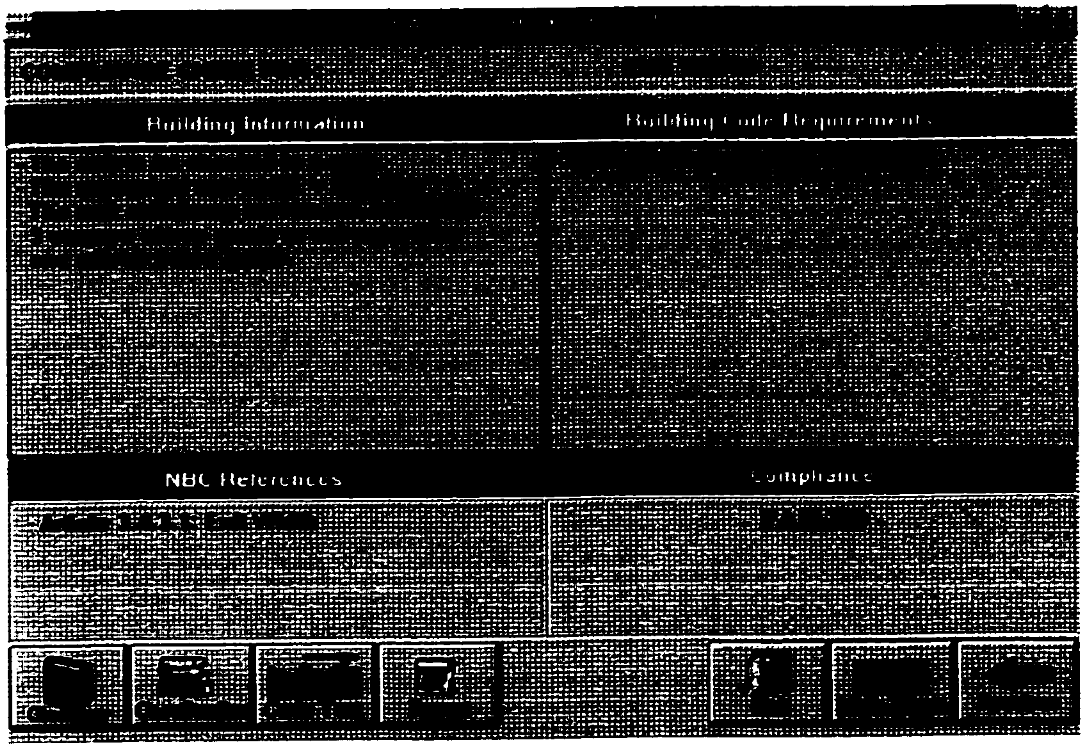
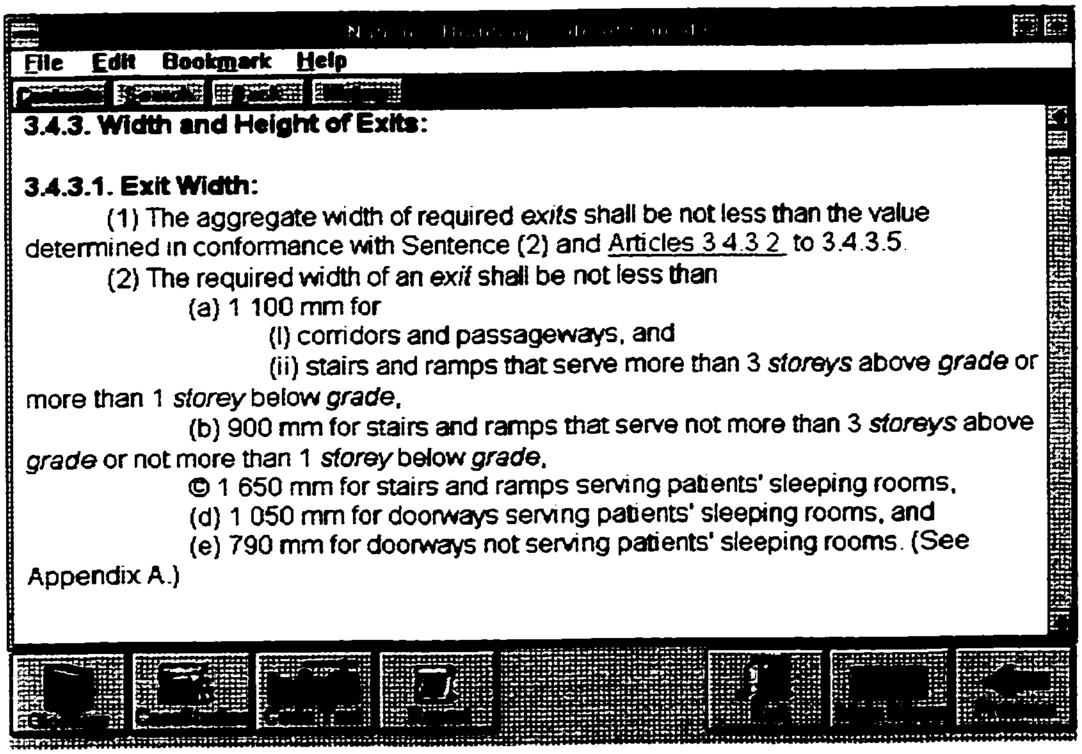


Figure 5.9. **Compliance Check Result**



3.4.3. Width and Height of Exits:

3.4.3.1. Exit Width:

(1) The aggregate width of required *exits* shall be not less than the value determined in conformance with Sentence (2) and Articles 3.4.3.2 to 3.4.3.5.

(2) The required width of an *exit* shall be not less than

(a) 1 100 mm for

(I) corridors and passageways, and

(ii) stairs and ramps that serve more than 3 *storeys* above *grade* or more than 1 *storey* below *grade*,

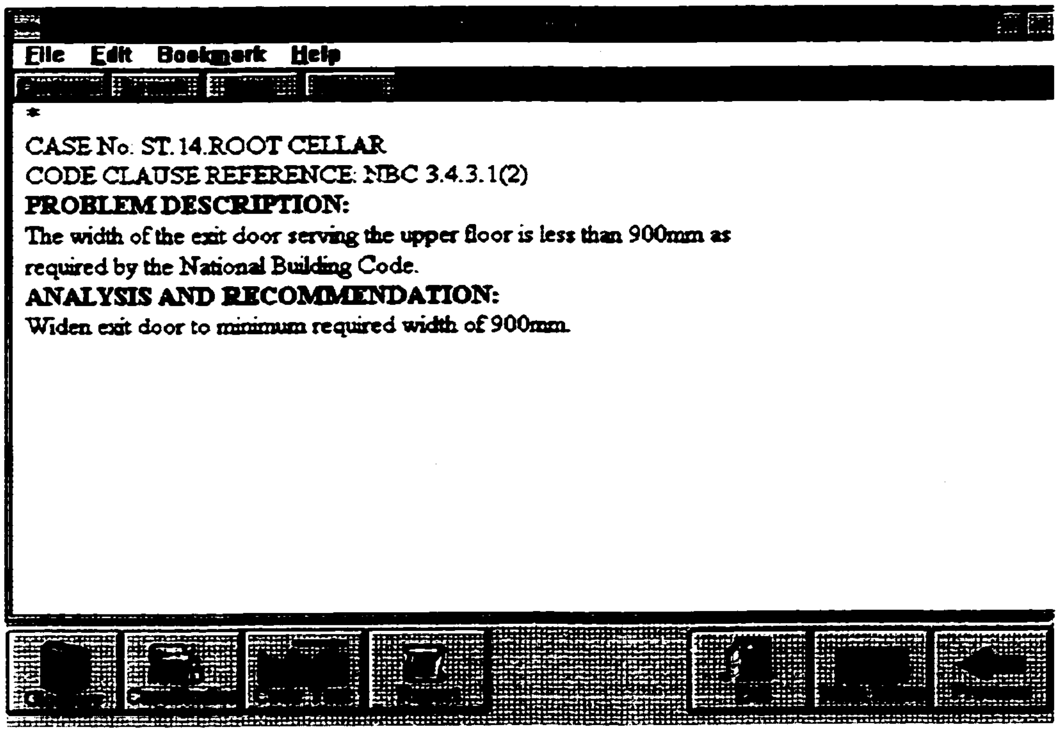
(b) 900 mm for stairs and ramps that serve not more than 3 *storeys* above *grade* or not more than 1 *storey* below *grade*,

© 1 650 mm for stairs and ramps serving patients' sleeping rooms,

(d) 1 050 mm for doorways serving patients' sleeping rooms, and

(e) 790 mm for doorways not serving patients' sleeping rooms. (See Appendix A.)

Figure 5.10. **Relevant Code Text**



*

CASE No. ST.14.ROOT CELLAR

CODE CLAUSE REFERENCE: NBC 3.4.3.1(2)

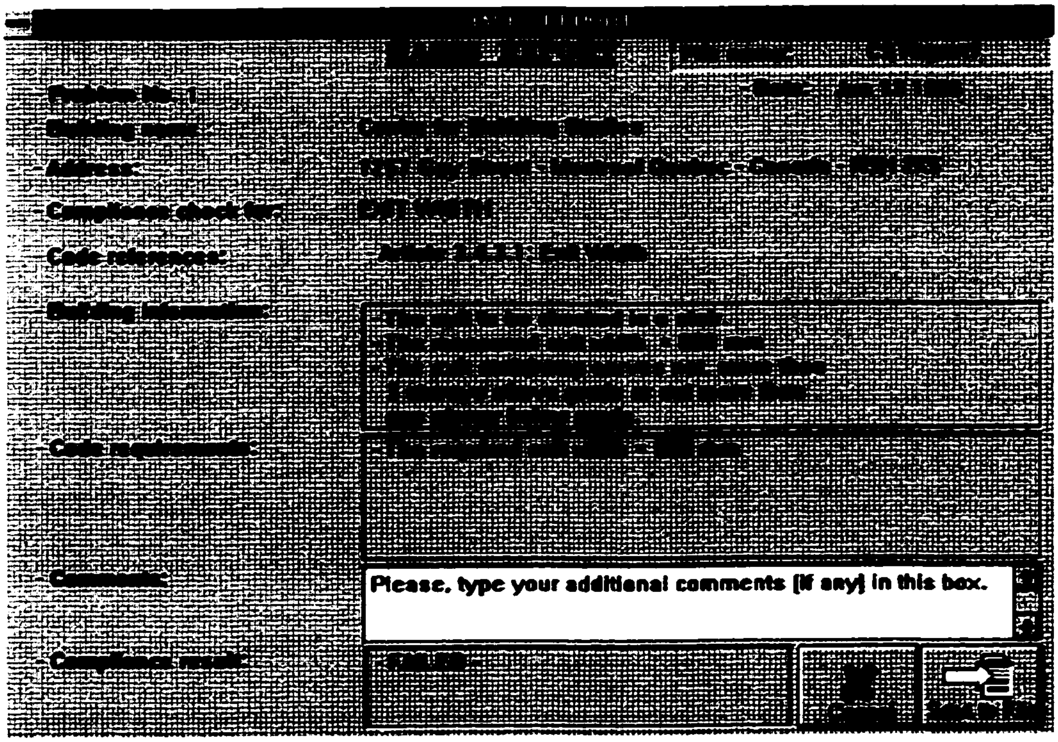
PROBLEM DESCRIPTION:

The width of the exit door serving the upper floor is less than 900mm as required by the National Building Code.

ANALYSIS AND RECOMMENDATION:

Widen exit door to minimum required width of 900mm.

Figure 5.11. **Pertinent Case Studies**



Please, type your additional comments (if any) in this box.

Figure 5.12. **Report Screen**

5.4 Level5-Object: An Expert System Development Tool

The software environment used in the development of HASES is Level5-Object [Level5-Object 1994] created by Information Builders Inc., a tool to facilitate the development of expert system applications in the Microsoft Windows environment. This section is divided into two sub-sections: the first presents the features of Level5-Object and the second discusses programming in such a software environment.

5.4.1 Features of Level5-Object

The most important features to be evaluated in the selection of such a development tool include the knowledge representation, inferencing methodologies, and the end-user interface.

Level5-Object integrates knowledge representation language with rule-base, logic and object-oriented programming methodologies. The knowledge representation language used to develop the HASES application is called PRL (Production Rule Language) which provides an object-oriented framework for organizing information and logic into the expert system. In Level5-Object, knowledge is represented in the form of the production rules, which are inherently suitable for representing the type of information found in building codes. In addition, as an object-oriented environment, Level5-Object allows the developer to represent building data as frames with objects and attributes. Relational links such as IS-A and INSTANCE are provided in PRL to enable inheritance between objects, and the relationships between object attributes are defined by rules. These rules consist of an antecedent and consequence written in the PRL format, which express the logical relationships between the information and the conclusions in the application. In Level5-Object, the Methods/Rules/Demons Editor is used to create rules (see Figure 5.15).

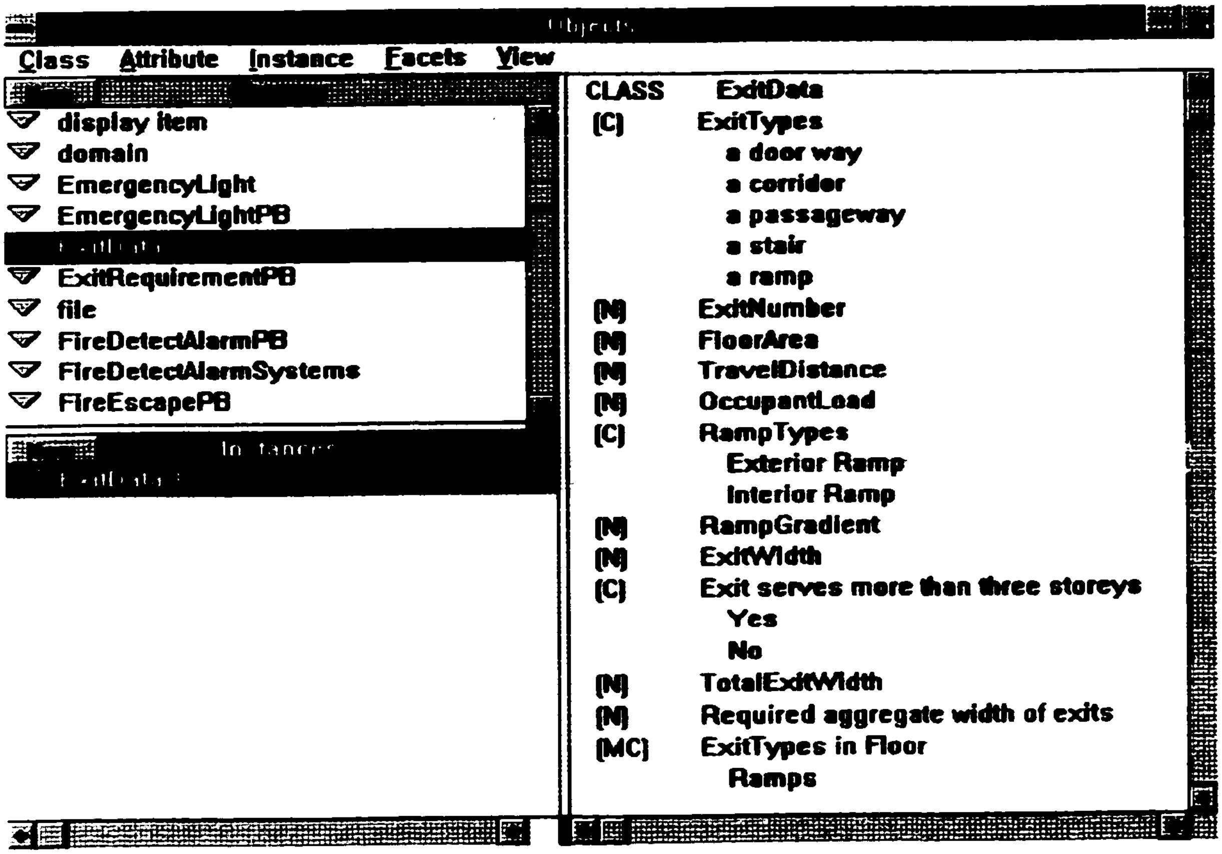
The inferencing methodologies used to carry out the reasoning process in Level5-Object include both forward chaining and backward chaining. Backward chaining, as a goal-oriented approach, is suited to solving diagnostic problems in code checking, where the objective is to verify the code compliance of a given building. However, the combination of forward chaining and backward chaining allows the inference engine to reach decisions more efficiently. The availability of both reasoning techniques in Level5-Object makes it easier to develop such applications as HASES.

The end-user interface provided by Level5-Object facilitates the development of an application due to a number of its capabilities. First, the PRL language which is used to encode the domain knowledge and logic into the application is simple to learn and read, and is similar to the natural English, thus imposing less programming effort. Second, it provides the capabilities to access and perform other actions on data within dBASE II, dBASE III, or dBASE IV databases. This enables the updating or modification of information in databases without any effect in the knowledge base. Also, a direct program-to-program communication facility is available for facilitating a possible communication link using an external file for data exchange between an external program and the knowledge base. Third, Level5-Object is capable of incorporate graphic information in the knowledge base. This capability makes it possible to create a graphical interface for data input.

5.4.2 Program Development in Level5-Object

Programming the HASES application has been accomplished in three steps. First, objects are defined using the Objects Editor. Next, a collection of objects are displayed using the Display Editor to create prompt forms. Finally, rules are implemented using the production rule language (PRL).

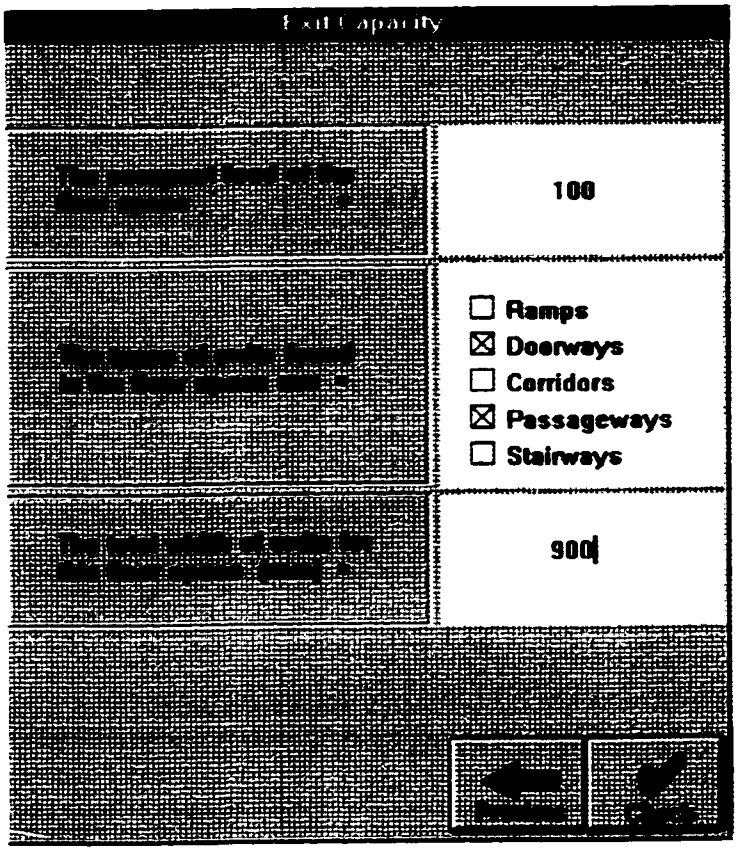
Objects defined in the HASES application may be either physical (e.g. building components) or abstract (e.g. code requirements) concepts to be checked for compliance. In Level5-Object, a class defines the general properties of a group of related objects. For example, the class Building Data defines the common properties of all buildings to be checked for code compliance. The development of the HASES application initially requires the definition of objects and attributes specific to each compliance category. The Object Editor Window available in Level5-Object allows the developer to edit the classes and their attributes. As an example, the definition of the class ExitData is given in Figure 5.13. The class ExitData is defined in association with its attributes such as ExitTypes, ExitNumber, FloorArea, etc. This class provides a good example of what type of values attributes may have. The first attribute, "ExitTypes", represents a compound attribute [C] which can assume only one value from a group of symbolic values (e.g. a door way, a corridor, a passageway, a stair, or a ramp). Second, the numeric attribute [N] represents numeric values or quantities. "ExitNumber", FloorArea", TravelDistance", ... are examples for such an attribute type. The third type "ExitTypes in Floor" is a multicompound attribute [MC], which represents attributes that can assume more than one value from a group of symbolic values (e.g. Ramps, Doorways, Corridors, Passageways, and Stairways). Other classes in the code compliance categories are defined in a similar manner.



CLASS ExitData
[C] ExitTypes
  a door way
  a corridor
  a passageway
  a stair
  a ramp
[N] ExitNumber
[N] FloorArea
[N] TravelDistance
[N] OccupantLoad
[C] RampTypes
  Exterior Ramp
  Interior Ramp
[N] RampGradient
[N] ExitWidth
[C] Exit serves more than three storeys
  Yes
  No
[N] TotalExitWidth
[N] Required aggregate width of exits
[MC] ExitTypes in Floor
  Ramps

Figure 5.13. **Definition of the Class ExitData**

The Display Editor Window provided by Level5-Object is used to create displays and the objects that comprise them. In Level5-Object, the individual parts or objects of a display, are called display items. Each display item created in the Display Editor is an instance of a class, which inherits all properties of the class. A number of built-in display items are available in Level5-Object to facilitate the creation of prompt forms and graphic displays. Several of them are demonstrated in Figure 5.14. The first display item "Textbox", as shown on the left side of the figure, allows the incorporation of text within a display. The second item "Checkbox Group" displays a group of checkboxes which represent the possible values of a multicompound attribute, for example, the types of exits found in the floor space may be "doorways" and "passageways". The third item "Promptbox" is used to prompts the end-user for an attribute's value. The fourth item "Pushbutton" is a standard Microsoft Windows button, which can be attached to a value or another display. When the pushbutton is activated, a set of actions contained in the rules attached to that pushbutton will be carried out. For example, the pushbutton CHECK in Figure 5.14, when being activated, will verify the exit capacity for code compliance and display the checking result.. The description of other built-in display items available in Level5-Object can be obtained from the Level5-Object Reference Guide [Level5-Object Reference Guide 1994]


Exit Capacity

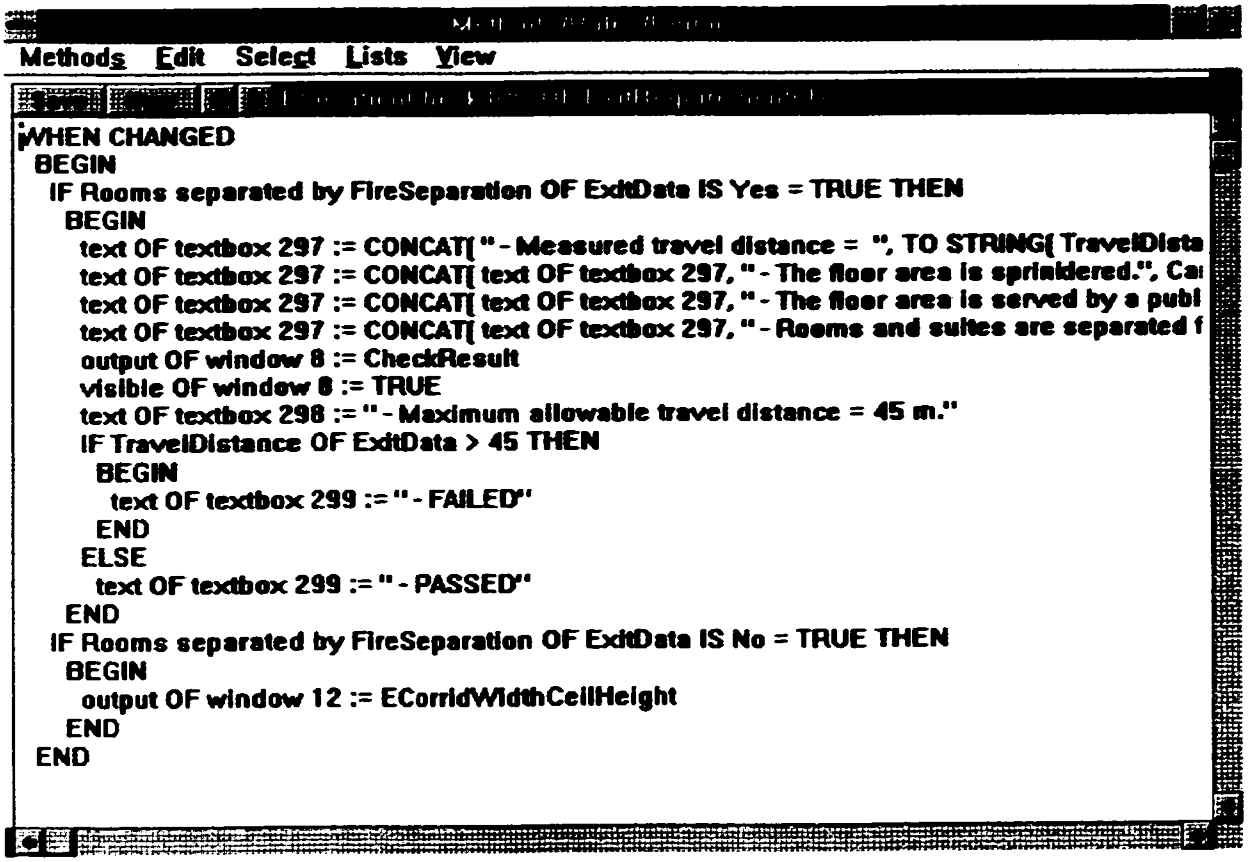
100

☐ Ramps
☒ Doorways
☐ Corridors
☒ Passageways
☐ Stairways

900

Figure 5.14. **Example of Level5-Object Built-in Display Items**

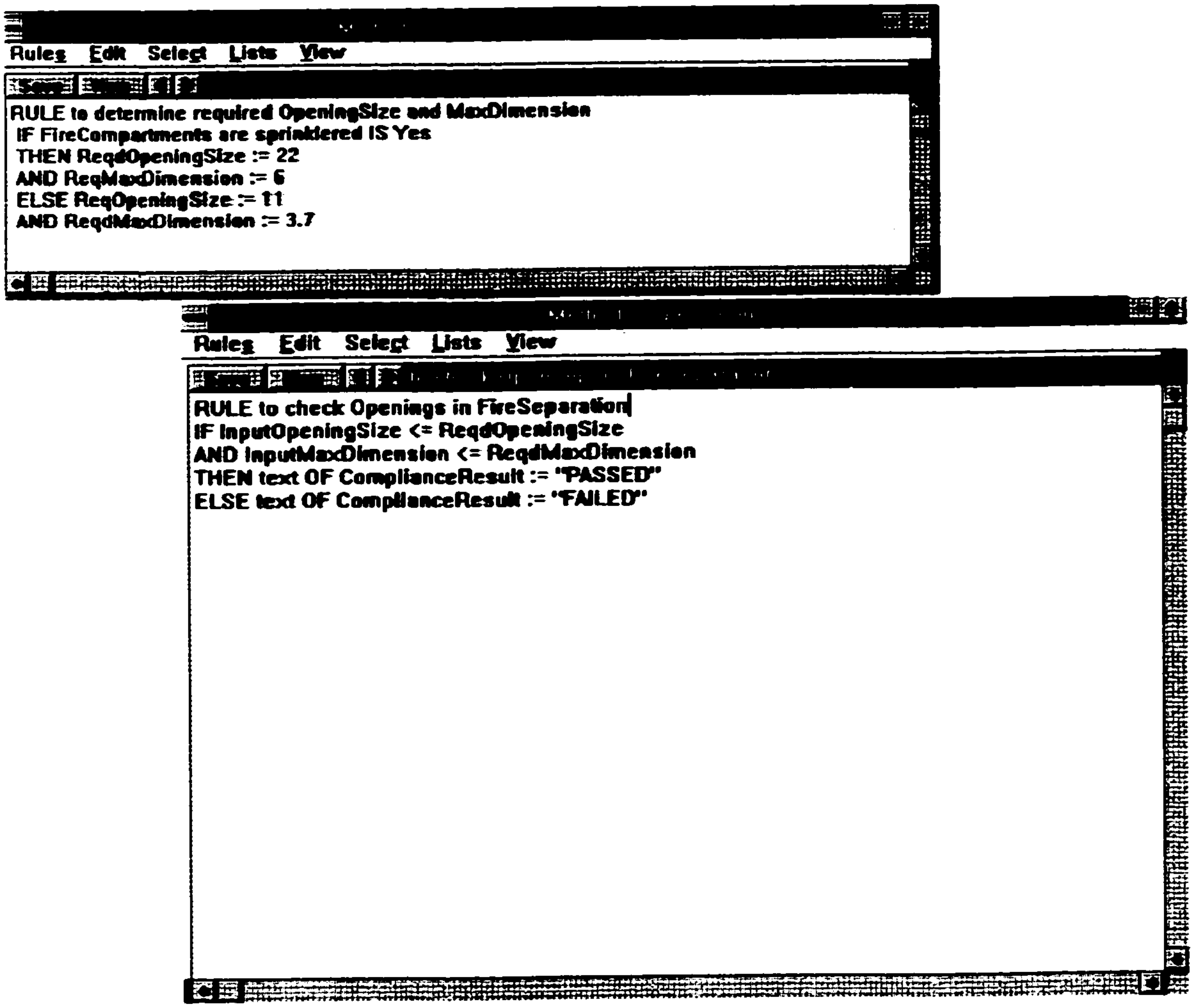
The final step in implementing the HASES application is to program the rules representing the knowledge in building codes, whose objects have been defined in the first step. A rule created in HASES is a program written in PRL, which is able to reason with information in order to reach conclusions regarding code checking results. The rules can be programmed by means of methods, backward-chaining rules, or forward-chaining demons. Two types of methods available in Level5-Object include WHEN NEEDED and WHEN CHANGED. The former specifies a procedure for determining an attribute's value and the latter establishes a procedure that Level5-Object executes when an attribute's value changes. Figure 5.15 shows the example of a WHEN CHANGED method created in the Methods/Rules/Demons Editor, which is attached to the pushbutton "ELocationCheck3PB". When this button is activated, the exit location is verified for code compliance.



```
WHEN CHANGED
 BEGIN
  IF Rooms separated by FireSeparation OF ExitData IS Yes = TRUE THEN
   BEGIN
    text OF textbox 297 := CONCAT[ " - Measured travel distance =  ", TO STRING[ TravelDista
    text OF textbox 297 := CONCAT[ text OF textbox 297, " - The floor area is sprinklered.", Ca
    text OF textbox 297 := CONCAT[ text OF textbox 297, " - The floor area is served by a publ
    text OF textbox 297 := CONCAT[ text OF textbox 297, " - Rooms and suites are separated f
    output OF window 8 := CheckResult
    visible OF window 8 := TRUE
    text OF textbox 298 := " - Maximum allowable travel distance = 45 m."
    IF TravelDistance OF ExitData > 45 THEN
     BEGIN
      text OF textbox 299 := " - FAILED"
     END
    ELSE
     text OF textbox 299 := " - PASSED"
   END
  IF Rooms separated by FireSeparation OF ExitData IS No = TRUE THEN
   BEGIN
    output OF window 12 := ECorridWidthCeilHeight
   END
 END
```

Figure 5.15. **Example of a WHEN CHANGED Method**

A rule is an IF-THEN-ELSE structured program in which the IF statement or antecedent begins with an expression such as an attribute while THEN introduces a conclusion consisting of a PRL command or a simple, compound, or multicompound attribute. When the antecedents have been proved true, the rule fires. The attributes in the conclusion are then established and the commands executed. If the antecedents are proved to be false, the alternate conclusion (ELSE clause) executes, if present. The syntax for a demon is the same as that for a rule. As an example, the creation of the rules "Checking openings in an interior fire separation" is given in Figure 5.16. The code checking process is performed through two rules: the first determines the code requirements concerning the Opening Size and Maximum Dimension and the second provides the compliance result.



```
RULE to determine required OpeningSize and MaxDimension
 IF FireCompartments are sprinklered IS Yes
 THEN ReqdOpeningSize := 22
 AND ReqMaxDimension := 6
 ELSE ReqOpeningSize := 11
 AND ReqdMaxDimension := 3.7
```

```
RULE to check Openings in FireSeparation
IF InputOpeningSize <= ReqdOpeningSize
AND InputMaxDimension <= ReqdMaxDimension
THEN text OF ComplianceResult := "PASSED"
ELSE text OF ComplianceResult := "FAILED"
```

Figure 5.16. **Examples of Level5-Object Rules**

5.5 Validation and Comments on HASES

This section discusses the validation of the HASES knowledge base and addresses several comments on the HASES prototype.

5.5.1 Validation

During the various stages of development, HASES has been tested by SIRICON Inc. at Centre for Building Studies of Concordia University and Public Works Canada (PWC) to ensure the validity of the knowledge in handling all compliance checking scenarios. The HASES reasoning has also been verified for consistency in compliance checking. This verification was carried out for each compliance category to make sure all code provisions have been correctly interpreted and conclusions provided by the HASES are reliable. Interpreting building codes is inherently a complex task, thus, the test for the HASES validity was carefully carried out by different professionals through various phases. In effect, having completed its first version, HASES was initially tested by Depanni S., a professional engineer at SIRICON Inc., and then was distributed to some code practitioners (i.e. Rod Davidge, Melinda Burke, Stephen Lo) at Public Works Canada for reviewing. Several suggestions by PWC have been incorporated to improve the HASES user interface and to enhance the system architecture making it valid for new versions of building codes.

5.5.2 Comments on HASES

Some encouraging responses to the user interface of the HASES prototype have been obtained from code practitioners. The major reasons for their interest of this HASES facility include: ease and convenience in use, simplicity in prompting formats for eliciting information, and automation in storing/updating building data, presenting compliance results, as well as generating reports. Also, the accessibility of code texts and relevant case studies has proved to be the desirable features of HASES.

Besides the above positive comments, there are a few aspects to which minor improvements are required to make the tool more comfortable to use. For example, the program should provide a detailed list of all subsections as well as the result of the code check, which allows users to keep track of every section of the program they entered. In the current HASES tool, the check of a single compliance category, in general, involves selections from several menus and many questions in various windows so that users might get confused and forget what section of the code they are currently in, as well as how many of the subsections of any given menu had not been checked. It was proposed to create a second report that summarizes which sections had been verified along with the compliance check result. This report would not replace the current report. It would open automatically each time the program was run, and would keep track of every section of the program the user entered. Another proposal to reduce confusion is to provide a description of the current section by filling the title bar in the active window with the subsection name. This suggestion has been incorporated in HASES as seen in screen outputs presented in subsection 5.3.4.

Another concern of the code practitioners is the limitation of the HASES knowledge base in its present form. As mentioned in Subsection 3.4.2, the code knowledge incorporated in the proposed system applied only to buildings classified in the occupancy group D (Business and personal services) and F-Division 3 (Low hazard industrial occupancies). Moreover, the regulatory information encoded in the HASES knowledge base is limited to the code requirements contained in Part 3, Use and Occupancy, of the NBCC. Although this is a reasonable concern, the current study focuses mainly on the development of a framework for an automated code compliance checking system. The proposed methodology for developing such a system is applicable to other Parts of the NBCC by extending the knowledge base with additional regulatory information.

Finally, it was also proposed to develop some additional features with respect to graphical representation of building components to be checked for compliance. This approach provides the user with a graphics-oriented module for entering data, which is found to be a more desirable data input mode.

Chapter 6

CONCLUSIONS AND RECOMMENDATIONS

6.1 Conclusions

Building code compliance checking during inspection is a complicated task. The field inspector normally does not have much time and expertise to properly interpret complex regulations of building codes such as Part 3 of the NBCC. The complexity of Part 3 results in part from the qualitative nature of the information and in part from the multiple cross references which cause confusion in interpreting texts. The lack of proper on-site inspection has been one of the cause of code violations found in numerous existing buildings.

Recent research projects on computer-based systems to assist in code compliance checking have attracted significant attention. Most of these projects focus on the development of expert systems for design assistance, hypertext systems for browsing through code documents or for retrieving relevant provisions from the code text. However, none of these systems were aimed at assisting on-site inspectors in automating the building code checking process. This thesis presents a systematic methodology to develop an automated compliance checking system for building inspection.

Several applications of current computer technologies such as expert systems, hypertext systems, etc. to automated code compliance checking have been investigated.

The intelligent system approach combining expert systems, databases, and hypertext techniques was selected as an efficient framework for developing an automated code checking system for building inspection. The expert system is capable of providing advice with diagnostic problems in code checking, whereas the database and hypertext systems are able to handle efficiently the building information management and provide access to the NBCC text as well as relevant case studies.

The development of the knowledge base for the intelligent system involved three basic steps: establishing specific compliance categories extracted from code documents, developing decision trees based on the procedural logic for each compliance category, and translating the decision trees into representation formats suitable for computer implementation. Rule-based techniques have been utilized for representing the knowledge contained in the building code requirements of Part 3 of the NBCC since the form of rules seems to match that of the code provisions; whereas frame-based representation has been found suitable for describing building objects (e.g. floors, zones, spaces etc.) to be checked for code compliance.

The proposed methodology for automated code compliance checking has been implemented in a prototype system namely HASES. The HASES system relies on specific knowledge and reasoning to interpret and verify the regulatory requirements in Part 3 of the NBCC. Basically, the HASES operations include three sequential tasks. First, it stores the general building input data in an external database, which can be extracted as required during the code checking process. Next, by selecting a specific compliance category to be checked from the main menu or submenus, the user is queried for information in such a manner that specific data required for checking is automatically collected, whilst avoiding those unnecessary. Finally, the system displays a summary of the building information and the code requirements used, as well as the final compliance checking results to be stored in text files as HASES reports. In addition, the user is provided with access to relevant code provisions and pertinent case studies.

The HASES implementation demonstrates the feasibility and practicality of developing a computer-based tool to assist building inspectors in automating the code compliance checking process. The advantages of HASES in facilitating the building inspection process can be summarized as follows:

- Easy to verify code compliance due to the accessibility of reference materials including code texts and relevant case studies.
- Convenient and time saving for inspections as well as report writing on the fly, as inspectors walk around a building.
- Consistent and reliable in compliance checking as well as highly efficient in solving code violation problems thanks to the availability of pertinent case studies.

6.2 Contributions

The thesis research attempts to demonstrate the feasibility of developing an automated building code compliance checking system for inspection making use of current computer technologies. It involved developing a framework for automated approach for the diagnostic of existing buildings during inspection. The present study results in a fundamental methodology upon which an automated code checking system can be developed. The contributions are summarized below.

Investigation of computer-based systems for code compliance checking

As presented in Chapter 3, the capabilities of computer-based applications to code compliance checking have been evaluated. Expert systems are found suitable for diagnosing problems with code compliance checking, whereas hypertext systems are efficient for handling information management. An expert system in combination with database and hypertext techniques has been identified as an efficient tool to assist building inspectors in automating the code compliance checking process.

Creation of a framework for automatic approach to code compliance checking

A comprehensive framework for automating the code compliance checking process has been created and adapted to Part 3 of the NBCC to develop an intelligent system, described in Chapter 4, for automated code compliance checking in building inspection. The expert system which is a major component of the intelligent system with a specialist knowledge base will efficiently assist inspectors on site in giving advice with diagnostic problems in code checking. Another component of the intelligent system is the external database capable of handling information management and providing access to code texts and relevant case studies. Although the prototype knowledge base accounted for only the code requirements in Part 3 of the NBCC, the proposed framework can be adapted to other parts of the NBCC or other building code documents to achieve a comprehensive automated code compliance checking system.

Interpretation of Part 3 of the National Building Code of Canada

The regulatory requirements contained in Part 3, Use and Occupancy, of the NBCC (1990) have been utilized as the main source of information for establishing the knowledge base for the intelligent system. This represents a complex regulatory document with numerous sections, subsections, articles, and cross references involved in the checking of any particular compliance item. Another element contributing to the complexity is the qualitative nature of the information found in NBCC Part 3, which may cause confusion in interpreting texts. In addition, since code texts are written in a natural language format, the knowledge in the building code is necessarily interpreted and translated into a suitable format for computer implementation. The interpretation process is characterized by three major tasks: the extraction of specific compliance categories from Part 3 of the NBCC, the development of decision trees for each compliance category, and the translation of these decision trees into rule-based formats which are ready for the intelligent system implementation.

Evaluation of important roles of case studies

The present study evaluates the roles of case studies in providing suitable recommendations for non-compliant buildings to meet code requirements. It is shown that the information contained in case studies is a reliable source to be consulted for remedial measures for the diagnostic problems in code checking. The incorporation of case studies in the automated code checking system will assist code practitioners in developing solutions for problems in code violations.

Development of a portable automated code compliance checking tool for building inspection

The intelligent system framework was implemented in a commercial software development tool Level5-Object to obtain a computer-based code checking tool named HASES. In addition to its automatic nature, the HASES software product is a portable tool which enables the field inspector to carry out, on the fly, the code compliance checking process as well as to generate result reports, as he walks around a building. The portability of the automated code checking tool makes it convenient in use on site.

6.3 Recommendations for Further Research

In order to fully exploit the capabilities of an intelligent system in automating the code compliance checking process, several research issues need to be addressed.

Extending HASES knowledge base

The regulatory requirements found in Part 3, Use and Occupancy, of the NBCC are the major information source to be used for developing the knowledge base of the prototype system. In addition, the knowledge base accounts for code regulations only applied to buildings classified in the occupancy group D (Business and personal services) or F-Division 3 (Low hazard industrial occupancies). Other occupancy groups such as A-Divisions 1, 2, 3, 4, B-Division 1, 2, C, etc. need to be addressed completing the knowledge base for the entire Part 3. Furthermore, the HASES knowledge base must be improved to be fully transparent so that the updating or modification of the knowledge due to building code revisions will not affect the knowledge base.

Enhancing the HASES user interface

The input method in the HASES user interface is a simple query mode where the expert system collects information by prompting the user to answer a direct question. The user interface could be enhanced to allow data to be entered by means of a spreadsheet, a database file, and a graphics-oriented module. Furthermore, the current HASES provides users with a "Glossary" button in case of an assistance in explaining a specific term is required. The improved interface should provide the explanation while prompting queries by using a "hotspots" approach where the user can click the word of interest to initiate a display of its definition.

Expanding external database for further case studies

HASES external database contains only 60 case studies extracted from a recent survey by Public Work Canada on Agriculture Canada buildings. A number of code violation problems do not have corresponding case studies in the HASES external database. The database should be expanded for storing more case studies accounting for as much violation circumstances as possible. In addition, it should be updated by adding new cases whenever code violations exist during the building inspection process.

Incorporating a graphic presentation module

A graphic representation of building components to be checked for compliance may be preferable, since it allows code practitioners to visualize the building in two or three dimensions, making it easy to elicit the building information, especially the geometry, whose descriptions in code text may not be explicit enough. Interlocked spaces, crawl spaces, roof access, building area, areas of refuge etc. are examples of such building information. Such a graphic presentation module should be incorporated in the HASES architecture for displaying the details of building components. It is also necessary to establish a new facility capable of exchanging data between the knowledge base and the graphic module. In addition, the incorporation of a graphic module would constitute a graphics-oriented input method where the inspector is prompted to enter information at specific locations on a graphics screen.

Integrating HASES with other building design systems

The present HASES is aimed at assisting inspectors in automating code compliance checking process. It is still isolated from other building design systems such as architectural design and structural design. It is possible to integrate HASES with these systems so that architects and engineers can verify their designs for code compliance during the design process. The integrated system should be able to read data from engineering drawings to be required for the code compliance checking.

REFERENCES

[Adeli 1988]

Adeli P.H. (1988). "Expert Systems in Construction and Structural Engineering". Chapman and Hall, NY, USA, p. xix.

[Allen 1992]

Allen R.H. (1992). "Expert Systems for Civil Engineers: Knowledge Representation". American Society of Civil Engineers, New York, NY, USA

[Alhussayni 1996, Personal Communication]

Alhussayni B. (1996), Building Inspector at 'Service des permis et inspection', Montreal, Quebec.

[Bedard and Gowri 1990]

Bedard, C. and Gowri, K. (1990), "Automating Building Design Process with KBES", Journal of Computing in Civil Engineering, Vol. 4, No. 2, pp. 69-83.

[Bedard and Ravi 1991]

Bedard C. and Ravi M. (1991), "Knowledge-Based Approach to Overall Configuration of Multistorey Office Buildings". Journal of Computing in Civil Engineering, Vol. 5, No. 4, October, 1991, ASCE, pp. 336-353.

[Blackmore et al. 1994]

Blackmore J., Leong F., and Sharpe R. (1994), "Intentions and Credences - An Australian Approach to Computerized Building Regulation". Proceedings of the First Congress held in conjunction with A/E/C Systems, ASCE, Vol. 1, pp. 131-136.

[BOCA 1993]

Building Officials and Code Administrators (1993). BOCA National Building Code/1993 - Twelfth Edition. Country Club Hills, IL, USA.

[Casson and Stone 1991]

Casson A. and Stone D. (1991), "An Expertext System for Building Standards". CIB W78 and CBS Joint International Workshops on Computer Integrated Construction and Computers and Building Standards, Montreal, Quebec.

[CD-ROM 1993]

Institute for Research in Construction, CD-ROM Master Study: Characteristics of the Market of High Density Electronic Storage and Delivery of Code and Standards Information, by Opinion Search, Inc., National Research Council Canada, Ottawa, Ontario, 27 p.

[Delis and Delis 1995]

Delis, E.A. and Delis, A. (1995), "Automated Fire-Code Checking Using Expert-System Technology". Journal of Computing in Civil Engineering, Vol. 9, No. 2, pp. 141-156.

[De Waard 1992]

De Waard Marcel (1992), "Computer Aided Conformance Checking", Joint CIB Workshops on Computer and Information in Construction, Montreal, Quebec, Canada, pp. 177-187.

[Dym et al. 1988]

Dym, C., Henchey, R., Delis, E.A., and Gonik, S. (1988), "A Knowledge-Based System for Automated Architectural Code Checking". Computer-Aided Design, Vol. 20, No. 3, pp. 137-145.

[Fazio et al. 1989]

Fazio P., Bedard C., and Gowri K. (1989), "Knowledge-Based System Approach to Building Envelope Design". Computer Aided Design, Vol. 21, No. 8, pp. 519-527.

[Fazio et al. 1990]

Fazio P., Bedard C., and Gowri K. (1990), "Constraints for Generating Building Envelope Design Alternatives", Evaluating and Predicting Design Performance, Proceedings of the 3rd International SUNY Buffalo CAD Symposium, Buffalo, New York, pp. 123-134.

[Flemming et al. 1994]

Flemming U., Coyne R., and Snyder J. (1994), "Case-Based Design in the SEED System". Proceedings of the First Congress held in conjunction with A/E/C Systems '94, ASCE, Washington D.C, USA, June 20-22, 1994, Vol. 1, pp. 446-453.

[Frye et al. 1992]

Frye M.J., Olynick D.M., and Pinkney R.B. (1992), "Development of an Expert System for the Fire Protection Requirements of the National Building Code of Canada", Joint CIB Workshops on Computer and Information in Construction, Montreal, Quebec, Canada, pp. 215-226.

[Garrett et al. 1995]

Garrett, J.H., Kiliccote, H., and Choi, B. (1995), "Providing Formal Support for Standards Usage within SEED", Civil and Environmental Engineering Department, Carnergie Mellon University, Pittsburg, USA, pp. 1-26.

[Gowri and Depanni 1994]

Gowri K. and Depanni S. (1994), "HASES: An Expert System Framework for Building Inspection", Centre for Building Studies and Siricon Inc., Concordia University, Montreal, Quebec, Canada.

[Harmon and Sawyer 1990]

Harmon Paul and Sawyer Brian (1990), "Creating Expert Systems for Business and Industry". John Wiley & Sons, Inc., New York, USA.

[Heikkila and Blewett 1992]

Heikkila Eric J. and Blewett Edwin J. (1992), "Using Expert Systems to Check Compliance with Municipal Building Codes". Journal of the American Planning Association, Vol. 58, No. 1, pp. 72-80.

[IRC 1995]

Institute for Research in Construction (1995), "Construction Innovation" Newsletter, Vol. 1, No. 1, July, 1995, Ottawa, Ontario, Canada, p. 2.

[ISO 1986]

International Organization for Standardization (ISO 8879-1986), Information Processing - Text and Office Systems - Standard Generalized Markup Language (SGML), Geneva, Switzerland.

[Kahkonen et al. 1992]

Kahkonen K., Bjork B.C., and Arola P., and Hult S. (1992), "Building Regulations as a Part of a Digitalised Building Information System", Joint CIB Workshops on Computer and

Information in Construction, Montreal, Quebec, Canada, pp. 271-276.

[Kiliccote et al. 1995]
Kiliccote H, Garrett J.H., Chmielenski T.J., and Reed K.A. (1995), "The Context-Oriented Model: An Improved Modeling Approach for Representing and Processing Design Standards", Proceedings of the First Congress held in conjunction with A/E/C Systems '94, ASCE, Washington D.C., Vol 1, pp. 145-152.

[Kumar and Topping 1988]
Kumar B. and B.H.V. Topping (1988), "Issues in the Development of a Knowledge-Based System for the Detailed Design of Structures". Artificial Intelligence in Engineering, pp. 295-314.

[Kumar et al. 1995]
Kumar, B., McLeod, I.A., and Neilson, A. (1995), "Representing Codes of Practice Information", Proceedings of the Sixth International Conference on Computing in Civil and Building Engineering, Berlin, Germany, Vol. 1, pp. 209-213.

[Level5-Object 1994]
Information Builders, Inc. (1994), "Level5-Object for Microsoft Windows, Release 3.5". New York, NY, USA.

[Level5-Object Reference Guide 1994]
Information Builders Inc. (1994), "Level5-Object for Microsoft Windows Reference Guide Release 3.5", New York, NY, USA.

[Liebing 1982]
Liebing R.W. (1982), "Systematic Construction Inspection", John Wiley & Sons, Inc., New York, USA.

[Maher 1987]
Maher M.L (1987), "Expert Systems for Engineers: Technology and Application", American Society of Civil Engineers, New York, NY, USA.

[Moulin 1992]
Moulin, B. (1992), "Automated Generation of Rules from the National Building Code Text", Proceedings of the Joint CIB Workshops on Computers and Information in Construction,

National Research Council, Institute for Research in Construction, Montreal, Canada, p. 343.

[NBCC 1990]

National Building Code of Canada (1990), Associate Committee on the National Building Code, 10th ed., National Research Council Canada, Ottawa, Ontario

[Nguyen 1996, Personal Communication]

Nguyen H.C., Civil Engineer at ' Service des permis et inspections', Montreal, Quebec.

[NRCC 1993]

National Research Council of Canada (1993), "Guidelines for Applications of Part 3", Institute for Research in Construction, Ottawa, Ontario.

[PWC 1991]

Public Works Canada (1991), "Health and Safety Evaluations of Agriculture Canada Research Stations". Building Clients Program, Architectural and Engineering Services, Vol. I, II and III, Ottawa

[Raphael et al. 1995]

B. Raphael, B. Kumar and I.A. McLeod (1995), "Potential Roles for Cases in a Design System". Proceedings of the 6th international conference on Computing in Civil and Building Engineering, Berlin, Germany, 12-15 July, 1995, Vol. I, pp.229-233.

[Rosenman and Gero 1985]

Rosenman M.A. and Gero J.S. (1985), "Design Codes as Expert Systems". Computer Aided Design, Vol. 17, No. 9, November 1985, pp. 399-409.

[Rosenman et al. 1986]

Rosenman M.A., Gero J.S., and Oxman R. (1986), "An Expert System for Design Codes and Design Rules", AI in Engineering: Tools and Techniques, pp.745-758.

[Sharpe 1991]

Sharpe R. (1991), "BC Aider-Expert Assistant for the BCA". Proceedings of the Fourth International Conference on Computing in Civil and Building Engineering, Tokyo, Japan, July 29-30, pp. 247.

[Thomas et al. 1992]

Thomas J.R., S.M. Cornick, D.A. Leishman, and D.J. Vanier (1992). Research at the Advanced Construction Technology Laboratory, Applications of Artificial Intelligence in Engineering VII, Co-published by Computational Mechanics publications, Southampton U.K. and Elsevier Applied Science, London, U.K., pp. 35-50.

[Vadas 1992]

Vadas Olivier (1992), "Artificial Intelligent Industrial Applications workshop". Confederation College, Thunder Bay, Ontario.

[Vanier 1991a]

Vanier D.J. (1991), "A HyperCard National Building Code", Electric Architect. Journal of Computers in Architecture, October/November, pp. 2-6.

[Vanier et al. 1994]

Vanier D.J., Thomas J.R., and Worling J.L. (1994), "Standards Processing 2000", Proceedings of the First Congress held in conjunction with A/E/C Systems '94, ASCE, Vol 1, pp. 137-144.

[Vanier 1989]

Vanier D.J. (1989), "Computerization of Building Regulations", National Council of Canada, Institute for Research in Construction.

[Yabuki and Law 1993]

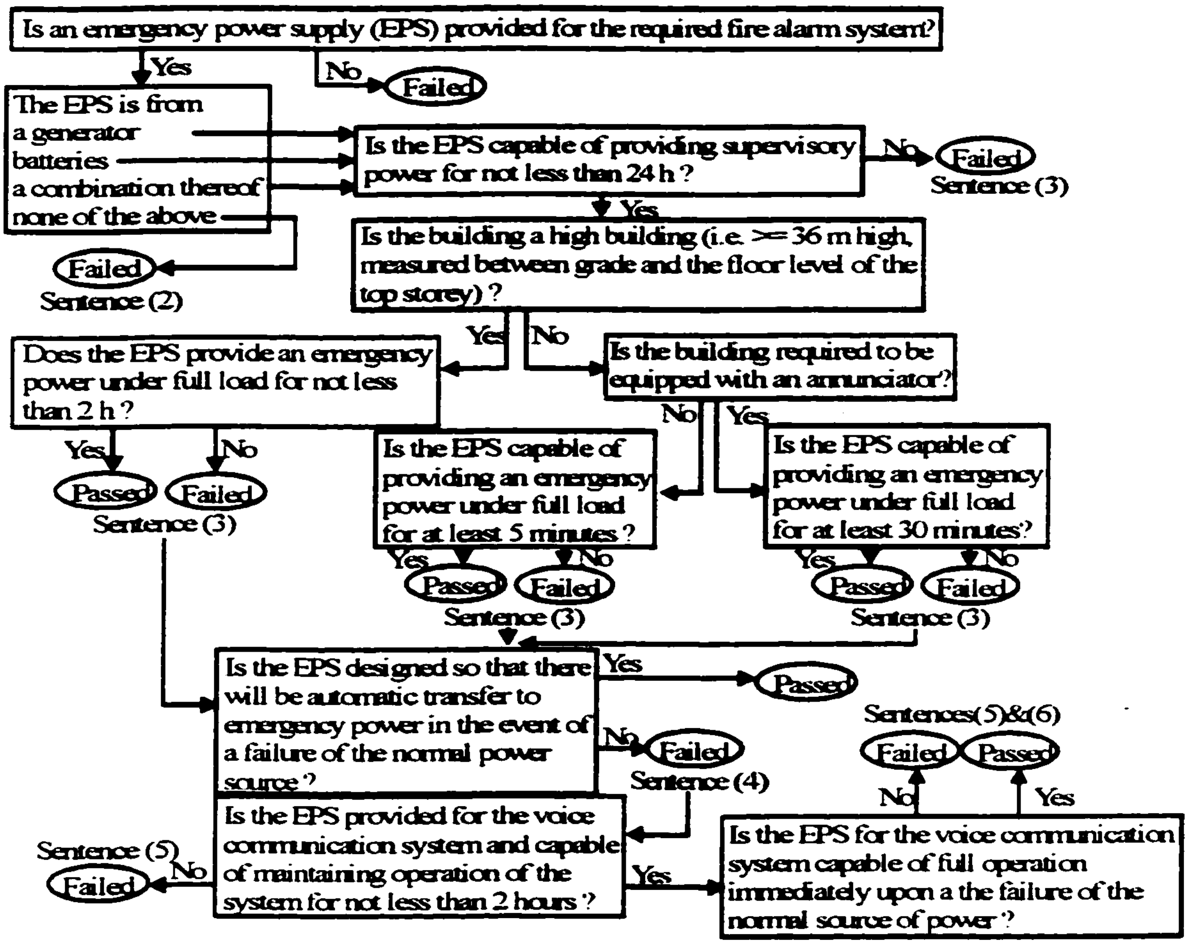
Yabuki N. and Law K.H. (1993), "HyperDocument Model for Design Standards Documentation", Journal of Computing in Civil Engineering, Vol. 7, No. 2, April 1993, ASCE, pp. 218-237.

APPENDIX

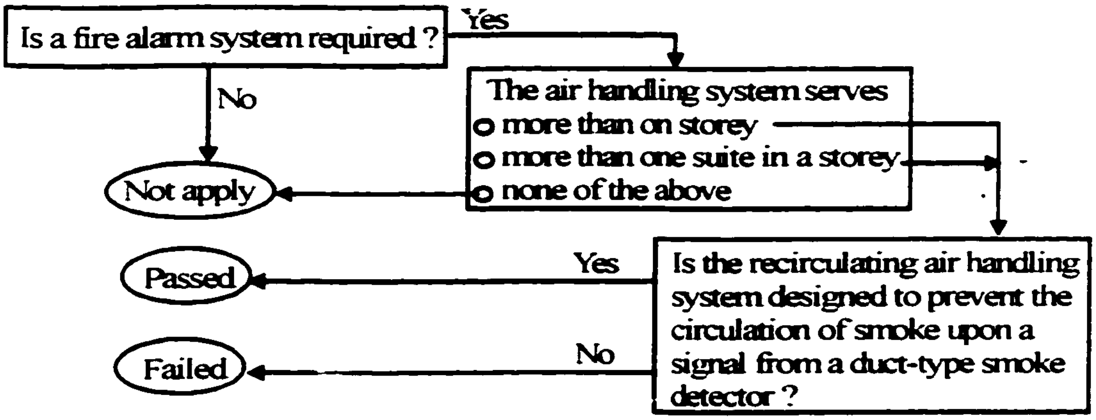
Decision Trees for Code Requirements in Part 3 of the NBCC

1/ FIRE DETECTION AND FIRE ALARM SYSTEMS

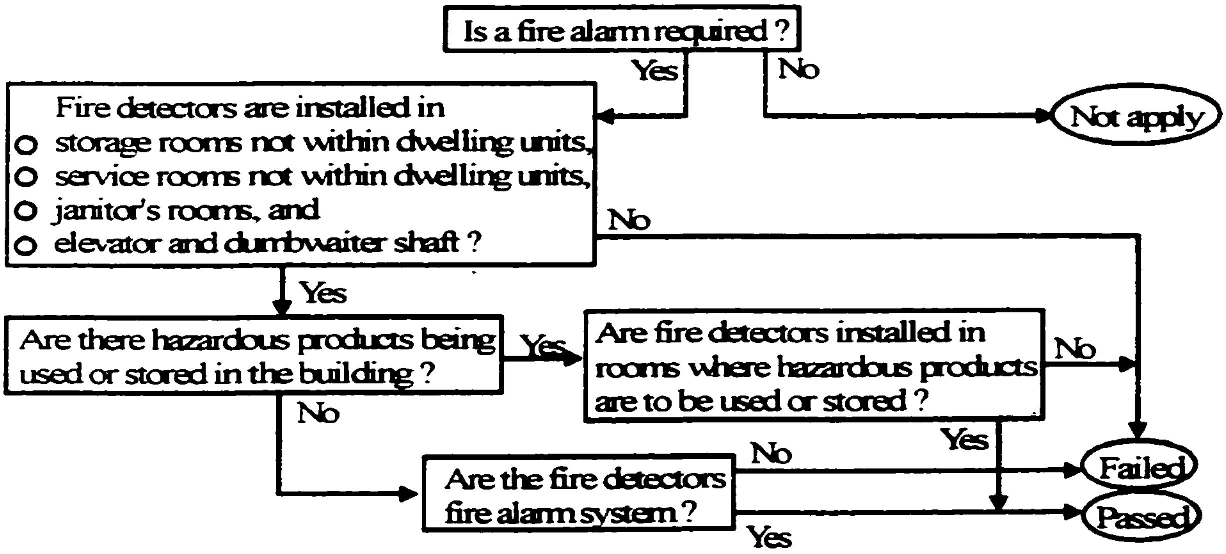
EMERGENCY POWER SUPPLY (Article 3.2.7.8)



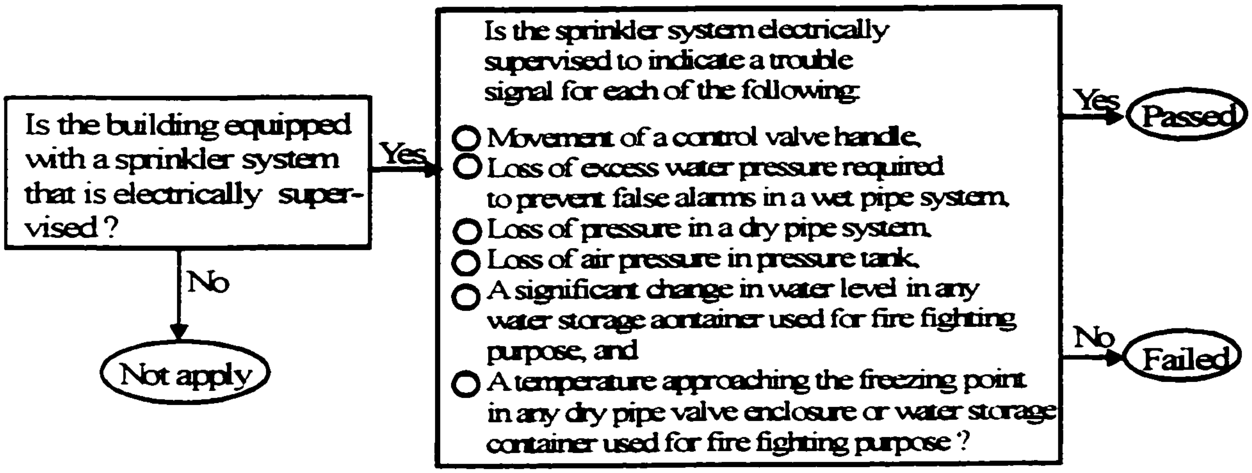
SMOKE DETECTORS IN AIR HANDLING DUCTS (Article 3.2.4.13)



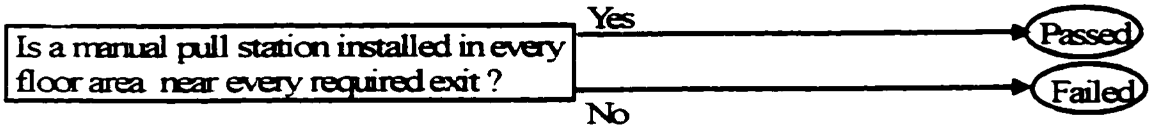
FIRE DETECTOR REQUIREMENTS (Article 3.2.4.10)



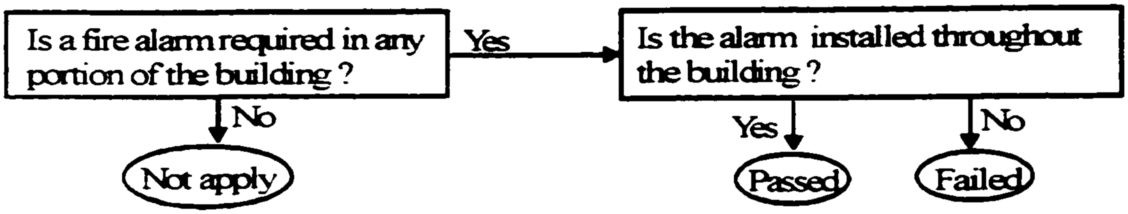
SUPERVISION OF SPRINKLER SYSTEMS (Sentence 3.2.4.16.(5))



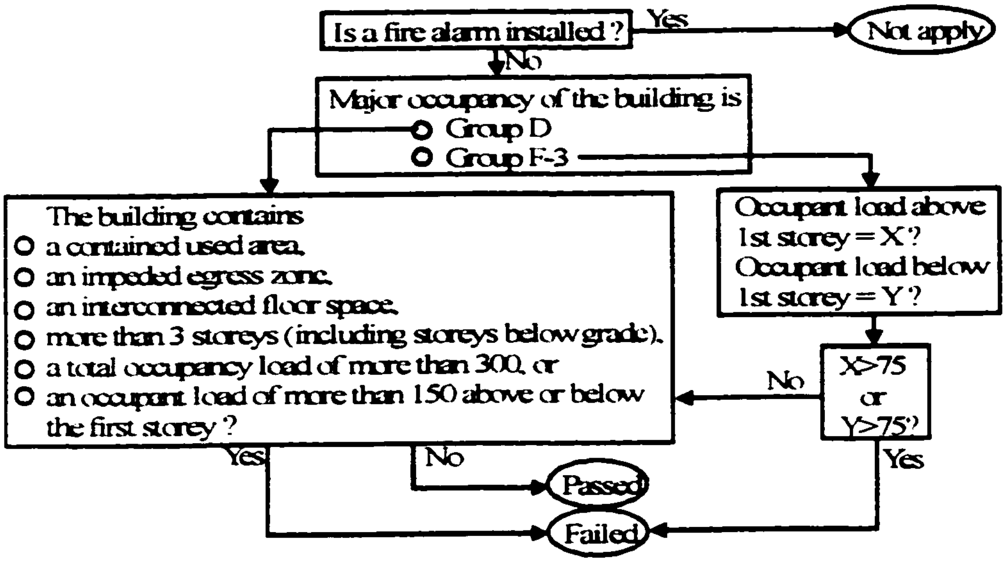
PULL STATION REQUIREMENTS (Article 3.2.4.17)



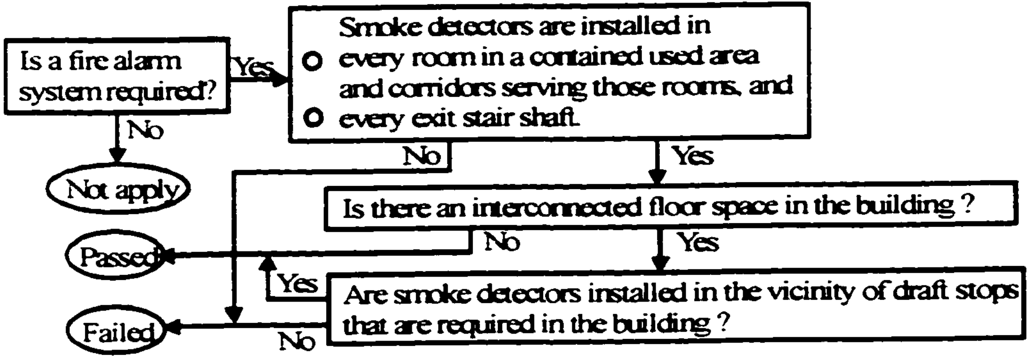
FIRE ALARM SYSTEM TO BE INSTALLED THROUGHOUT BUILDING (Sentence 3.2.4.2.(3))



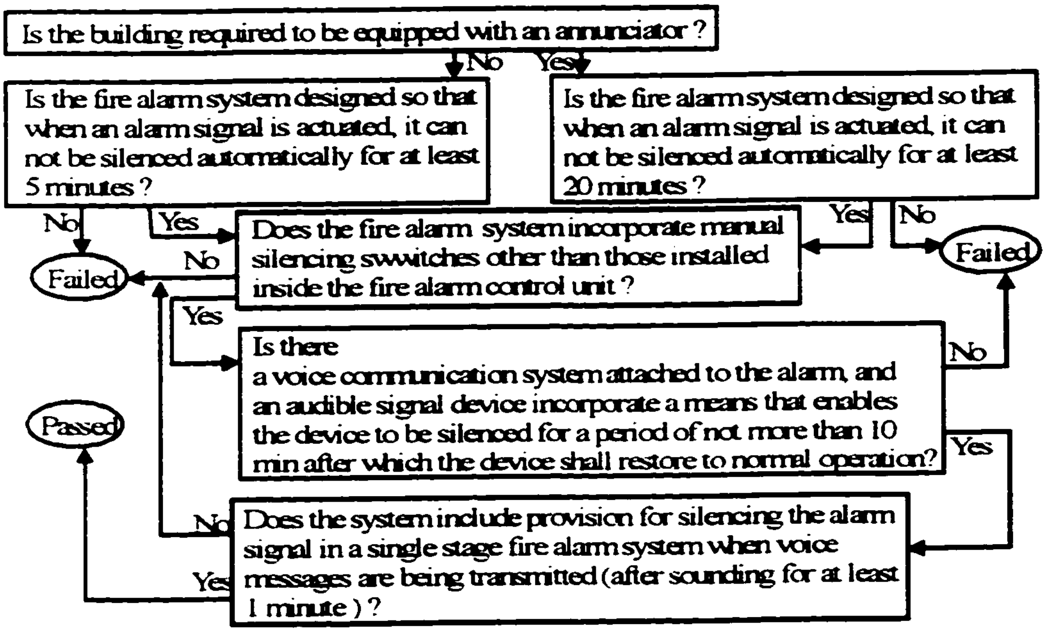
FIRE ALARM SYSTEM REQUIRED (Sentence 3.2.4.1.(1))



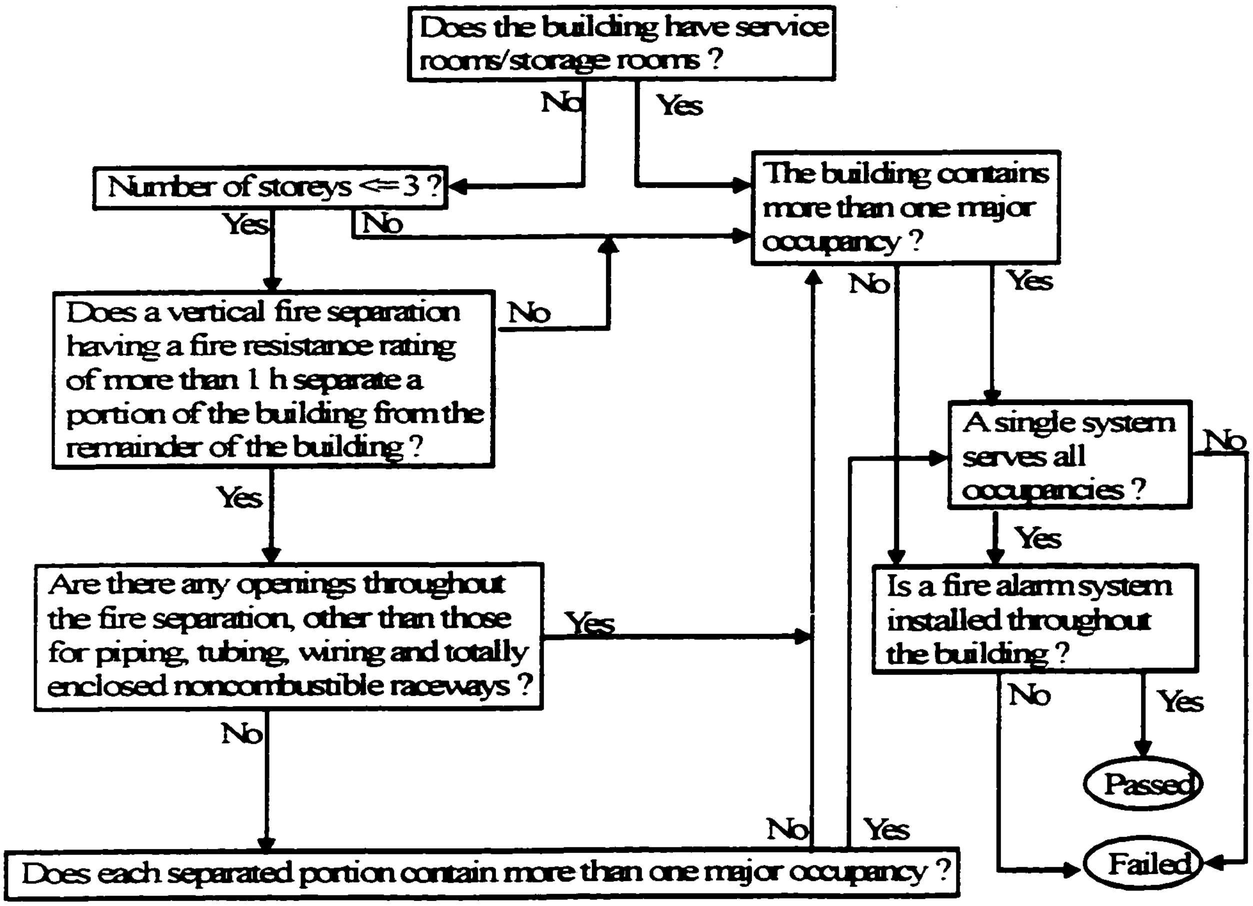
SMOKE DETECTOR REQUIREMENTS (Article 3.2.4.12)



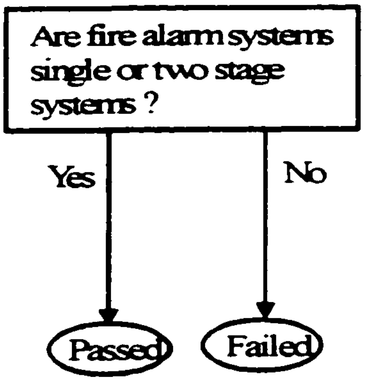
RESTRICTIONS ON FIRE ALARM SYSTEMS (Sentences 3.2.4.6.(1) & (2))



CONTINUITY OF FIRE ALARM SYSTEMS (Article 3.2.4.2)

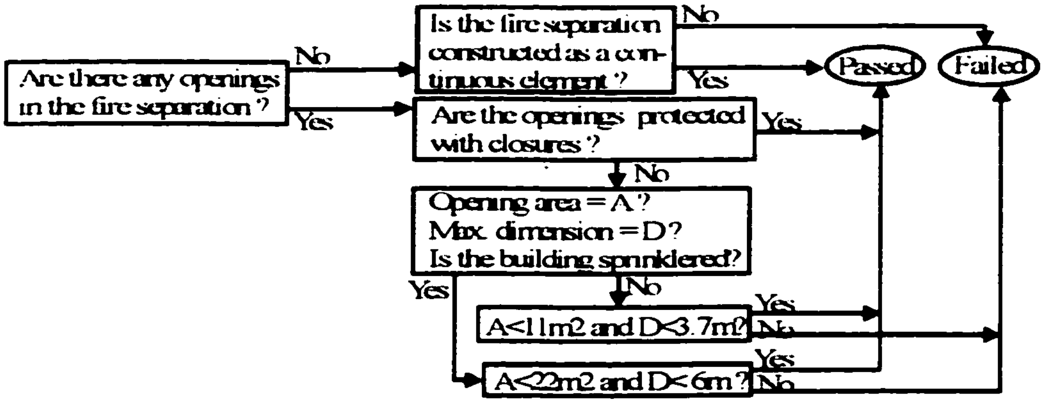


SINGLE OR TWO STAGE FIRE ALARM SYSTEMS (Clause 3.2.4.3. (1d))

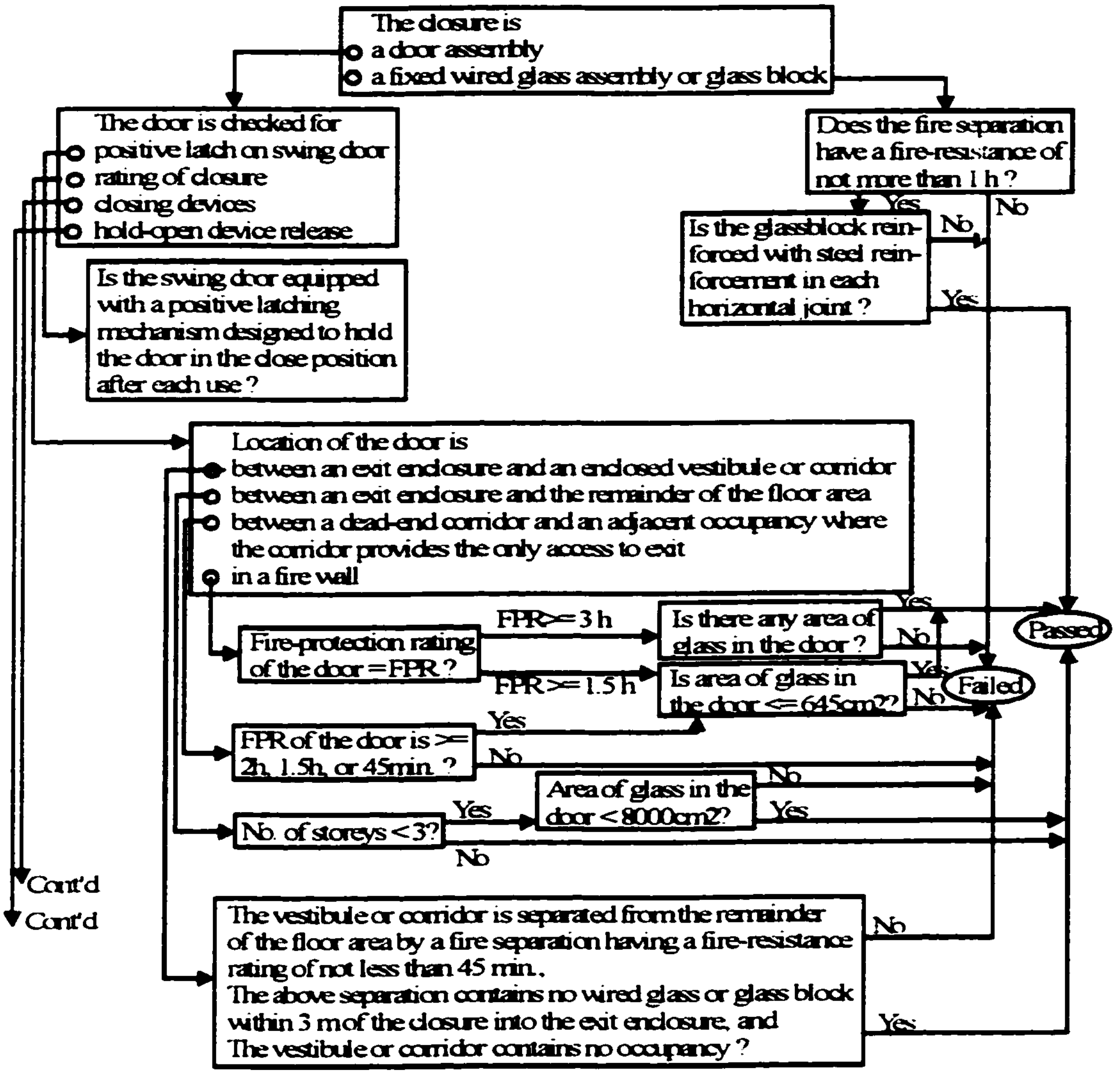


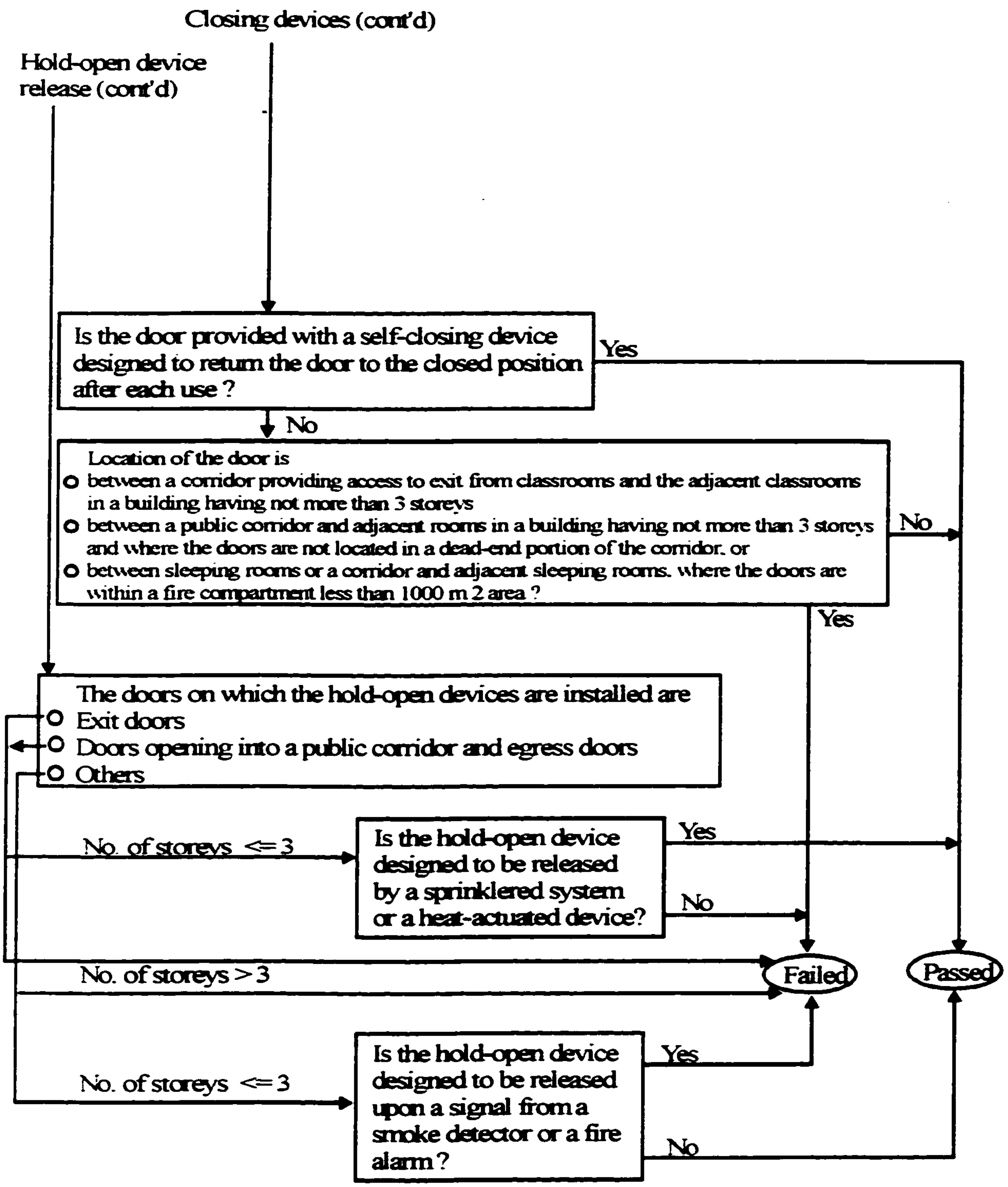
2/ FIRE SEPARATIONS

GENERAL REQUIREMENTS (Article 3.1.8.1)



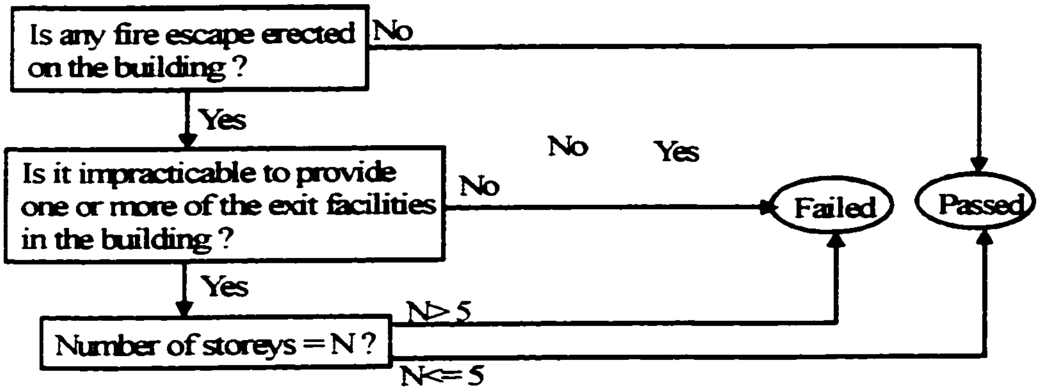
CLOSURES OF OPENINGS IN FIRE SEPARATIONS (Articles 3.1.8.11, 3.1.8.12, & 3.1.8.13)



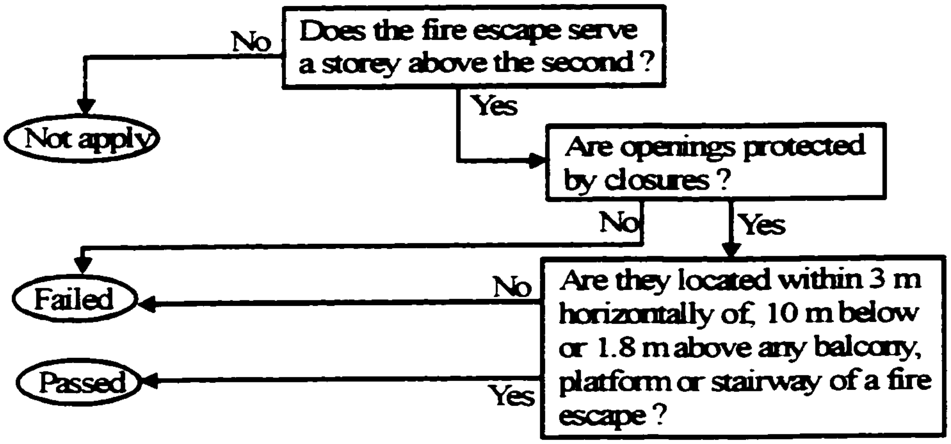

Closing devices (cont'd)
Hold-open device release (cont'd)
Is the door provided with a self-closing device designed to return the door to the closed position after each use ?
Yes
No
Location of the door is
○ between a corridor providing access to exit from classrooms and the adjacent classrooms in a building having not more than 3 storeys
○ between a public corridor and adjacent rooms in a building having not more than 3 storeys and where the doors are not located in a dead-end portion of the corridor, or
○ between sleeping rooms or a corridor and adjacent sleeping rooms, where the doors are within a fire compartment less than 1000 m 2 area ?
No
Yes
The doors on which the hold-open devices are installed are
○ Exit doors
○ Doors opening into a public corridor and egress doors
○ Others
No. of storeys <=3
Is the hold-open device designed to be released by a sprinklered system or a heat-actuated device?
Yes
No
No. of storeys > 3
Failed
Passed
No. of storeys <=3
Is the hold-open device designed to be released upon a signal from a smoke detector or a fire alarm?
Yes
No

3/ FIRE ESCAPES

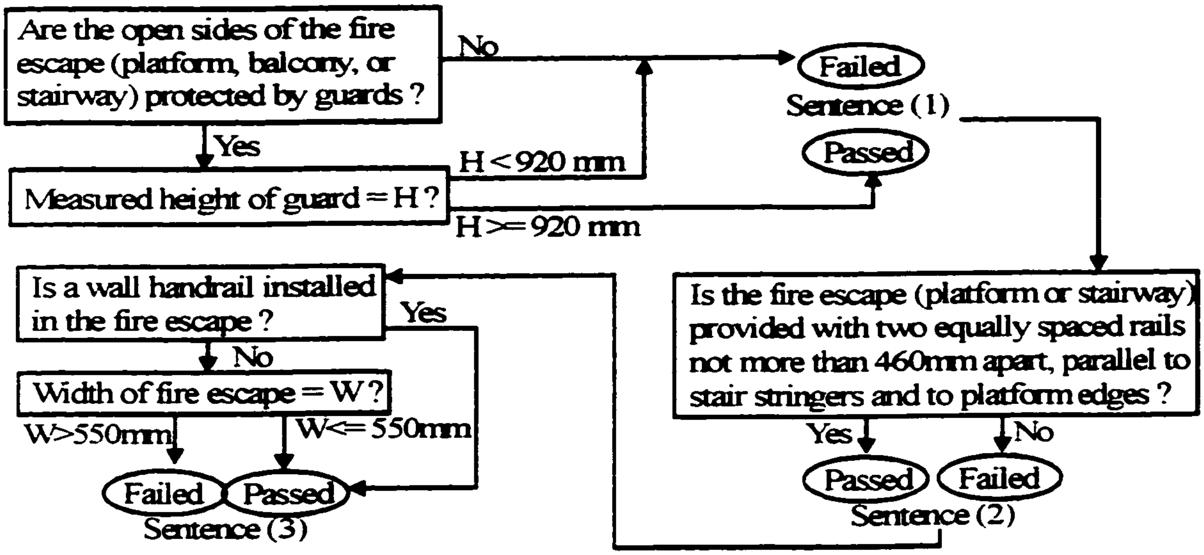
FIRE ESCAPE PROVISIONS (Article 3.4.7.1)



PROTECTION OF EXTERIOR OPENINGS NEAR FIRE ESCAPES (Article 3.4.7.4)

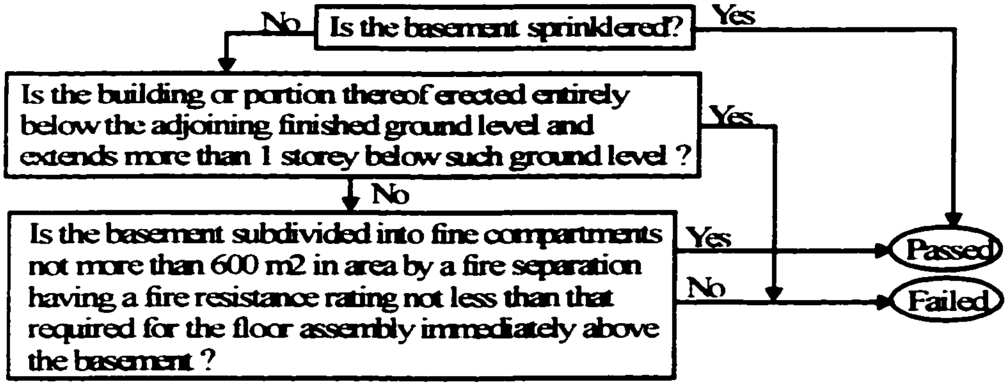


GUARDS AND RAILINGS (Article 3.4.7.6)

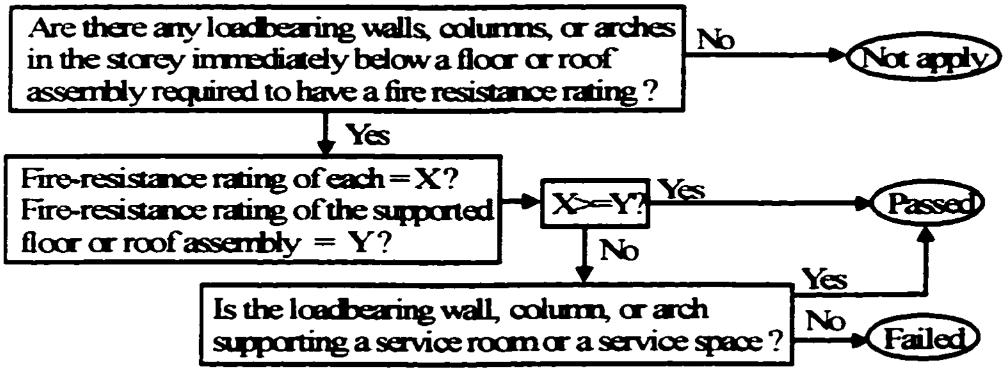


4/ STRUCTURAL FIRE PROTECTION

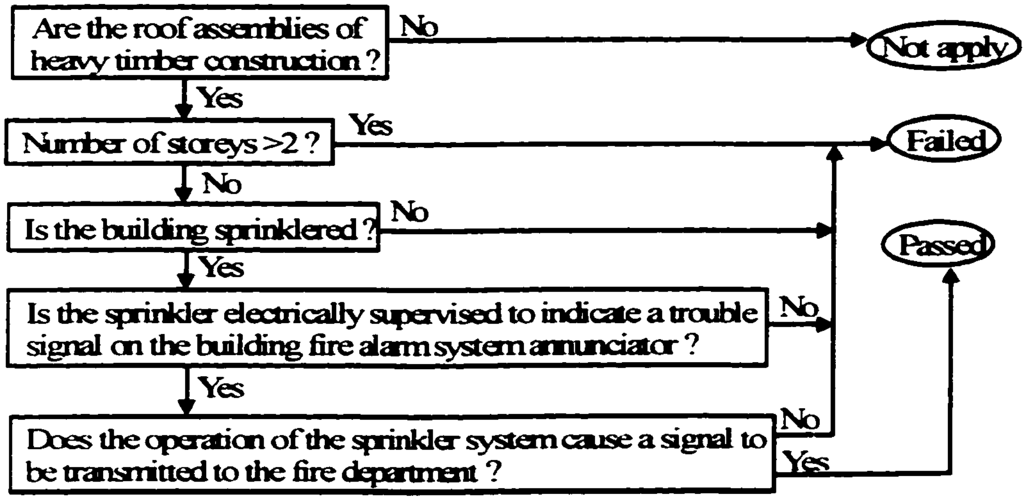
MANDATORY SPRINKLERING OF BASEMENTS (Article 3.2.1.5)



LOAD BEARING CONSTRUCTION SAME RATING AS SUPPORTED CONSTRUCTION (Article 3.1.7.6)

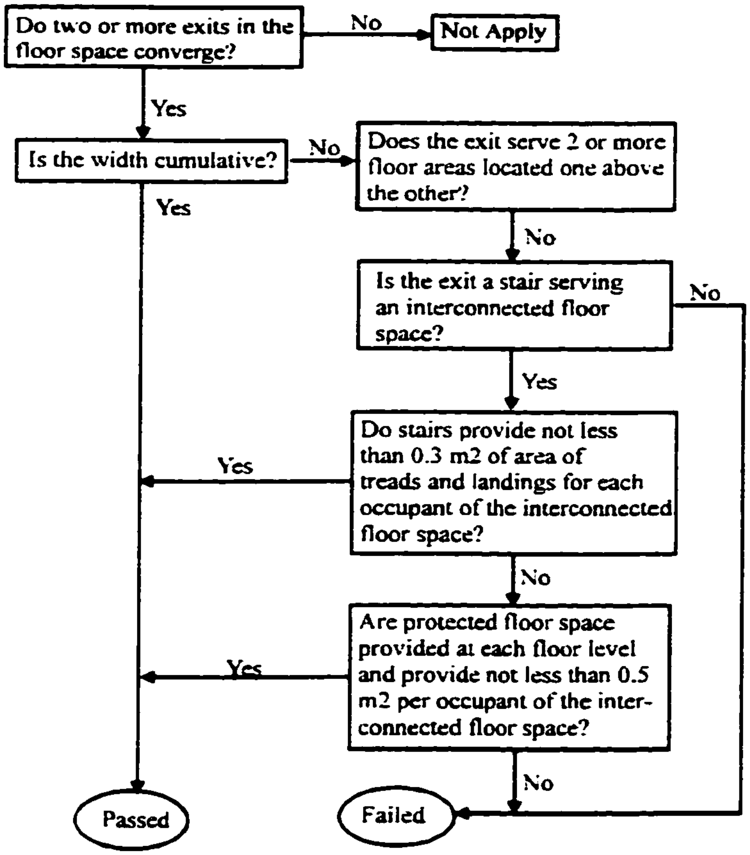


CONDITION TO PERMIT HEAVY TIMBER ROOFS (Article 3.2.2.13)
SPRINKLERS IN LIEU OF ROOF ASSEMBLY RATING (Article 3.2.2.12)

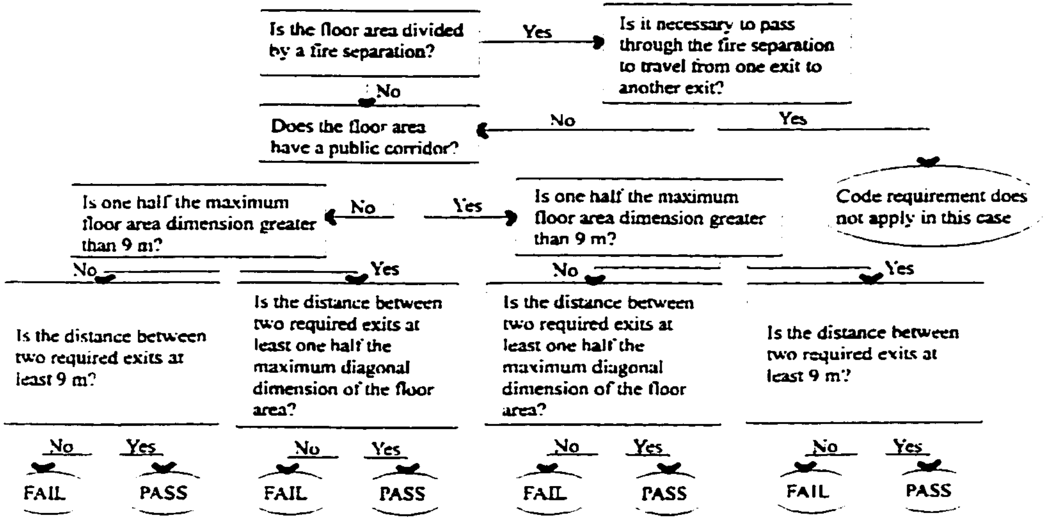


5/ EXIT REQUIREMENTS

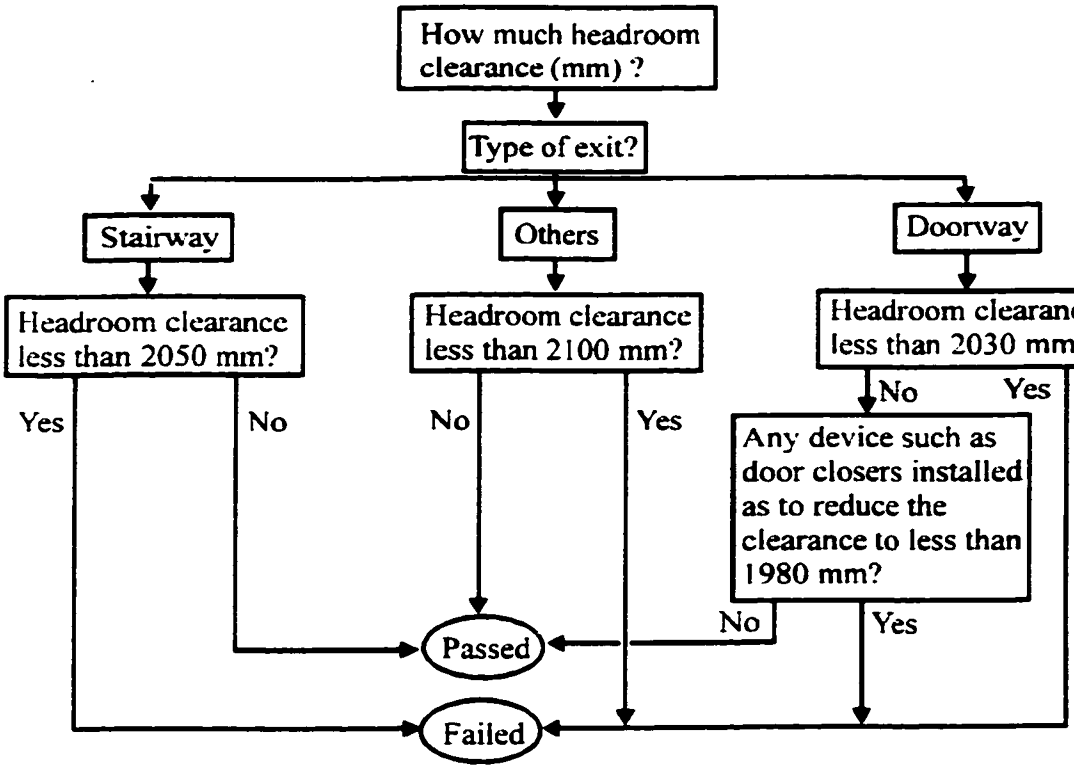
CUMULATIVE EXIT WIDTH (Article 3.4.3.3)
EXITS FROM INTERCONNECTED FLOOR SPACE (Article 3.4.3.4)



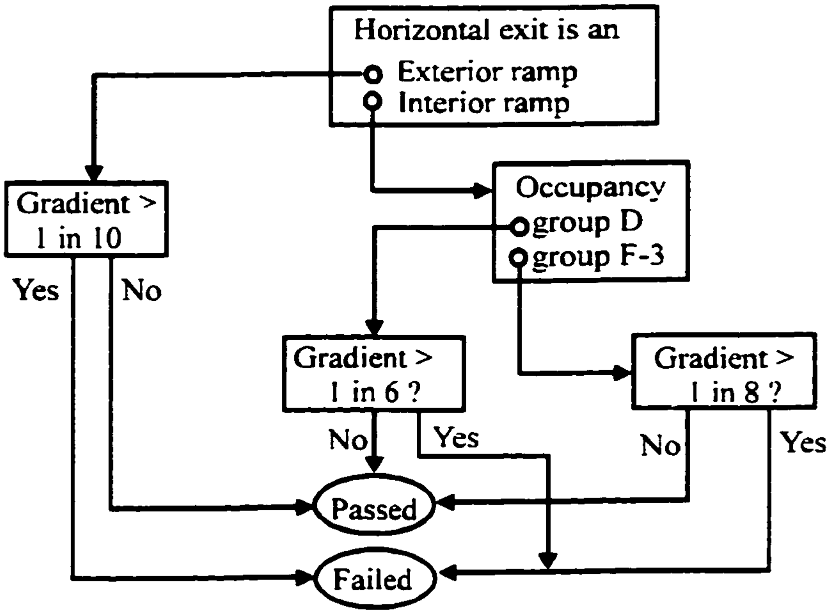
LEAST DISTANCE BETWEEN EXITS



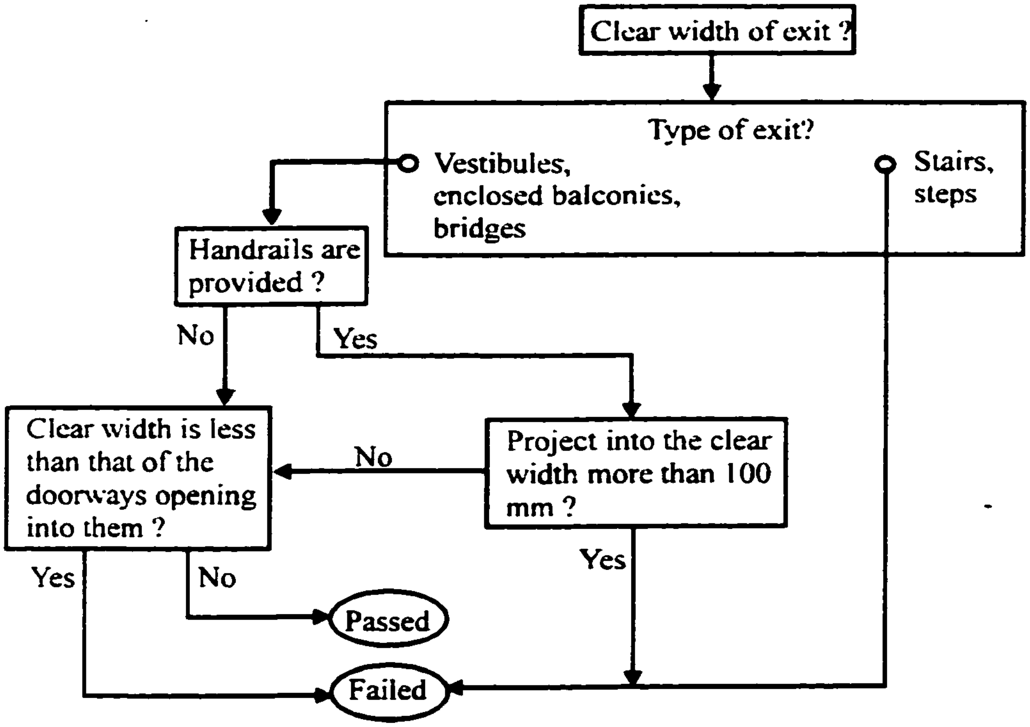
HEADROOM CLEARANCE (Article 3.4.3.7)



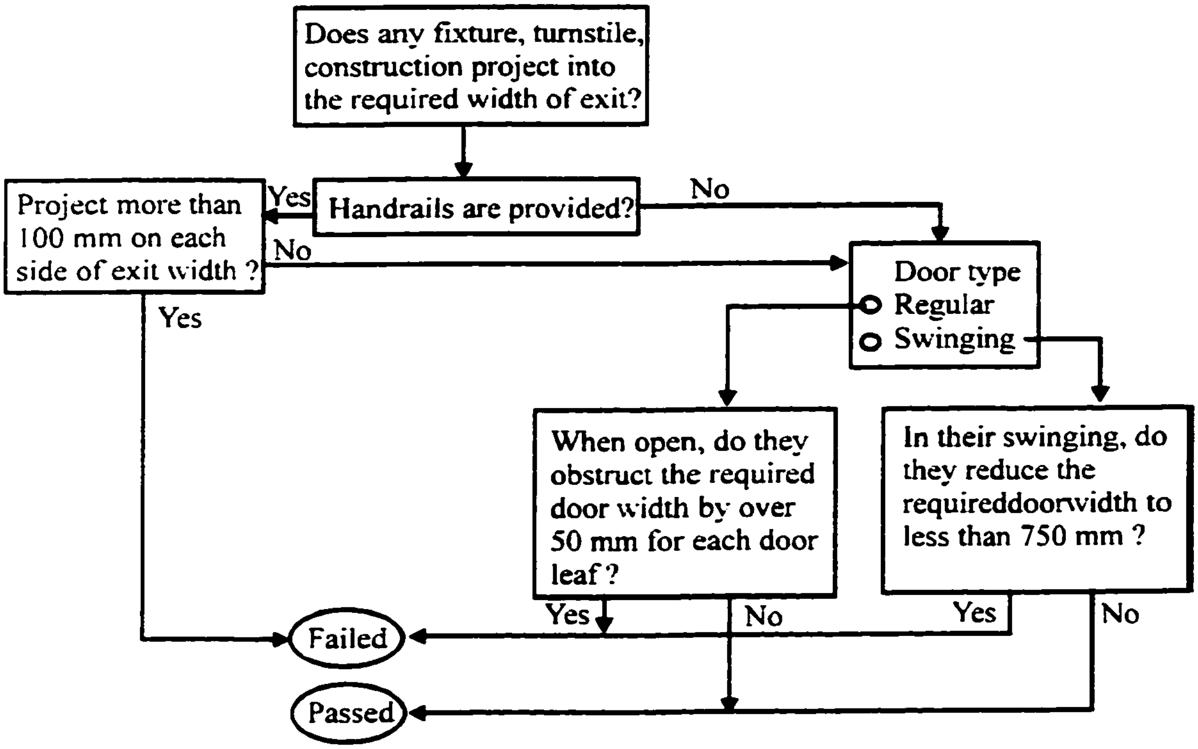
GRADIENTS IN HORIZONTAL EXITS (Article 3.4.6.9)



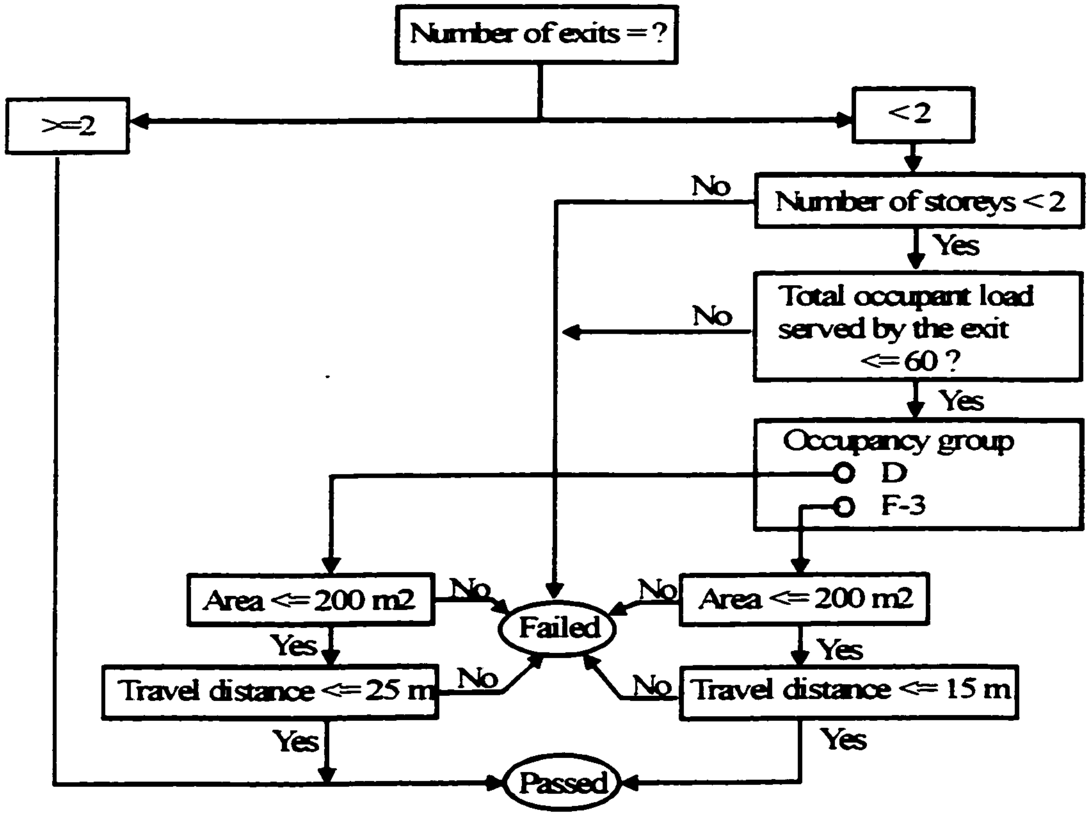
CLEAR WIDTHS ON HORIZONTAL EXITS (Article 3.4.6.9)



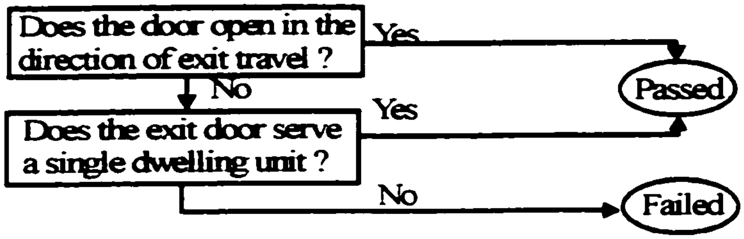
NO PROJECTION INTO REQUIRED WIDTH OF EXITS (Article 3.4.3.6)



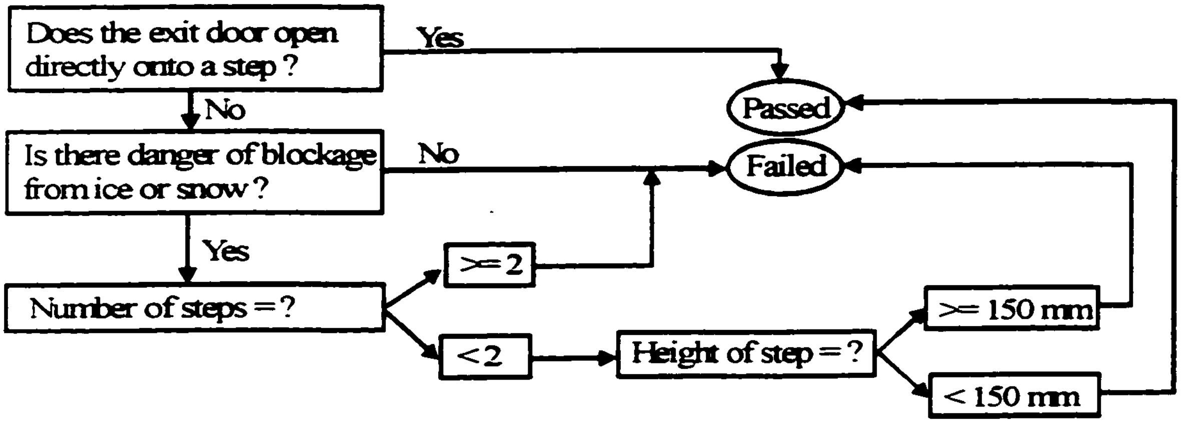
MINIMUM NUMBER OF EXITS (Article 3.4.2.1)



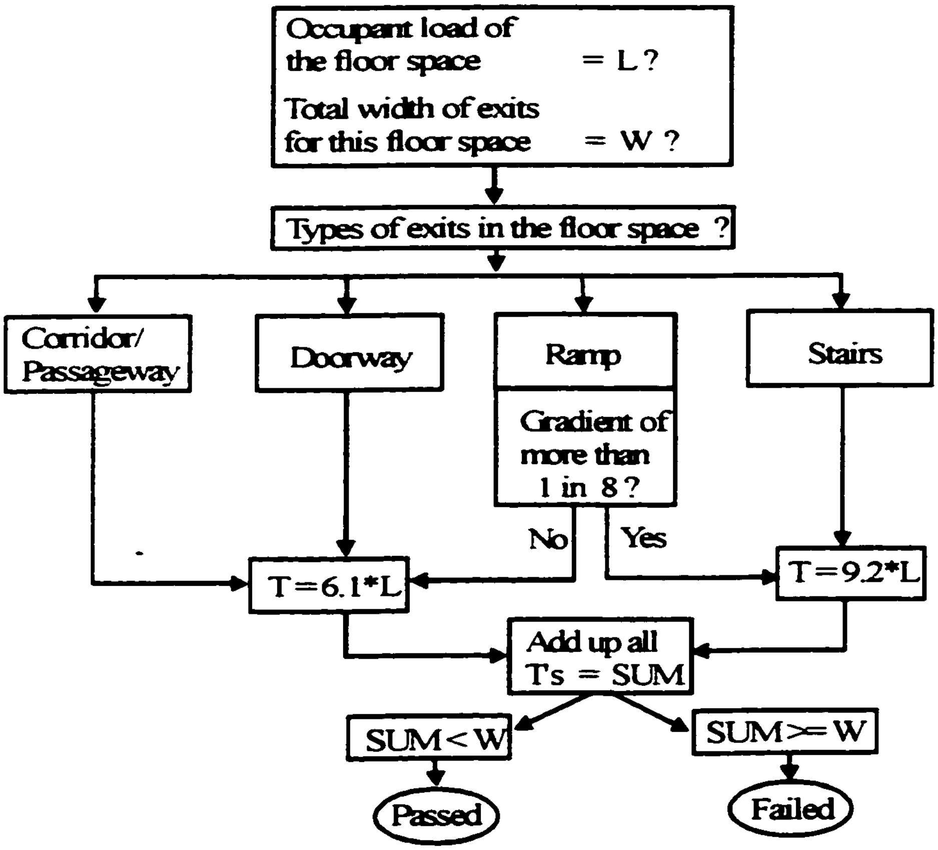
DIRECTION OF OPENING OF EXIT DOORS (Article 3.4.6.11)



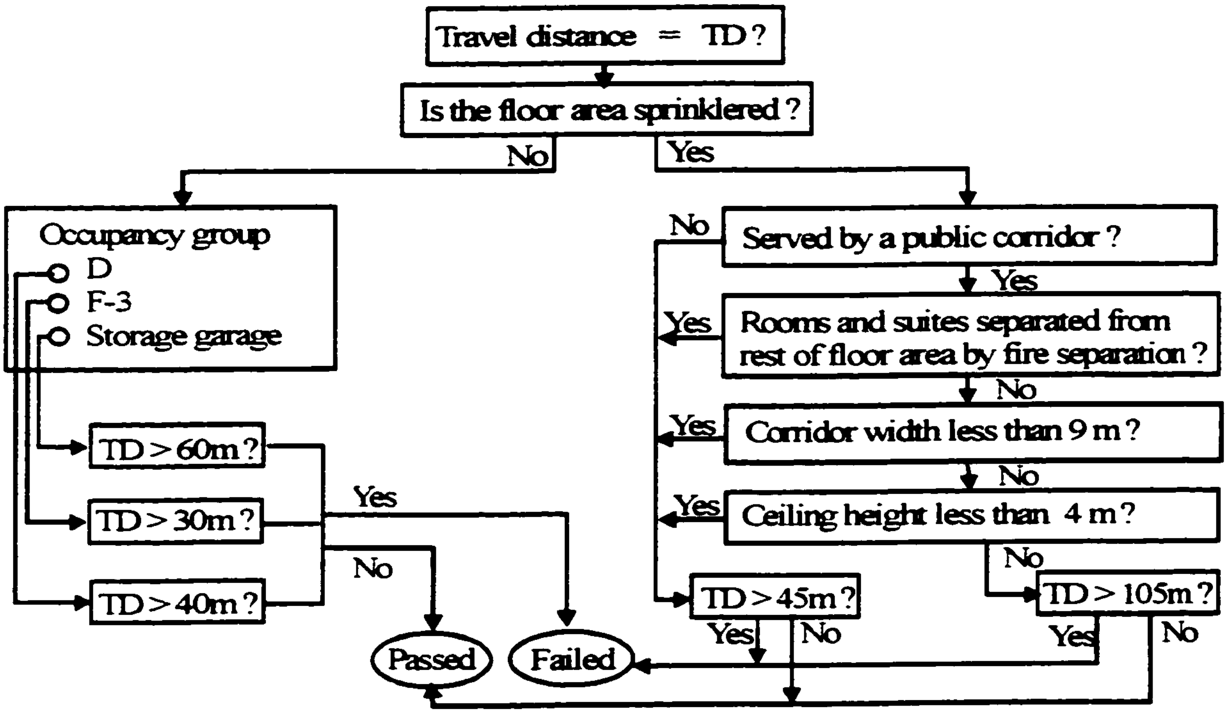
EXIT DOOR TO OPEN ONTO LANDING (Article 3.4.6.10)



EXIT CAPACITY (Article 3.4.3.5)

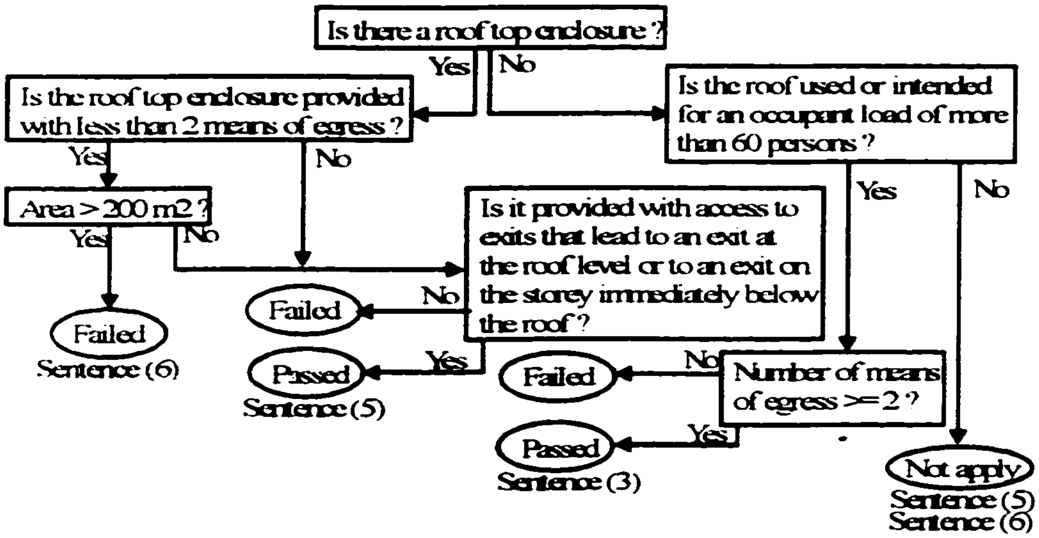


LIMIT ON LOCATION OF SINGLE EXIT (Article 3.4.2.5)

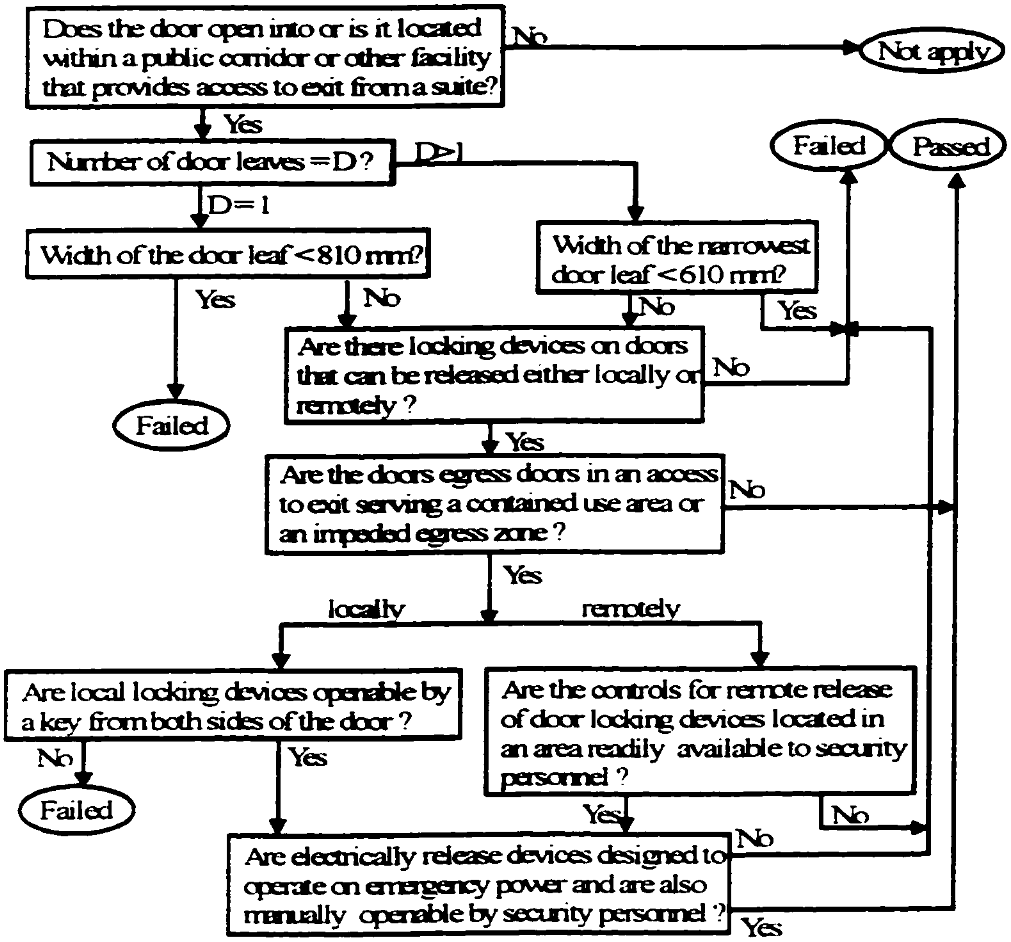


6/ MEANS OF EGRESS

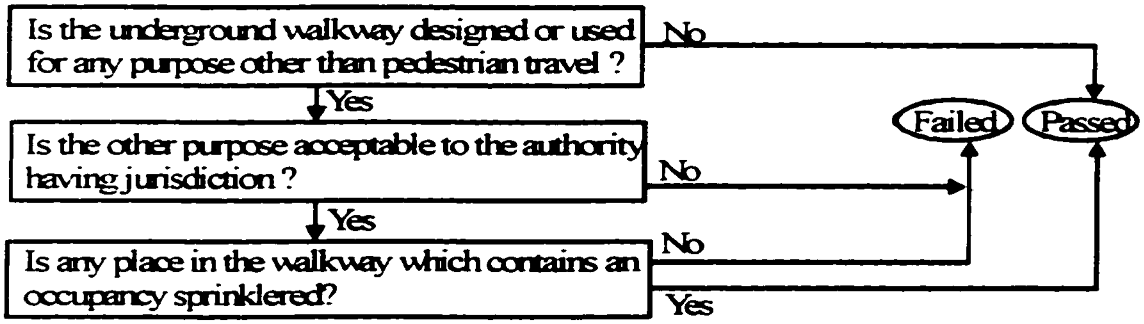
MEANS OF EGRESS FROM ROOF (Sentences 3.3.1.3.(3), (5), & (6))



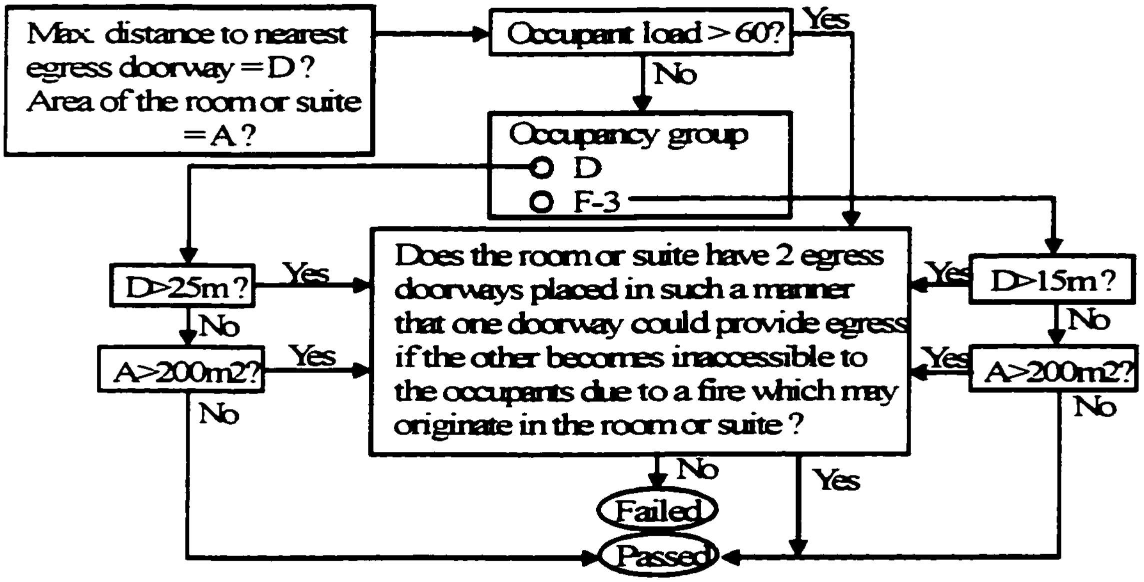
WIDTH OF DOOR LEAVES (Sentence 3.3.1.12.(1))
RESTRICTIONS ON SPECIAL DEVICES FOR LOCKING DOORS (Sentence 3.3.1.12.(2))



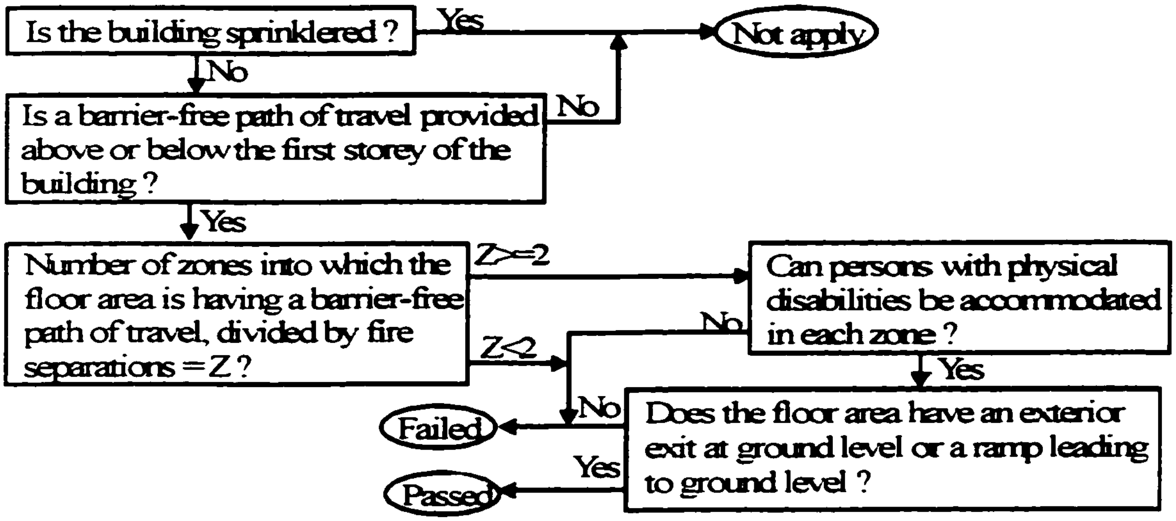
RESTRICTIONS OF OCCUPANCY IN UNDERGROUND WALKWAY
(Sentence 3.2.3.21. (1))



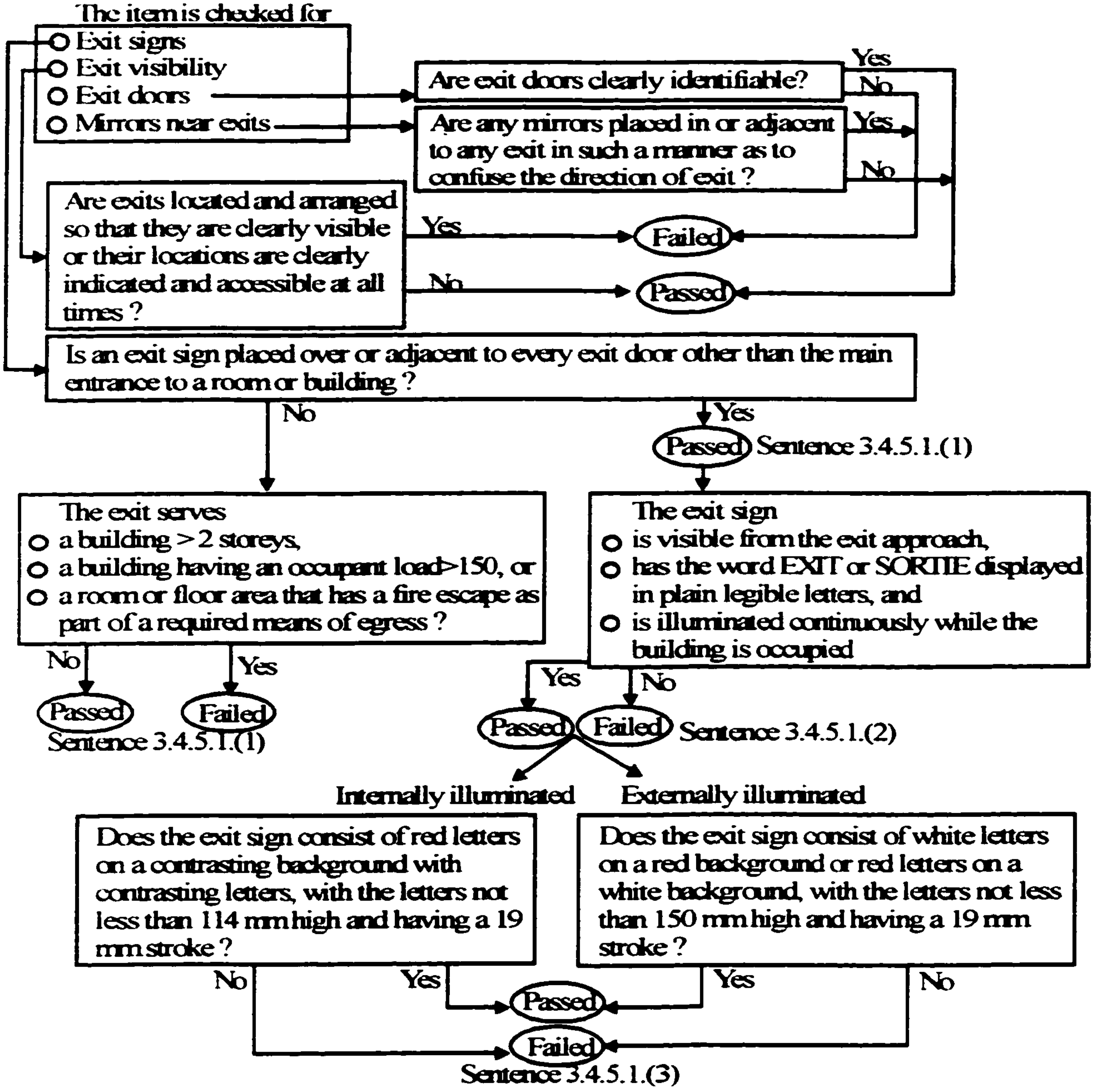
EGRESS FROM SUITES (Article 3.3.1.5)



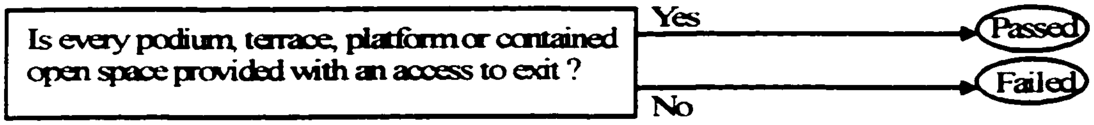
ZONE DIVISION OF FLOORS WITH BARRIER FREE ACCESS (Clause 3.3.1.7.(1b))
EXTERIOR EXIT (Clause 3.3.1.7.(1d))



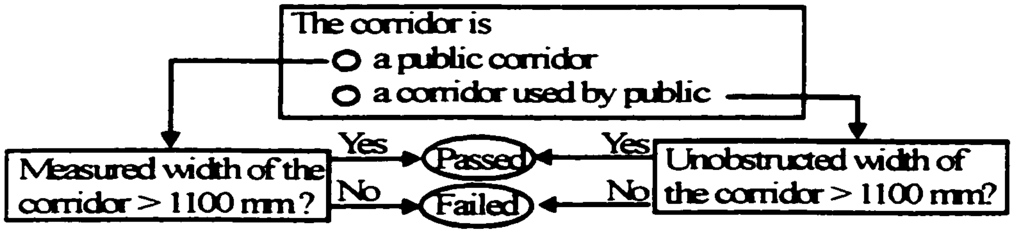
NO MIRRORS ADJACENT TO EXITS (Article 3.4.1.9)
EXIT SIGN REQUIREMENTS (Sentences 3.4.5.(1)&(2))
EXITS CLEARLY VISIBLE AT ALL TIMES (Sentence 3.4.2.5.(4))
EXIT DOORS CLEARLY IDENTIFIABLE (Sentence 3.4.6.10.(3))



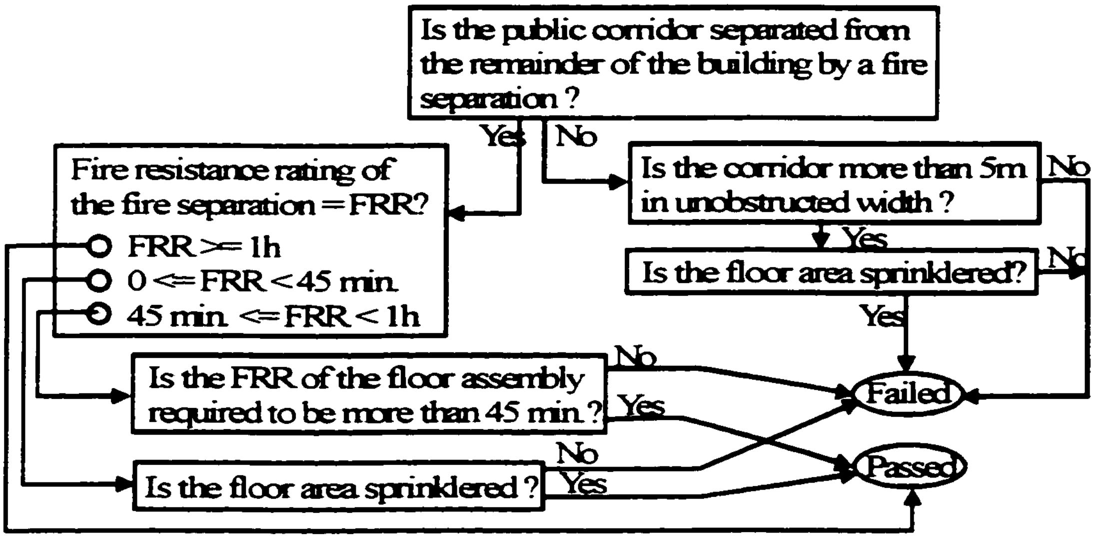
ACCESS TO EXIT FROM ROOFS, PODIUMS, ETC. (Sentence 3.3.1.3.(1))



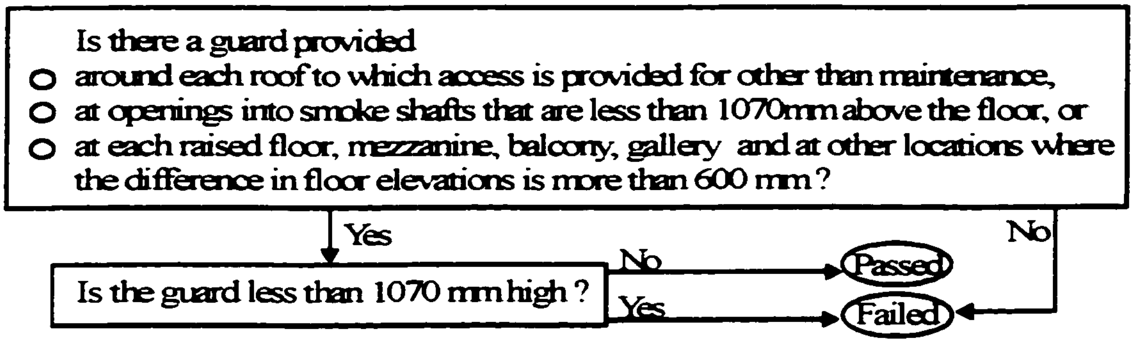
MINIMUM WIDTH OF PUBLIC CORRIDOR (Sentence 3.3.1.9.(1))
MINIMUM WIDTH OF CORRIDOR USED BY THE PUBLIC (Sentence 3.3.1.9.(2))



SEPARATION OF PUBLIC CORRIDORS (Article 3.3.1.4)

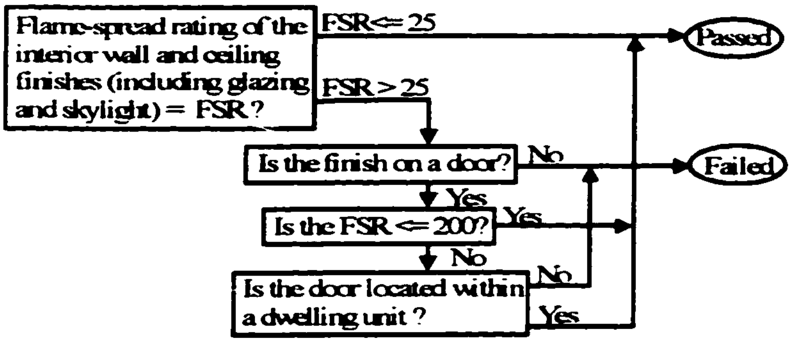


HEIGHT OF GUARDS - ROOFS, SHAFTS, RAISED FLOOR (Sentence 3.3.1.17 (1))

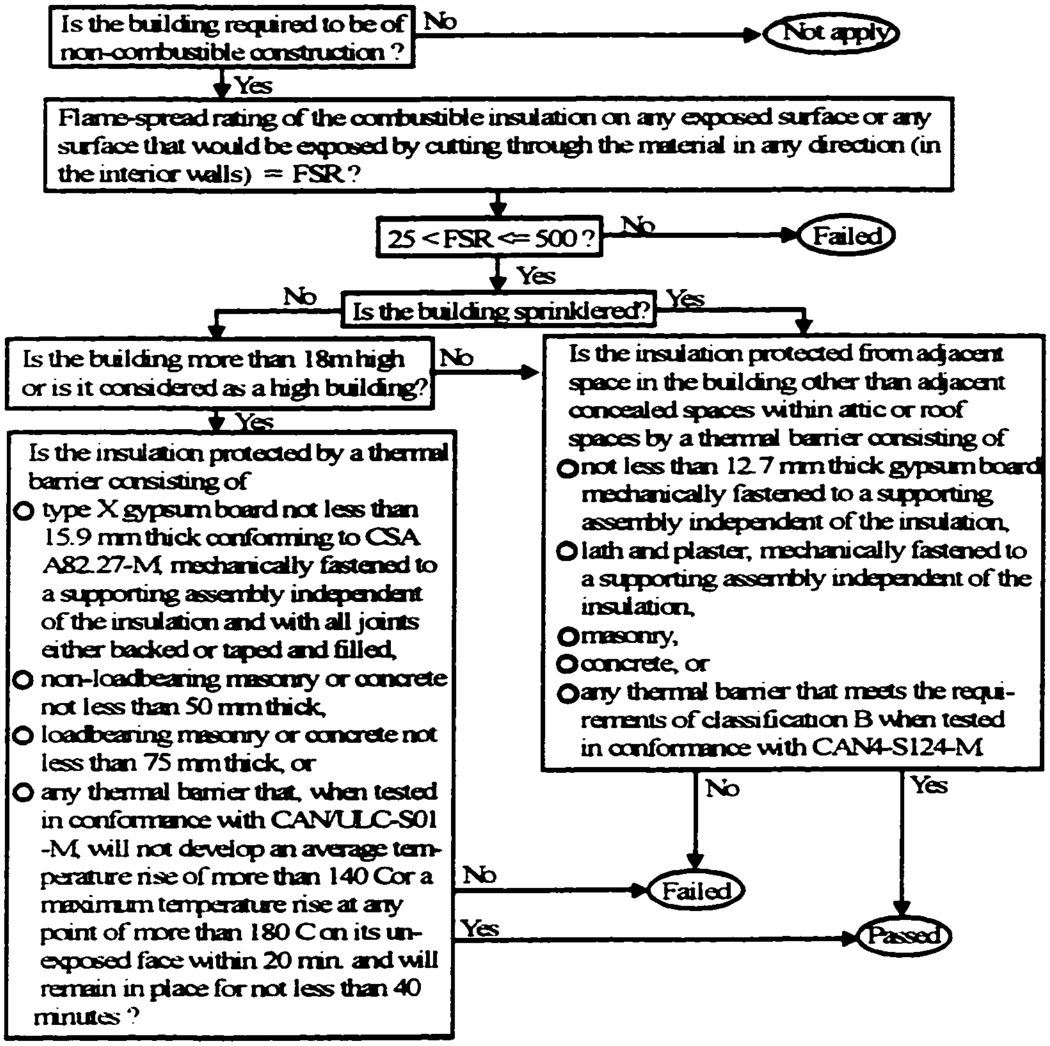


7/ INTERIOR FINISH AND INSULATION PROTECTION

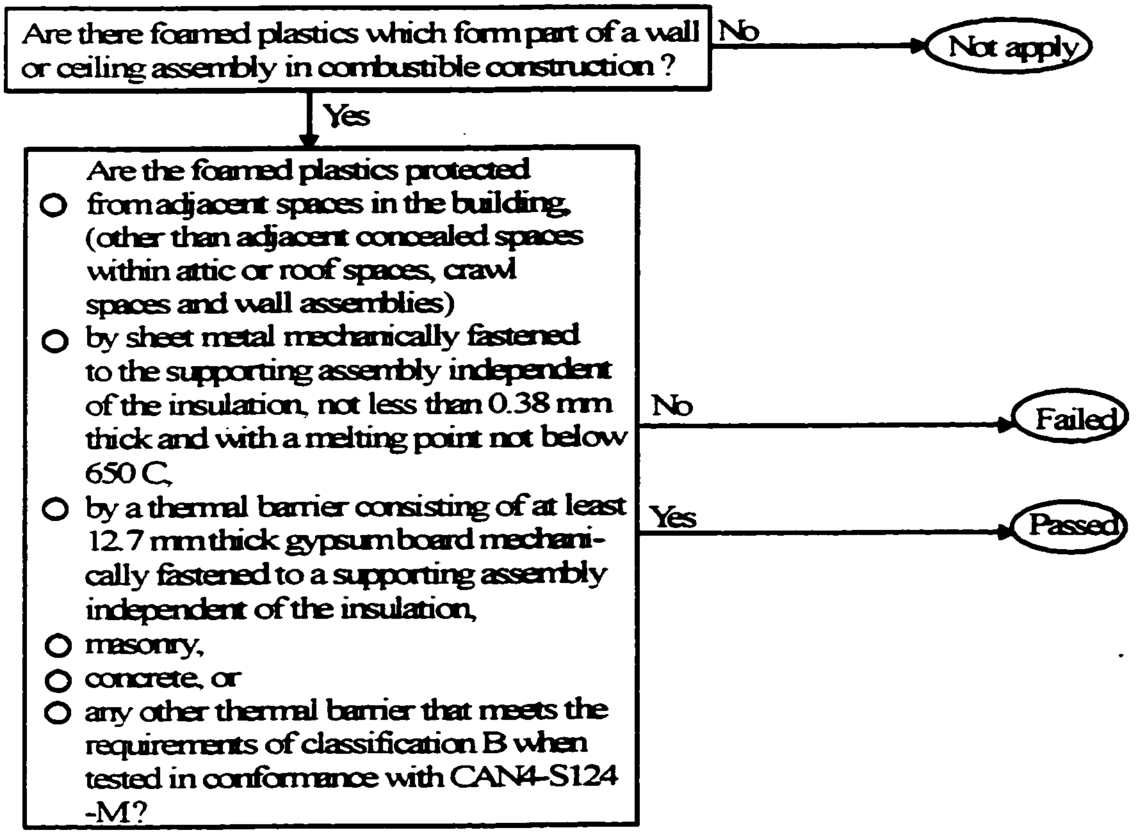
FLAME-SPREAD RATING OF INTERIOR FINISH (Sentences 3.1.13.2.(1)&(2))



PROTECTION OF INSULATION INCLUDING FOAMED PLASTIC (Sentence 3.1.5.11.(4))



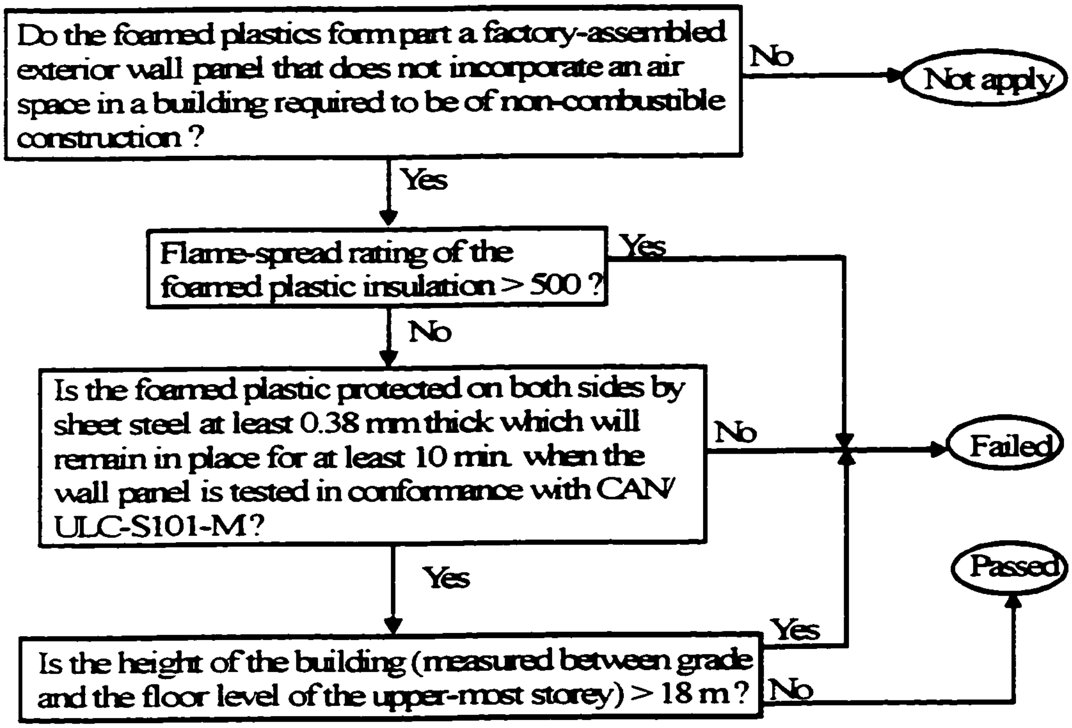
PROTECTION OF FOAMED PLASTIC INSULATION (Article 3.1.4.2)



Are there foamed plastics which form part of a wall or ceiling assembly in combustible construction ?

Are the foamed plastics protected
- ○ from adjacent spaces in the building, (other than adjacent concealed spaces within attic or roof spaces, crawl spaces and wall assemblies)
- ○ by sheet metal mechanically fastened to the supporting assembly independent of the insulation, not less than 0.38 mm thick and with a melting point not below 650 C,
- ○ by a thermal barrier consisting of at least 12.7 mm thick gypsum board mechanically fastened to a supporting assembly independent of the insulation,
- ○ masonry,
- ○ concrete, or
- ○ any other thermal barrier that meets the requirements of classification B when tested in conformance with CAN4-S124-M?

PROTECTION OF FOAMED PLASTIC FACTORY PANELS (Sentence 3.1.5.11.(6))



Do the foamed plastics form part a factory-assembled exterior wall panel that does not incorporate an air space in a building required to be of non-combustible construction ?

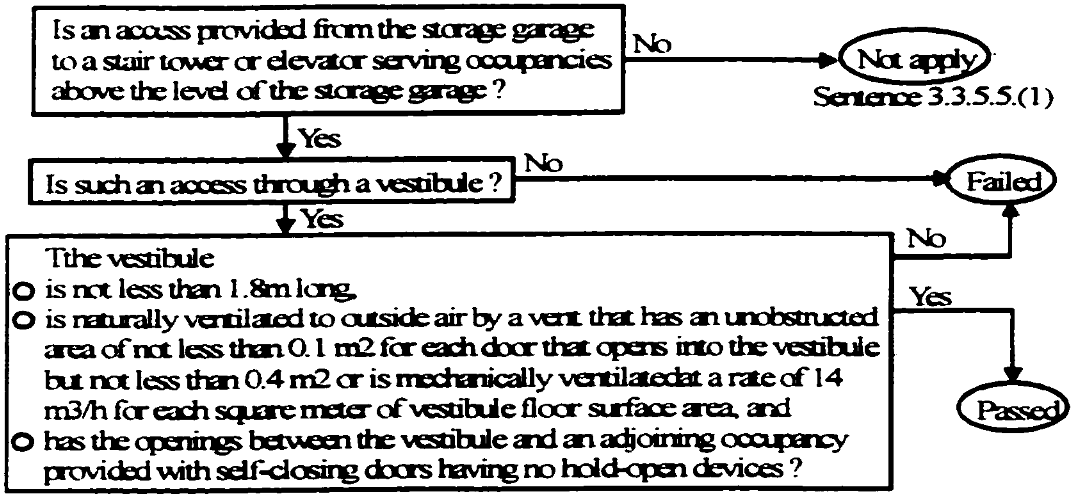
Flame-spread rating of the foamed plastic insulation > 500 ?

Is the foamed plastic protected on both sides by sheet steel at least 0.38 mm thick which will remain in place for at least 10 min. when the wall panel is tested in conformance with CAN/ULC-S101-M?

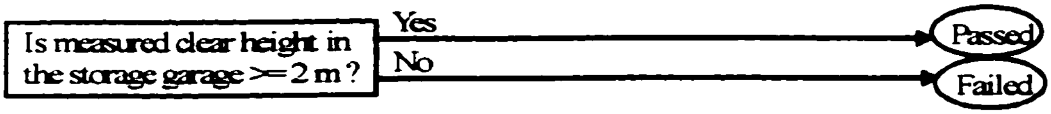
Is the height of the building (measured between grade and the floor level of the upper-most storey) > 18 m ?

8/ STORAGE AND REPAIR GARAGES

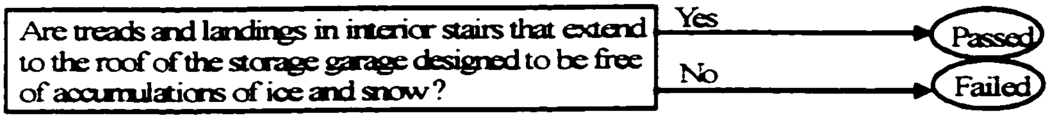
VESTIBULES TO STAIRS IN GARAGES (Sentences 3.3.5.5.(1) & 3.3.5.8.(3))



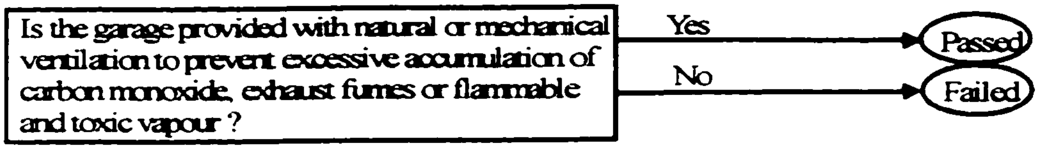
CLEAR HEIGHT IN A STORAGE GARAGE (Sentence 3.3.5.5.(5))



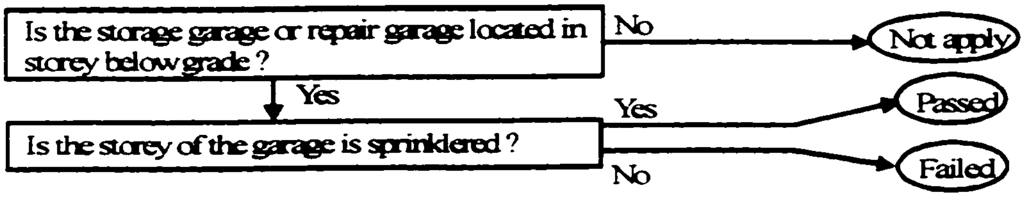
GARAGE STAIRS KEPT FREE OF ICE AND SNOW (Sentence 3.3.5.5.(2))



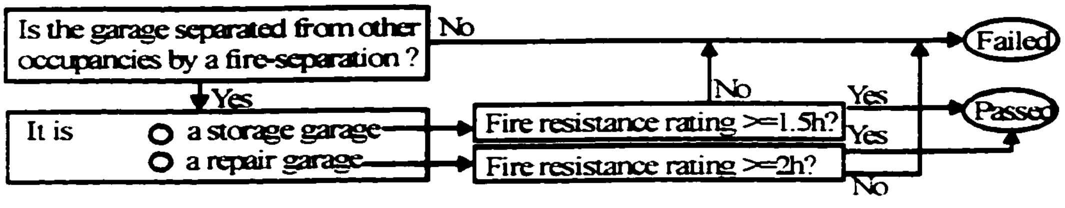
VENTILATION FOR GARAGES (Sentence 3.3.5.5.(4))



SPRINKLERING FOR GARAGES (Sentence 3.3.5.5.(8))

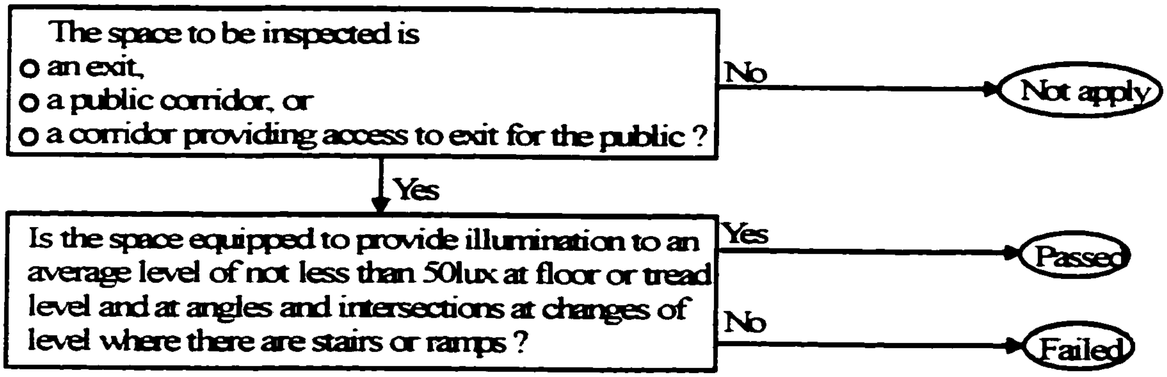


SEPARATION OF GARAGES (Articles 3.3.5.6 & 3.3.5.7)

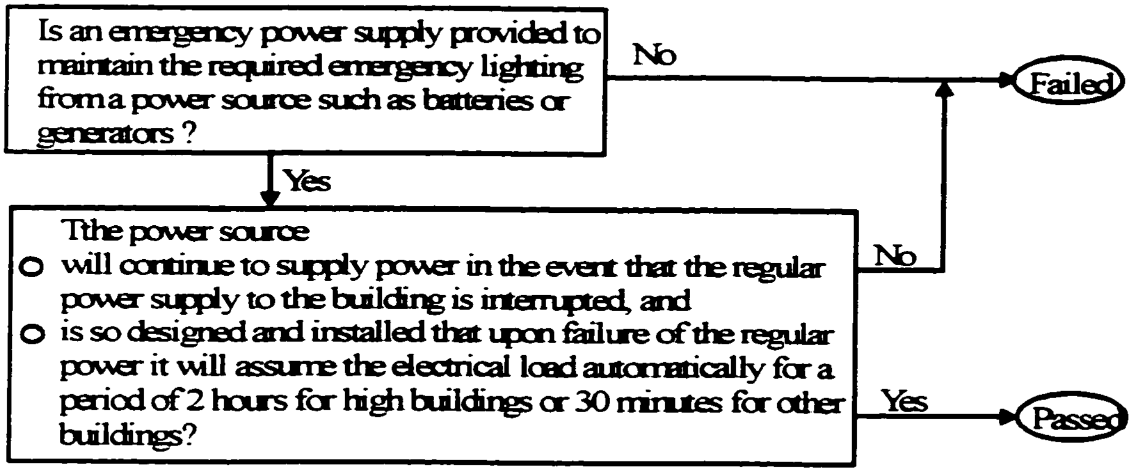


9/ EMERGENCY LIGHTING

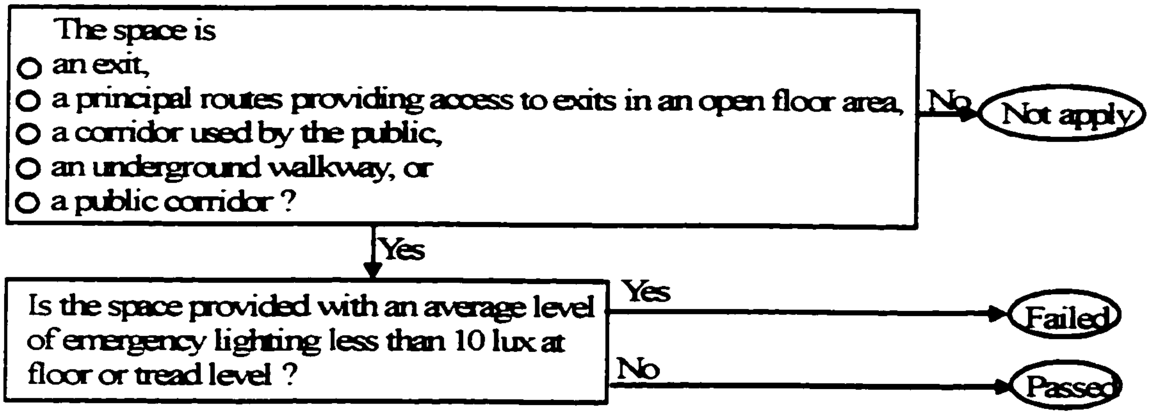
MINIMUM LIGHTING LEVELS IN CORRIDORS AND EXITS (Article 3.2.7.1)



PROVISIONS OF EMERGENCY LIGHTING (Article 3.2.7.4)

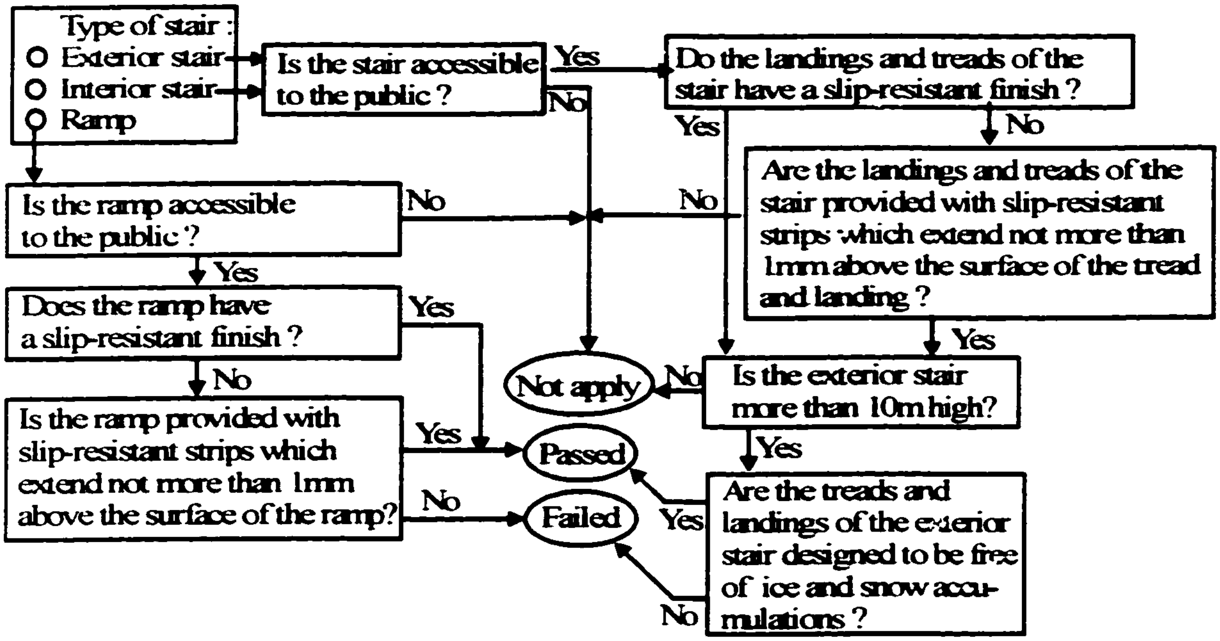


EMERGENCY LIGHTING FOR CORRIDORS (Article 3.2.7.3)

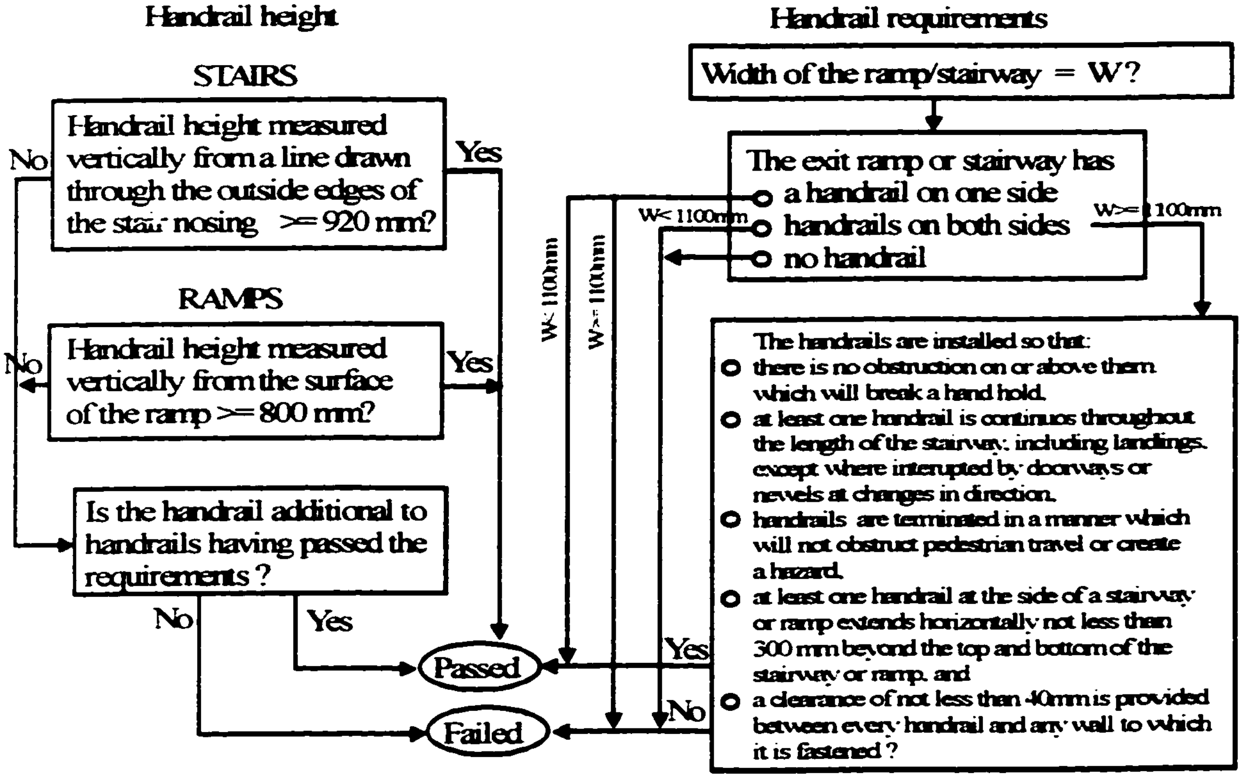


10. STAIR REQUIREMENTS

SLIP-RESISTANT FINISH ON TREADS & LANDINGS EXTERIOR STAIRS OVER 10M FREE OF ICE (Article 3.4.6.1)

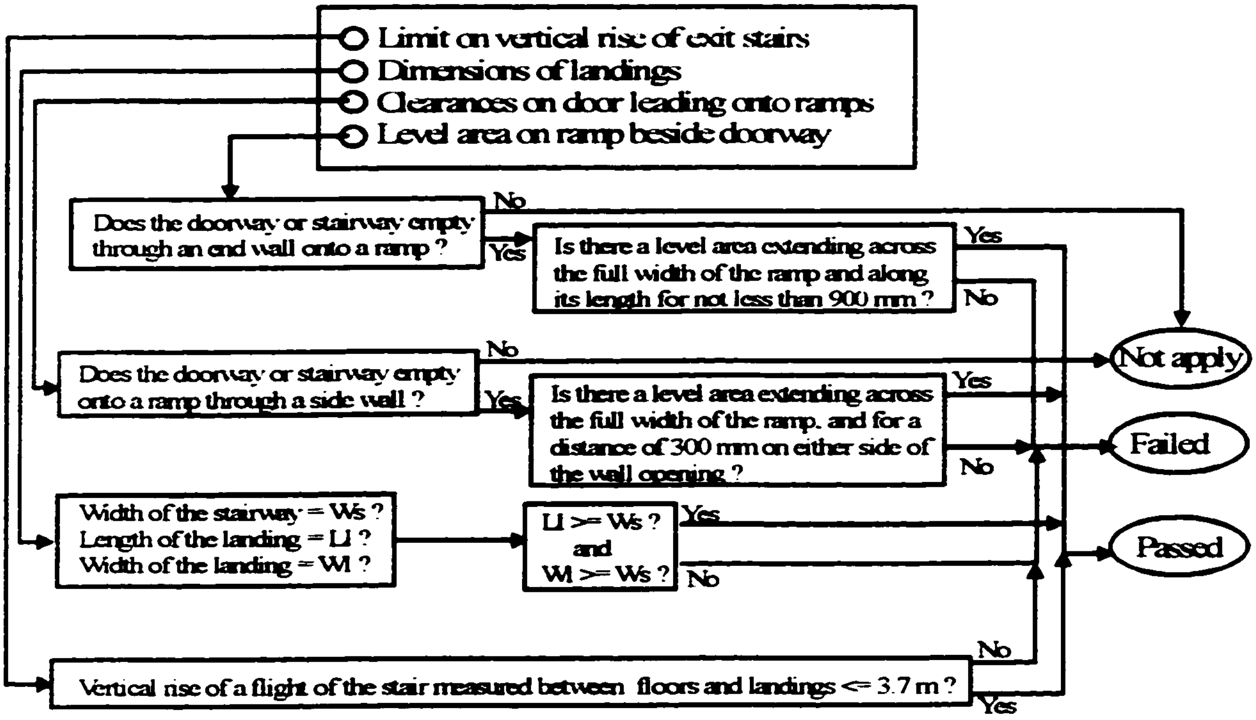


HANDRAIL REQUIREMENTS FOR STAIRS & RAMPS (Article 3.4.6.4)

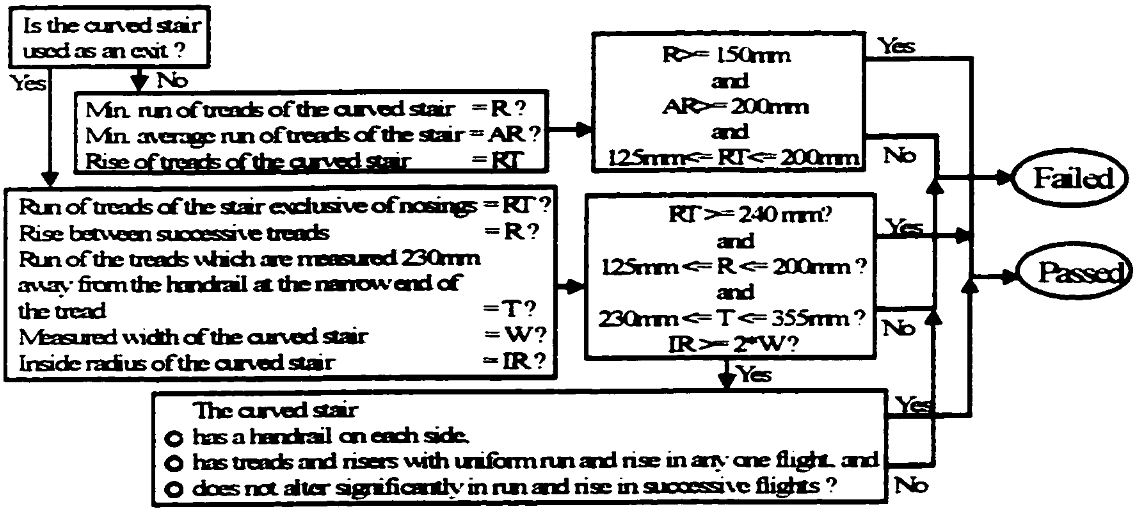


LANDINGS AND MAXIMUM VERTICAL RISE OF STAIR FLIGHTS (Article 3.4.6.3)

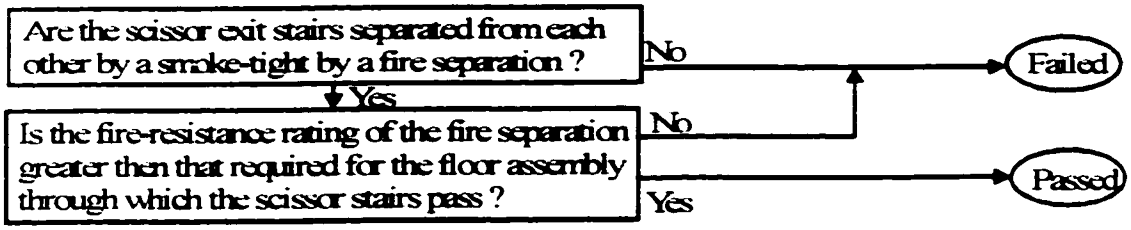
The item is checked for



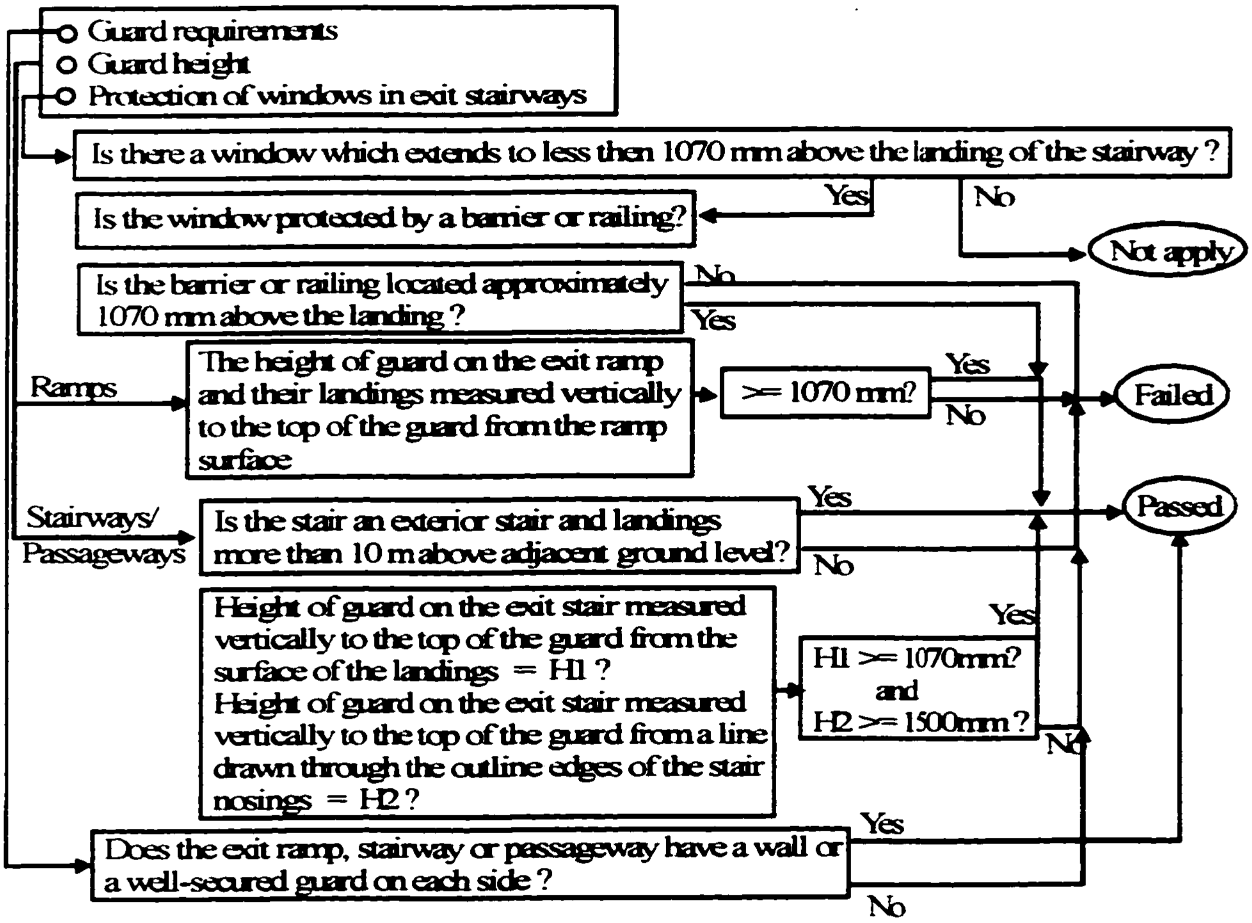
CURVED STAIR REQUIREMENTS (Articles 3.4.6.8 & 3.3.1.15)



SMOKE TIGHTNESS OF SCISSOR STAIRS (Sentence 3.4.4.4.(2))

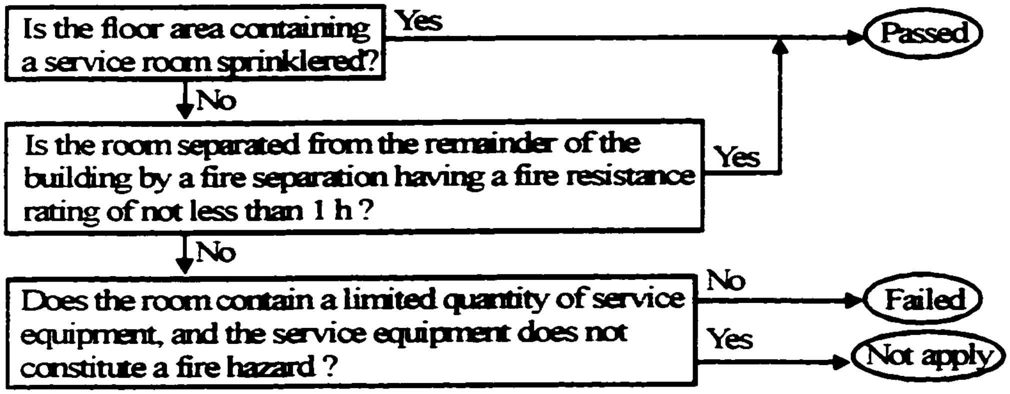


GUARD REQUIREMENTS FOR STAIR RAMPS AND PASSAGEWAYS (Article 3.4.6.5)

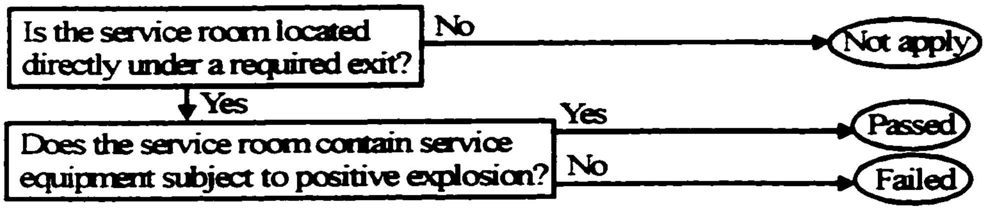


11/ SERVICE FACILITIES

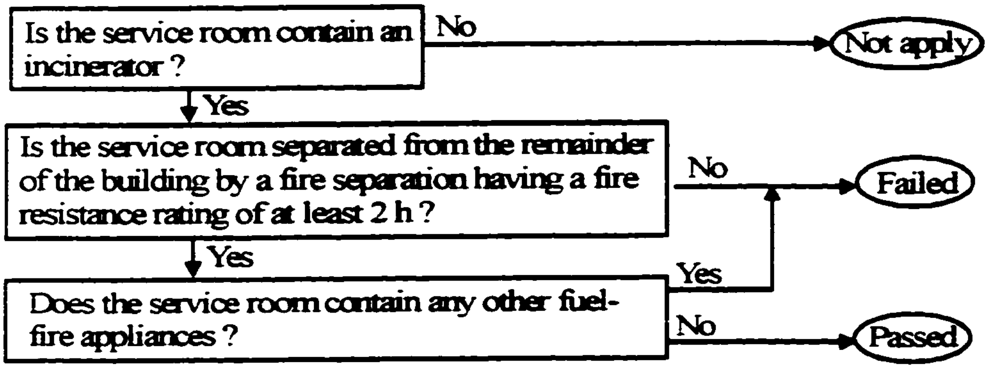
SEPARATION OF GENERAL SERVICE ROOMS (Sentences 3.5.2.1.(3) & 3.5.2.2.(3))



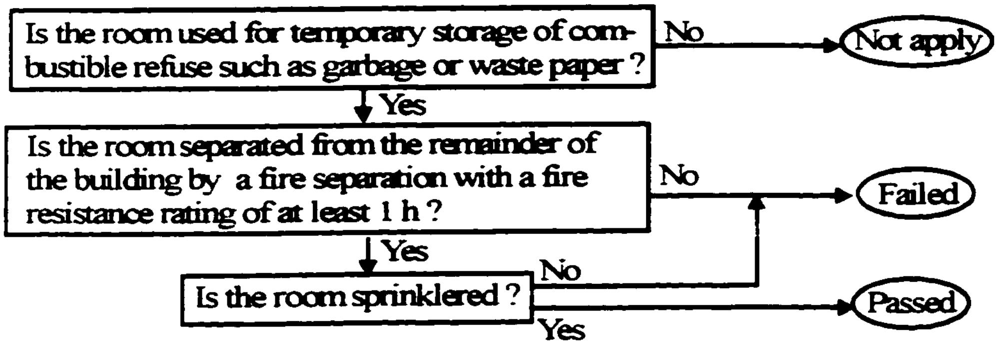
NO HAZARDOUS SERVICE ROOMS UNDER EXITS (Article 3.5.2.4)



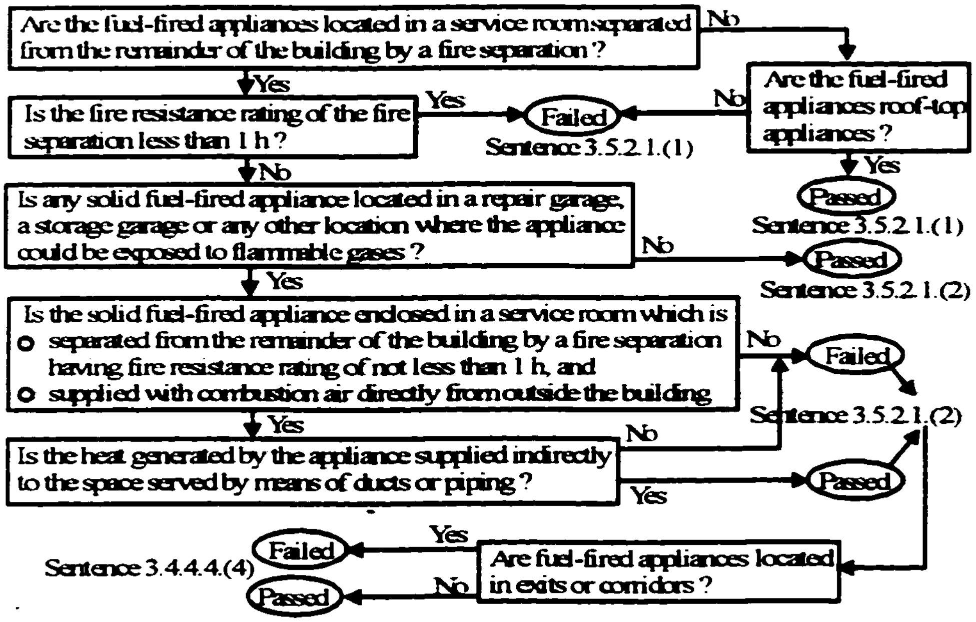
SEPARATION OF SERVICE ROOMS WITH INCINERATORS (Article 3.5.2.6)



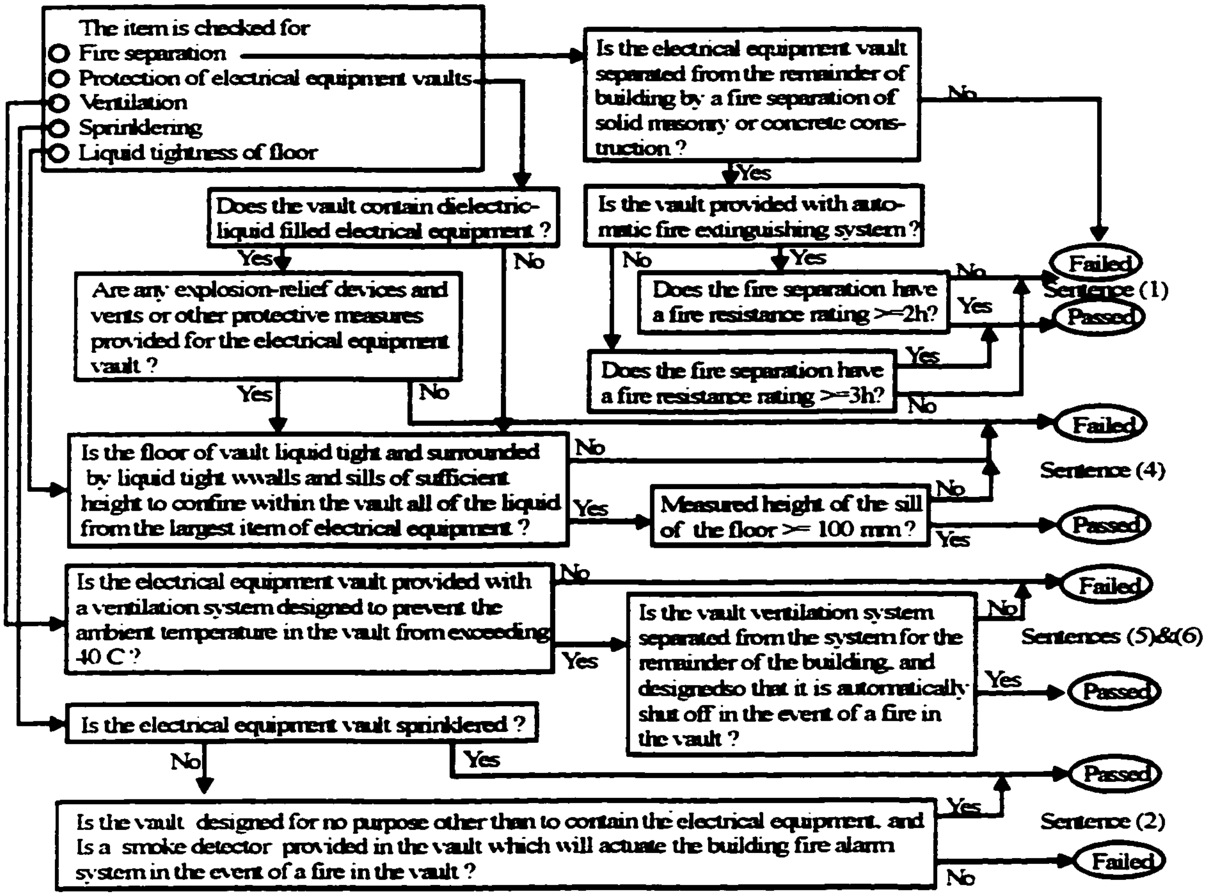
SEPARATION OF ROOM FOR TEMPORARY STORAGE (Article 3.5.2.7)



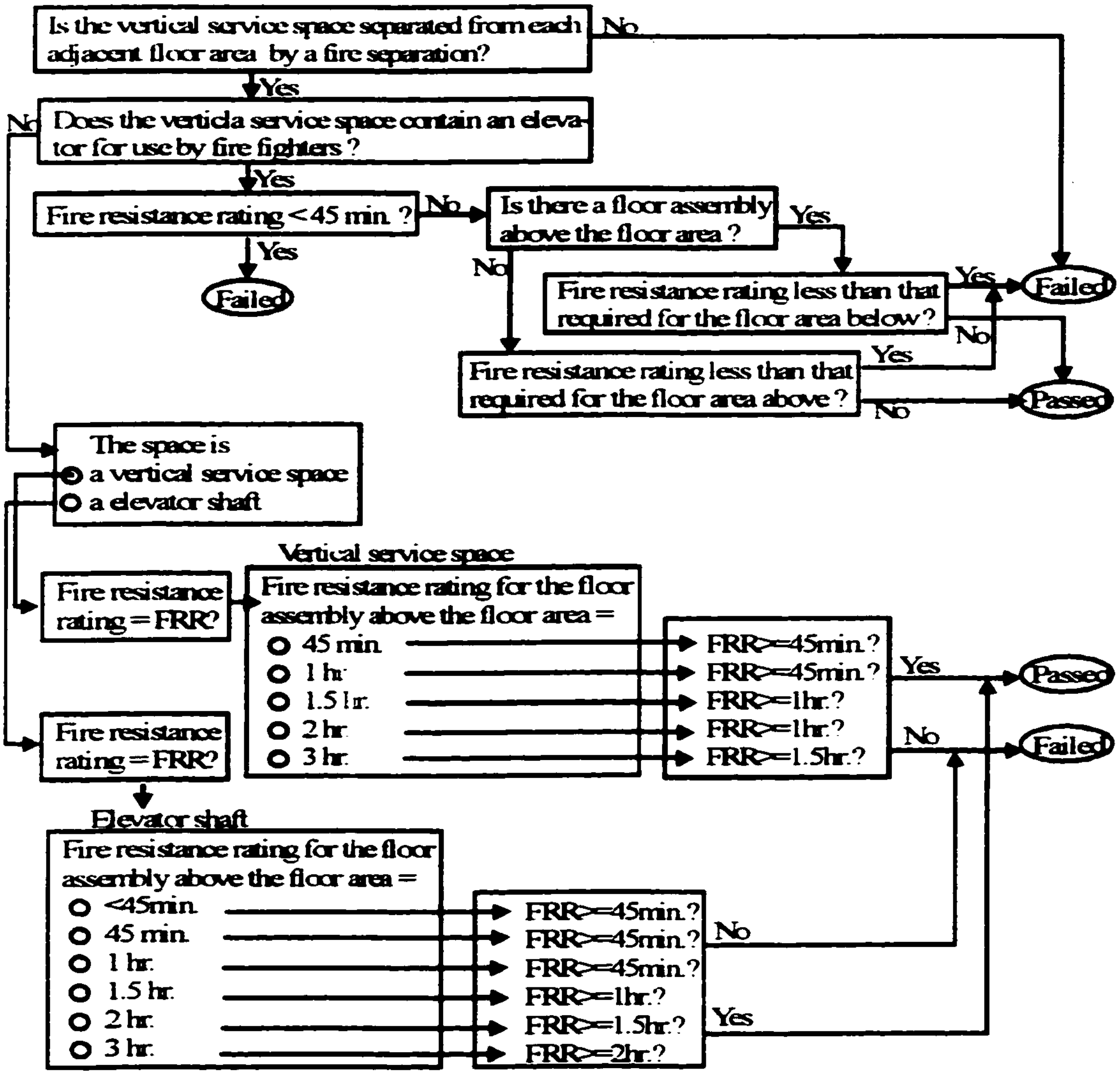
FUEL-FIRED APPLIANCES IN SERVICE ROOMS (Article 3.5.2.1)
NO FUEL-FIRED APPLIANCES IN EXITS OR CORRIDORS (Sentence 3.4.4.4.(4))



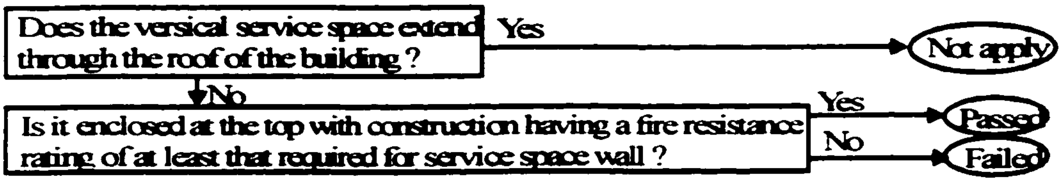
ELECTRICAL EQUIPMENT VAULTS (Article 3.5.2.9)



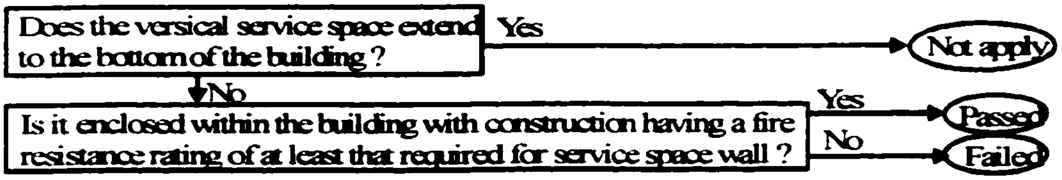
SEPARATION OF VERTICAL SERVICE SPACES (Sentence 3.5.3.1.(1))
SEPARATION OF SHAFTS WITH FIRE FIGHTERS ELEVATORS (Sentence 3.5.3.1.(2))



SEPARATION AT TOP OF VERTICAL SERVICE SPACE (Sentence 3.5.3.1.(3))

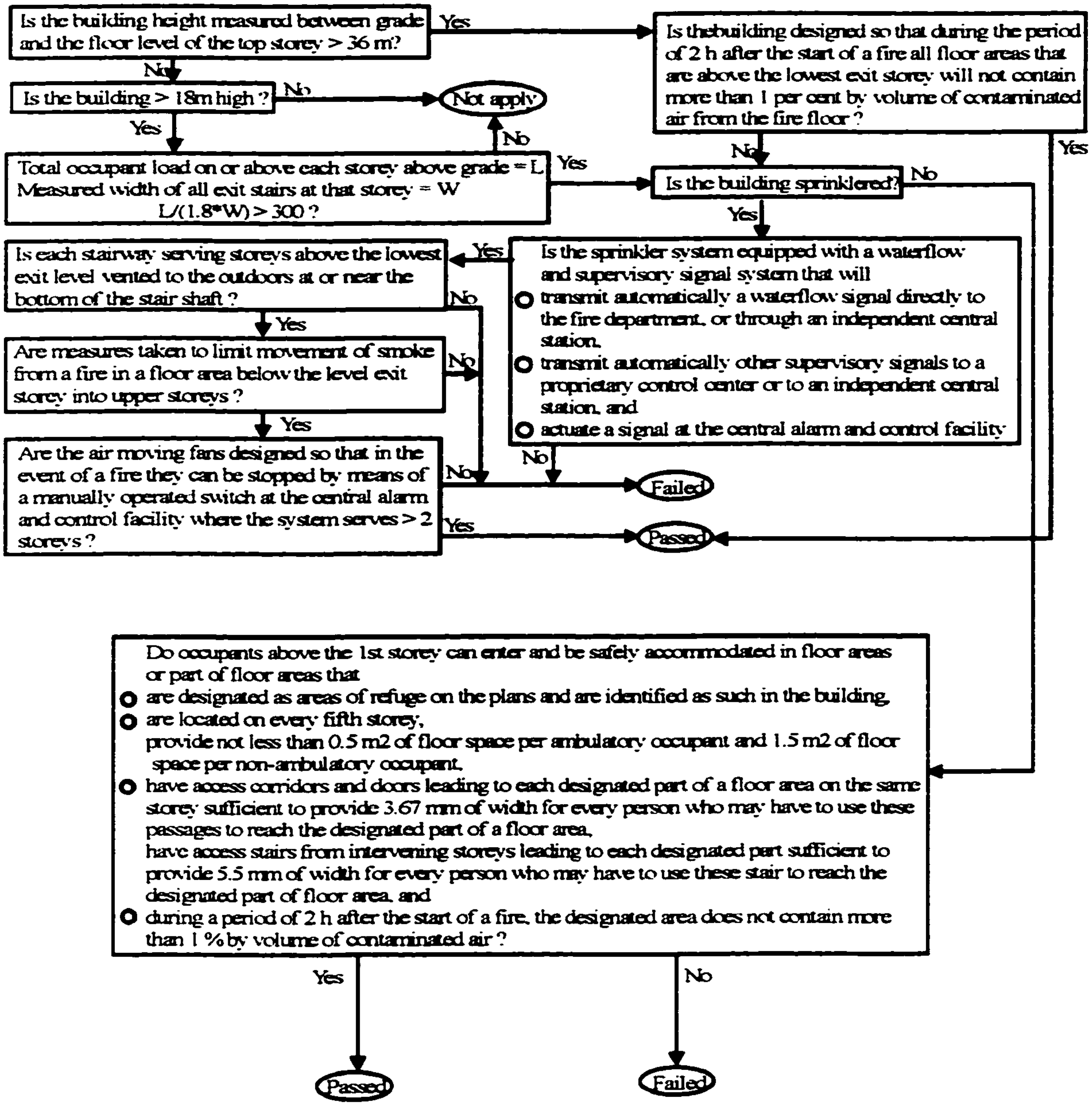


SEPARATION AT BOTTOM OF VERTICAL SERVICE SPACE (Sentence 3.5.3.1.(4))

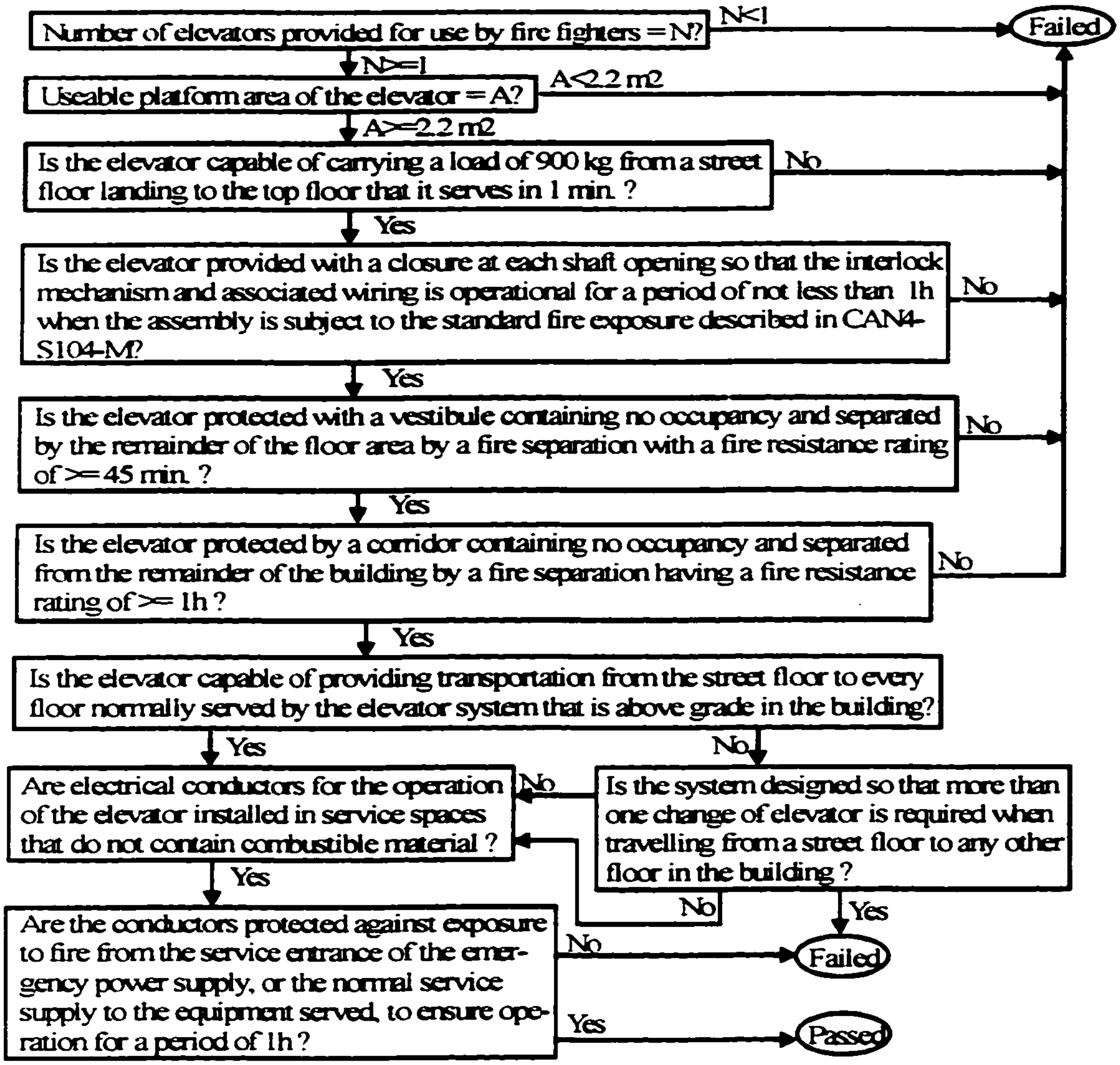


12/ HIGH BUILDING REQUIREMENTS

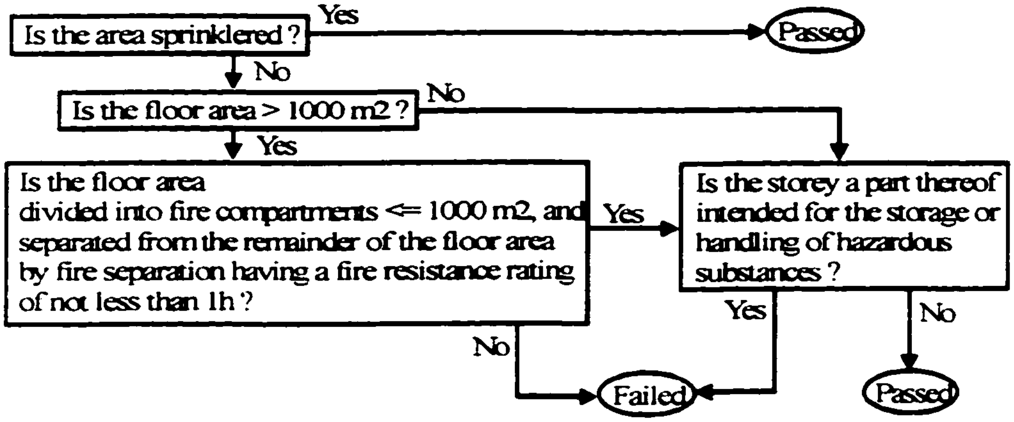
SMOKE CONTROL SYSTEM IN HIGH BUILDINGS
SMOKE LIMITS IN UPPER STOREYS
SMOKE LIMITS IN EXIT STAIRS
AREAS OF REFUGES (Articles 3.2.6.1 - 3.2.6.3)



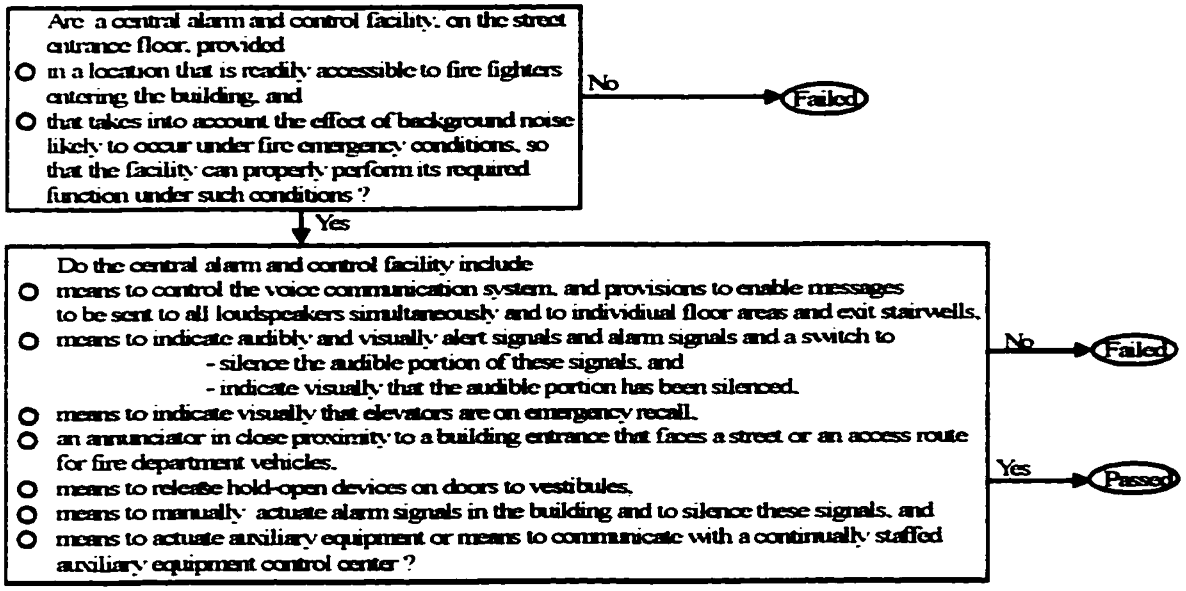
ELEVATOR FOR USE BY FIRE FIGHTERS (Article 3.2.6.9)



MANDATORY SPRINKLERING OF DEFINED SPACES (Article 3.2.6.11)



CENTRAL ALARM AND CONTROL FACILITY REQUIREMENTS (Article 3.2.6.12)

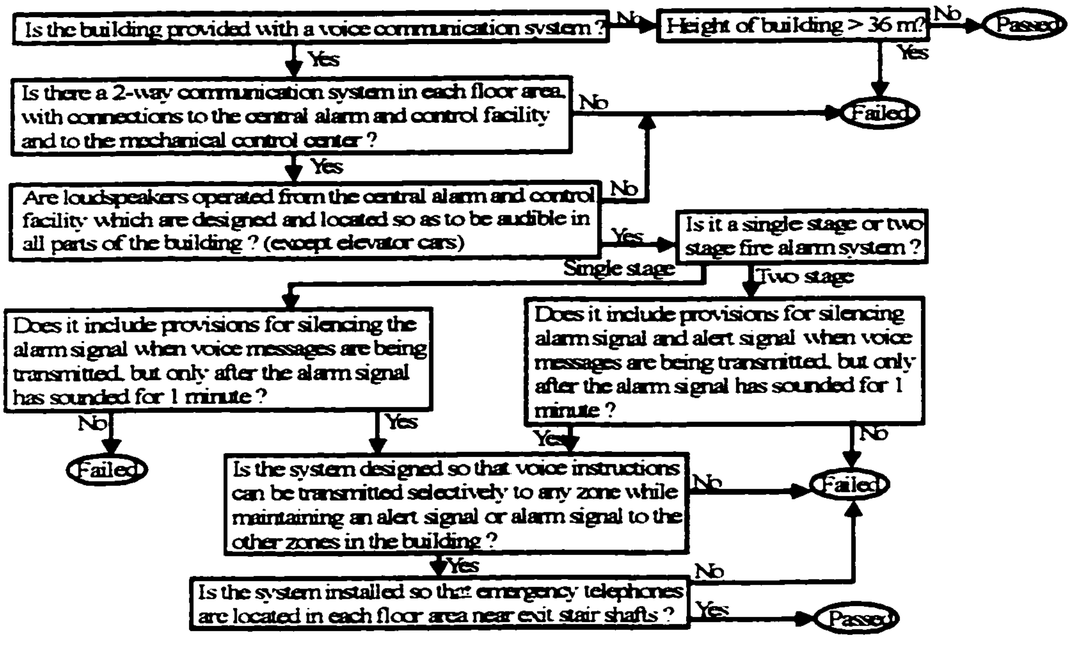


Are a central alarm and control facility, on the street entrance floor, provided
○ in a location that is readily accessible to fire fighters entering the building, and
○ that takes into account the effect of background noise likely to occur under fire emergency conditions, so that the facility can properly perform its required function under such conditions ?

Do the central alarm and control facility include
○ means to control the voice communication system, and provisions to enable messages to be sent to all loudspeakers simultaneously and to individiual floor areas and exit stairwells,
○ means to indicate audibly and visually alert signals and alarm signals and a switch to
- silence the audible portion of these signals, and
- indicate visually that the audible portion has been silenced.

○ means to indicate visually that elevators are on emergency recall.
○ an annunciator in close proximity to a building entrance that faces a street or an access route for fire department vehicles.
○ means to release hold-open devices on doors to vestibules.
○ means to manually actuate alarm signals in the building and to silence these signals, and
○ means to actuate auxiliary equipment or means to communicate with a continually staffed auxiliary equipment control center ?

VOICE COMMUNICATION SYSTEMS (Articles 3.2.6.13 - 3.2.4.22)



Is the building provided with a voice communication system ?

Height of building > 36 m?

Is there a 2-way communication system in each floor area, with connections to the central alarm and control facility and to the mechanical control center ?

Are loudspeakers operated from the central alarm and control facility which are designed and located so as to be audible in all parts of the building ? (except elevator cars)

Is it a single stage or two stage fire alarm system ?

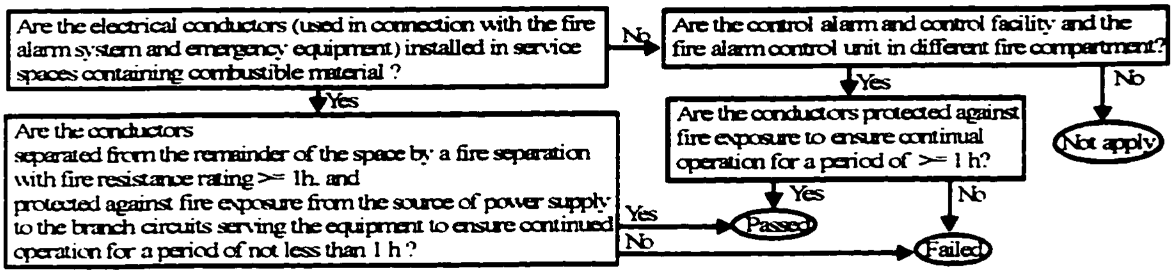
Does it include provisions for silencing the alarm signal when voice messages are being transmitted, but only after the alarm signal has sounded for 1 minute ?

Does it include provisions for silencing alarm signal and alert signal when voice messages are being transmitted, but only after the alarm signal has sounded for 1 minute ?

Is the system designed so that voice instructions can be transmitted selectively to any zone while maintaining an alert signal or alarm signal to the other zones in the building ?

Is the system installed so that emergency telephones are located in each floor area near exit stair shafts ?

PROTECTION OF ELECTRICAL CONDUCTORS (Article 3.2.6.14)



Are the electrical conductors (used in connection with the fire alarm system and emergency equipment) installed in service spaces containing combustible material ?

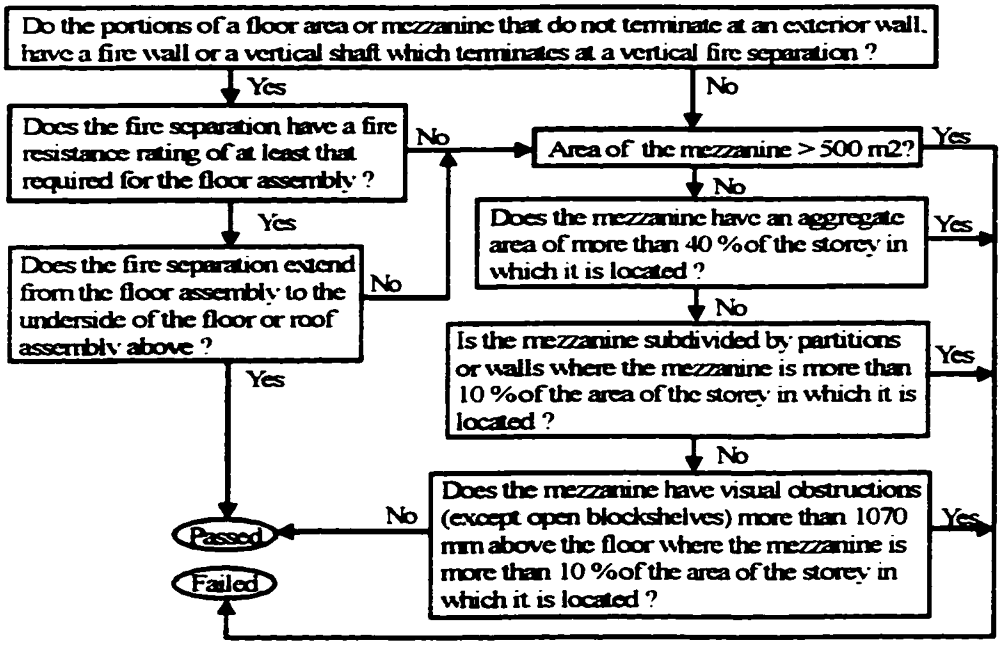
Are the control alarm and control facility and the fire alarm control unit in different fire compartment?

Are the conductors
separated from the remainder of the space by a fire separation with fire resistance rating >= 1h, and
protected against fire exposure from the source of power supply to the branch circuits serving the equipment to ensure continued operation for a period of not less than 1 h ?

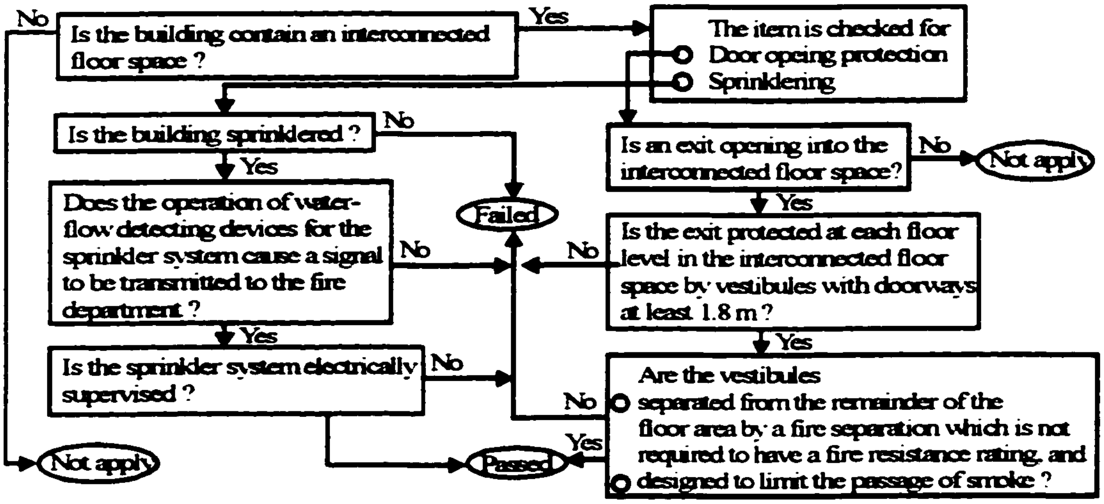
Are the conductors protected against fire exposure to ensure continual operation for a period of >= 1 h?

13/ MISCELLANEOUS REQUIREMENTS

MEZZANINE EGRESS REQUIREMENTS (Article 3.2.8.1)



DOOR OPENING PROTECTION (Article 3.2.8.4)
INTERCONNECTED FLOOR SPACE REQUIRING BUILDING TO BE SPRINKLERED (Article 3.2.8.6)



UNDERGROUND WALKWAY NONCOMBUSTIBLE FINISHES (Article 3.1.13.9)
SMOKE BARRIER DOORS IN UNDERGROUND WALKWAYS (Article 3.2.3.21)