Development of an Expert System for the Fire Protection Requirements of the National Building Code of Canada

M. J. Frye, D. M. Olynick, and R. B. Pinkney[1]

Abstract

In Canada, the standard for fire safety for new buildings, reconstruction of buildings including alterations and additions, and buildings involving a change in occupancy, is established in Part 3, Use and Occupancy, of the National Building Code of Canada. While the fire protection requirements contained in this section of the Code are very explicit, inexperienced or infrequent users of the Code often find it confusing and overwhelming because of the number of requirements which apply or seem to apply to a given building. An experienced code user or expert understands what information is relevant and will generally use a systematic process to determine the fire protection requirements that are applicable.

Because the human approach to fire protection analysis is, in fact, systematic and logically sequential, and because the knowledge contained in codes and standards is largely in the form of rules, an expert system can be developed to effectively simulate human competence in fire protective design.

This paper describes the development of a user-friendly expert system that closely mimics the human approach used in the fire protection analysis of those buildings regulated by Part 3, Use and Occupancy of the National Building Code of Canada. The principal fire protection requirements of the Code have been incorporated into the expert system. The resulting expert system will be useful to the experienced code user as a code assistant, and to the inexperienced or infrequent code user who requires code information when no expert is available.

Introduction

Construction is one of the largest industries in Canada [Clark 86]. It employs more than 650,000 Canadians and has produced a capital stock of more than $1,100 billion. The efficiency, effectiveness and cost of construction is clearly a vital concern to all Canadians.

The standards and codes that regulate building construction strongly influence the industry. The ability of user groups to understand and correctly apply these regulations to building design has considerable effect on the success or lack of success of the final product. An increasingly sophisticated public is less willing to accept the latter.

In Canada, the minimum standard for fire protective design is established in Part 3, Use and Occupancy, of the National Building Code of Canada(NBC) (NBC 1990). User groups of this part of the Code include designers, building contractors, regulatory authorities, and building owners and property managers.

While the fire protection requirements contained in this section of the Code are very explicit, inexperienced or infrequent users often find the Code confusing and overwhelming because of the number of requirements that apply or seem to apply. The

[1]Department of Civil Engineering, University of Manitoba, Winnipeg, Canada

frustration in trying to extract specific provisions from the general document leads many users to remark that building requirements are too complex and ambiguous. The Code must, however, take into account a wide variety of buildings and building products. As a legal document it must be written so that it can be enforced under law [Hewitt 86].

Over the last decade, considerable time, effort and money have been spent in an effort to make the Code more understandable and easier to use. Some initiatives have focused on training through formal education programs, seminars, and workshops. Others have involved the development of Code commentaries, illustrated codes, and definitive technical information. Despite these efforts, expertise in Part 3 has been slow to develop. Code knowledge and performance exhibited by the different Code user groups still vary widely both between groups and within a group.

The knowledge gap that still exists between and within the different user groups contributes to inappropriate use of the regulations, subjective code interpretations, confusion, litigation and increased construction costs. It is therefore essential that all building code practitioners develop comparable expertise in the effective and efficient use of the Code.

More recently, attention has turned to the computer in the belief that it is the single most important key to improved performance. While initial emphasis has been on retrieval systems [PTCBS 90], artificial intelligence (AI) research has exposed the potential of expert systems, the impact of which is only just beginning.

This paper discusses the work that has taken place at the University of Manitoba and the City of Winnipeg in the development of an expert system for the fire protection requirements contained within Part 3 of the NBC.

Development of the Part 3 Expert System

Expert Systems

An expert system is a computer program that represents and reasons with knowledge of some specialist subject with a view to solving problems or giving advice [Jackson 90].

The program may completely fulfil the function of a human expert, or it may play the role of an assistant to a human decision maker. The decision maker may be an expert in his or her own right, in which case the program may justify its existence by improving the productivity of the decision maker. Alternatively, the expert system may provide the technical assistance necessary to raise the level of performance of a novice decision maker to the level of an expert. This improvement in performance was discussed by Vadas (1992), and is illustrated in Fig.(1).

Expert System Shells

Expert systems have two basic components: the *knowledge base* and the *inference engine.* The knowledge base is a collection of facts and decision rules used in the program. The inference engine is the control mechanism that performs the reasoning necessary to solve a problem or provide advice. In addition to the knowledge base and inference engine, an expert system contains a *user interface*, an *explanation facility* and a *knowledge acquisition facility.*

The fact that the inference engine and the reasoning mechanisms incorporated into the inference engine are separate from the knowledge base permits the development of *expert system shells*. An expert system shell contains all the basic components needed to support an expert system other than domain-specific knowledge. The shell is an expert system with an empty knowledge base. The task of the expert system developer is to translate expert knowledge for the problem domain into rules that conform to the logical and syntactical requirements of the shell Heikkila 92].

Shells vary with respect to knowledge base capacity, reasoning-mechanisms used, ability to link to external files, graphics capabilities, explanation facilities, methods for dealing with uncertainty, and price. The selection of the shell should be based on the problem to be solved, the abilities of the developer, and the needs of the user.

Personal Consultant Plus (PCPlus)

The shell used in the development of this expert system was Personal Consultant Plus (PCPlus) [PCPlus 87]. PCPlus was created by Texas Instruments to facilitate the development of expert system programs that can be run on personal computers. Some of the most important features of PCPlus are the scheme for knowledge representation, the reasoning mechanisms, and the user interface.

PCPlus uses the three most common forms of knowledge representation: *parameters*, *rules*, and *frames*. A *parameter* is a structure that identifies or contains a piece of information that the expert system uses to arrive at a solution to a problem. *Rules* express the relationships between parameters and the conclusions about them. A *frame* is a structure that provides for the grouping of parametric data and rules.

Frames can be linked to one another to form a *decision tree* allowing a large problem to be represented by a series of subproblems. The *inheritance facility* allows for information to be used repeatedly in different program frames. Inheritance allows any frame to access parametric information in frames above it in the decision tree and to access rules in frames below it in the decision tree.

In the development of this expert system, the global problem of determining what fire protection requirements apply to a given building design can be broken down into a number of subproblems. Each subproblem can be represented by one or more subframes. The objective of each subframe is to solve an individual subproblem. In the case of Part 3 of the NBC this facilitates the development of the knowledge base on an incremental basis. This will also be important when it is necessary to update the knowledge base as a result of changes to the Code.

Forward chaining and *backward chaining* are the reasoning mechanisms common to most rule-based systems. Both of these mechanisms are available in PCPlus. Backward chaining is goal oriented and is therefore more suited to code analysis where the objective is to determine the applicability of a particular code requirement to a given building design. However, forward chaining can be combined with backward chaining to reach decisions more efficiently. The program developed here makes use of both reasoning techniques.

The user interface of PCPlus accommodates both natural language modules and graphic presentation modules. The shell comes with a versatile graphics adapter. Drawings can be imported from almost any other computer graphics program. Drawings generated by Harvard Graphics [Harvard Graphics 91] and AutoCAD [AutoCAD 88] are used extensively in this program. The user interface is likely to have the greatest influence on user acceptance of the expert system. This is discussed in more detail in a later section of this paper.

Program Development

The minimum design requirements for fire protection are found in Part 3, Use and Occupancy of the NBC. These requirements are the principal resource for building the knowledge base. Part 3 provides a tangible focus for transcribing facts and rules into the knowledge base [Heikkila 92].

However, human expertise is essential to the successful development of this expert system for a number of reasons. A code expert is familiar with the intent and organization of the Code. The expert knows how the Code requirements are to be interpreted and when one set of requirements may govern over another. In some cases, requirements contained in another code or standard may be applicable. Without the additional resource of code experts the development of this expert system would have been difficult, if not impossible.

Experienced building department plan checkers are the most knowledgeable user group. These Plan Examiners use a structured process when they conduct a Part 3 analysis of a building design. A Part 3 manual checklist [Winnipeg 90] reflects this process. The process allows the Plan Examiner to quickly eliminate those Code requirements that are not applicable and focus on the requirements that are. The primary construction requirements are determined before dealing with more specific issues. Plan Examiners can be confident in making some conclusions about Code compliance even at an early stage of the review. The input of the City of Winnipeg Plan Examiners and the manual checklist used by them had considerable influence in the development of this expert system.

Part 3 of the NBC is, in itself, very large and complex, containing many sections and subsections that are somewhat independent of each other. This facilitated an incremental approach to the development of the knowledge base. The idea behind this approach was to build up the knowledge base by working with manageable knowledgeable blocks.

The problem domain was initially narrowed to Section 3.2., Size and Occupancy Requirements for Fire Safety [Olynick.89]. This section of the NBC provides the structural fire protection requirements for buildings and is generally regarded as the starting point in the determination of the fire protection requirements applicable to a given building. The expert system was then expanded incrementally as other sections and subsections were added to the knowledge base. The current status of the knowledge base is reflected in the program main menu and submenu choices, which are illustrated in part in Fig.(2).

Each independent fire protection topic was represented by a separate frame in the knowledge base. Subproblems within a topic were represented by subframes where the efficiency of the expert system could be improved.

The information necessary to reach a conclusion and the conclusion reached were represented by a decision tree for each frame. A typical decision tree is illustrated in Fig.(3). Each path in the decision tree represents a rule in the knowledge base. After all possible paths through the decision tree have been transcribed into the knowledge base, the knowledge base is complete and the expert system can reach a conclusion with absolute certainty.

The development of a decision tree is sometimes time-consuming; however, it has several advantages: this format is easily understood by code experts and an error in logic can be detected quickly; it is immediately evident if the Code is silent on a particular requirement or if the Code contradicts itself; modification of the knowledge base is simplified if the Code changes; programming is relatively simple.

The order in which information is requested in the decision tree should reflect the order in which a code expert would request information. Experts generally prefer to ask the minimum number of questions necessary to reach a conclusion provided the conclusion can be reached with absolute certainty. It is desirable that the expert system ask questions in the same order that they would be posed by an expert. This will not necessarily happen if the expert system is allowed to reason independently to a conclusion. Although the conclusion will be the same, confidence in the system may be eroded if the sequence of questions is substantially different from that which is expected. Thus some degree of procedural programming is necessary; that is, the programmer should sometimes override the automatic sequences generated by the shell in favour of the sequences used in the corresponding manual procedure.

Testing of the expert system serves two purposes. The first and most obvious is to determine whether the expert system is reaching the correct conclusion. This type of testing takes place continuously throughout the development of a program. It should be done by the programmer working directly from a decision tree that has been verified by an expert, or by a code expert working directly with the program. During this phase of testing the logical sequence of user prompts is also checked.

The suitability and effectiveness of the user interface must also be tested. Needless to say, effective communication between the expert system and the end user is crucial to the success of the program. A number of choices are available for the interface method. The system should be tested by both novices and experts. This testing takes more time and it may be necessary to strike a balance between the preferences of the two groups. The user interface is discussed in more detail in the next section of this paper.

User Interface

The ease with which a user can use an expert system program, respond to queries, and understand the conclusions reached at the end of a consultation session depends on the user interface. The user interface consists of such items as menu generators, natural language modules and graphic presentation modules.

A number of options usually exist for the user interface method. The input method may be a simple query mode where the expert system elicits information by asking the user to respond to a direct question. Alternatively, the interface may be graphics oriented where the user is requested to enter information at specific locations on a graphics screen. Other options include data input by means of a spreadsheet or database file. The developer must keep in mind that the format for eliciting information and presenting conclusions must be attractive and easily understood by the expert system user. Since many potential users are wary and distrustful of computers to begin with, the favour that the expert system achieves will often depend on the user interface.

In the development of an expert system for a building code, it is necessary to be sensitive to the code experience of the user. Experienced code users will become annoyed if the system is overly helpful or if they perceive that too many trivial questions are asked before a conclusion is reached. They generally prefer a system designed on the assumption that the user will understand and respond correctly to a system prompt. While this approach is perhaps desirable for the knowledgeable code practitioner, it may be inappropriate for an inexperienced code user. The novice must often be asked several questions to ensure that correct information is provided. A graphics-oriented interface may be more appropriate in this case. Similar guidelines apply to the output of conclusions.

In this expert system, input information is generally requested by a simple query mode rather than by a graphical interface. However, the user is usually provided with a **<HELP>** option if assistance in responding to a prompt is required. Most of the help screens are combinations of text modules and graphic displays. A typical **<HELP>** screen is illustrated in Fig.(4). Conclusion screens are either a text module, graphic module or combination. This is illustrated in Fig.(5).

Conclusions

The expert system is presently undergoing in-house testing by Part 3 Code experts as well as by a few members of the design community in Winnipeg. Present attention focuses on improving the user interface. It is expected that the program will be introduced into the design community in the summer of 1992.

The objectives in the development of this expert system have been to improve the productivity of a code expert by providing a useful assistant and to provide the expert assistance necessary to improve the performance of the novice. Whether these objectives will be accomplished and what impact this program will have on the construction industry remain to be seen; however, initial response has been encouraging.

References

[AutoCAD 88] AutoCAD, Reference Manual for AutoCAD Autodesk Inc. Oakland, California, 1988

[Clark 86] Clark, Ferrers R. S., "Trends and Issues in Construction." *The Construction Industry and the Issues Facing It, An Overview.* Institute for Research in Construction, National Research Council of Canada, pp. 1 - 9., 1986

[Harvard Graphics 91] Harvard Graphics User's Manual. SPC Software Publishing. Mountain View, California., 1991

[Heikkila 92] Heikkila, Eric J., and Edwin J. Blewett., "Using Expert Systems to Check Compliance with Municipal Building Codes". *Journal of the American Planning Association 58*, 1, pp. 72 - 80., 1992

[Hewitt 86] Hewitt, Robert A., *Building Regulation and the National Codes. The Construction Industry and the Issues Facing It, An Overview.* Institute for Research in Construction, National Research Council of Canada, pp. 35 - 46., 1986

[Jackson 90] Jackson, Peter., *Introduction to Expert Systems.* Saint Louis, Missouri, Addison -Wesley, 1990

[NBC 90] NBC. *National Building Code of Canada, 1990.* Associate Committee on the National Building Code. National Research Council of Canada. Ottawa, Canada., 1990

[Olynick 89] Olynick, D. M., Pinkney, R. B., and Frye, M. J., 'A Prototype Expert System for the Fire Protection Requirements of the National Building

Code of Canada.' *Proceedings of the International Conference on Municipal Code Administration Building Safety and the Computer*, Manitoba Building Officials Association. pp. 299 - 319., 1989

[PCPlus 87] PCPlus. *Reference Manual for PCPlus by Texas Instruments.* Austin, Texas., 1987

[PTCBS 90] PTCBS., Provincial/Territorial Committee on Building Standards, Subcommittee on Computerized Code Products. Meeting Agenda., 1990

[Vadas 92] Vadas, Oliver., *Artificial Intelligence Industrial Applications Workshop.* Confederation College, Thunder Bay, Ontario., 1992

[Winnipeg 90] City of Winnipeg. *Part 3 Checklist.* Plan Examination Branch, Building Inspections Division, Department of Environmental Planning. Winnipeg, Canada, 1990

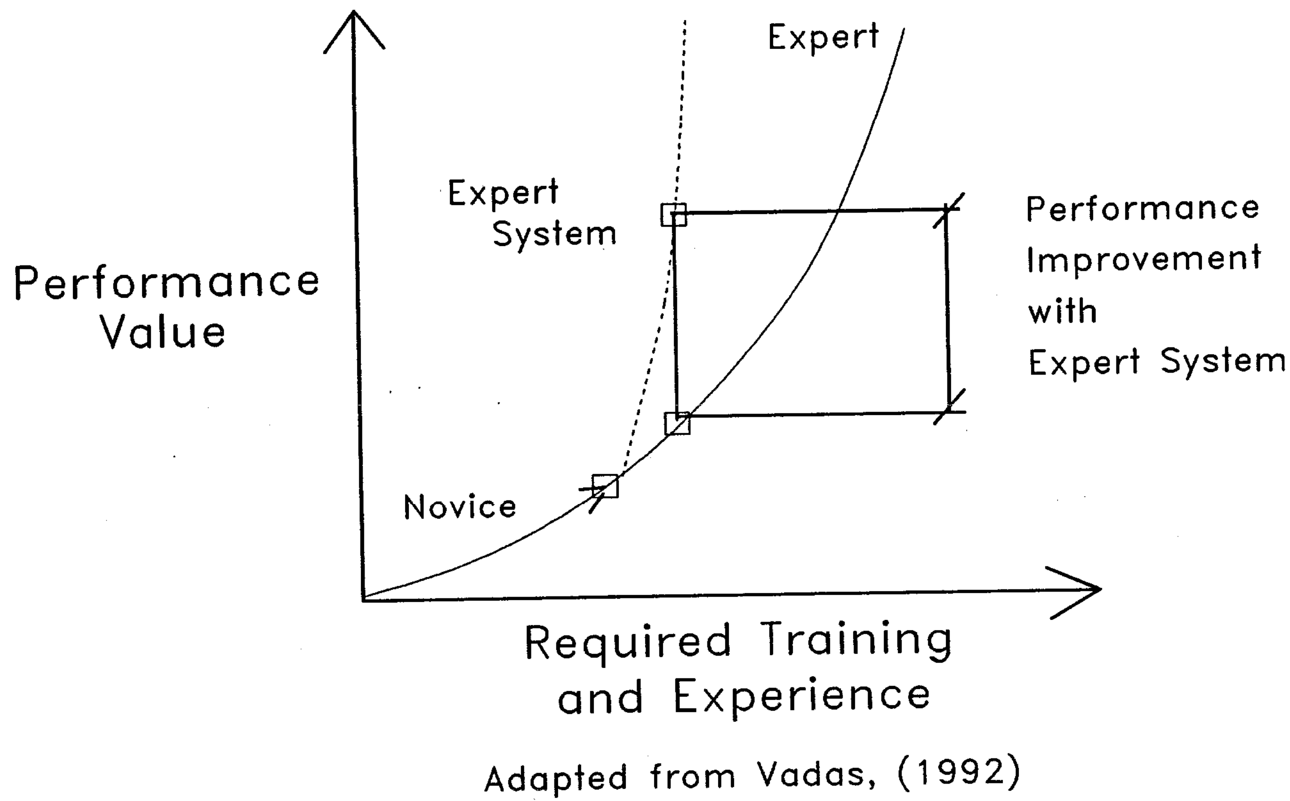


Fig.(1) Performance Improvement With Expert System

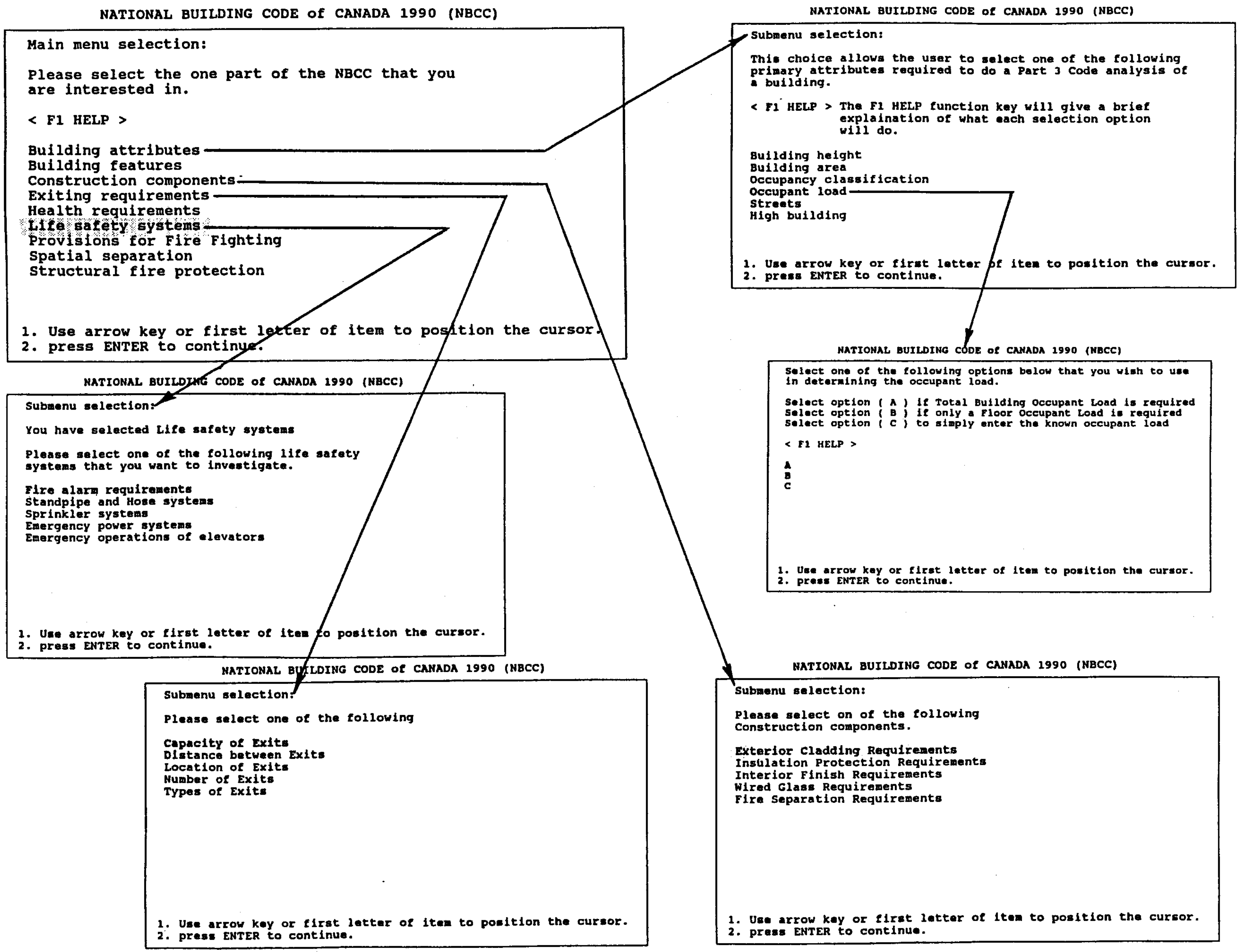


Fig.(2) Part 3 Expert System Main and Submenu Selections

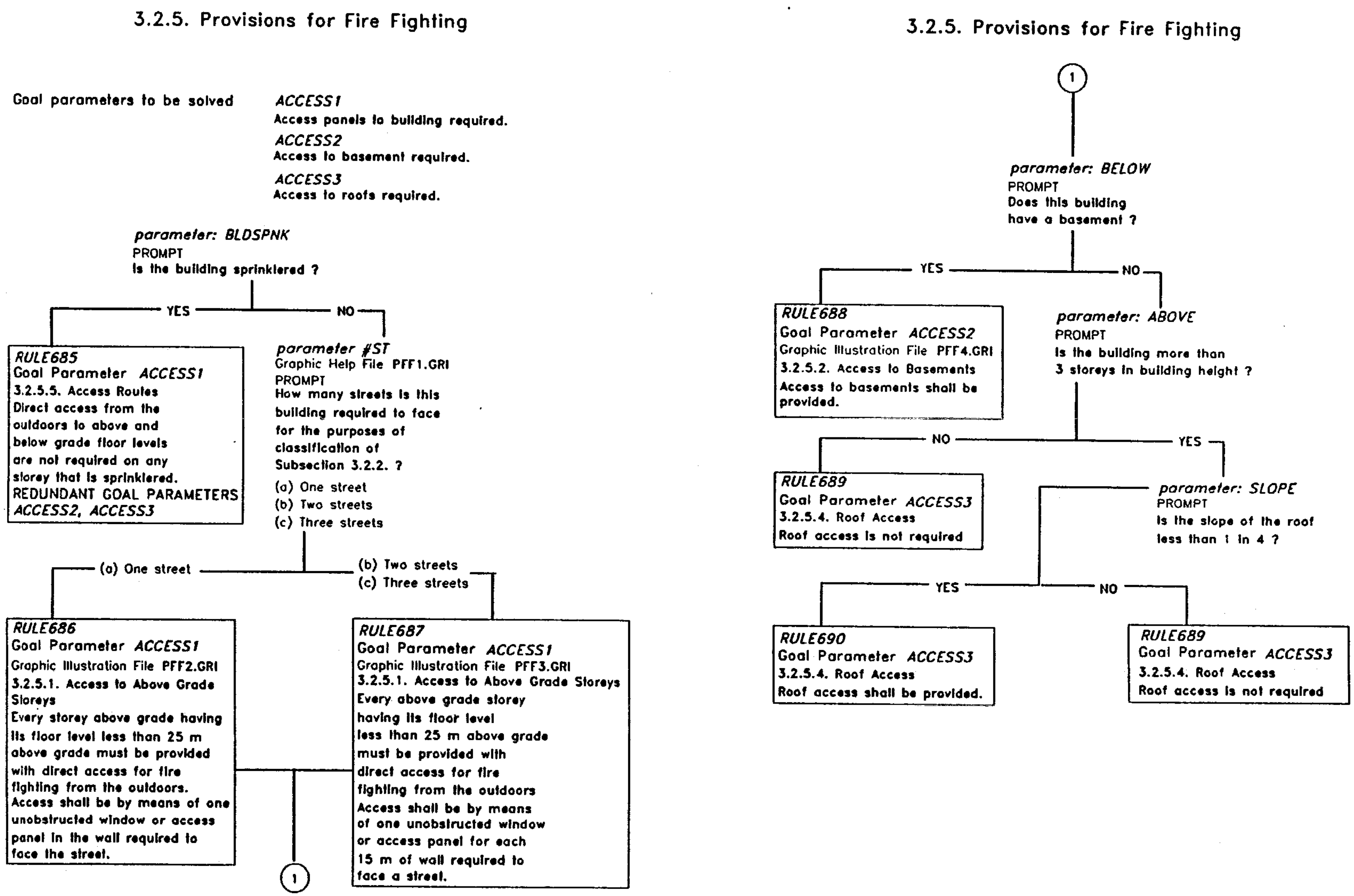


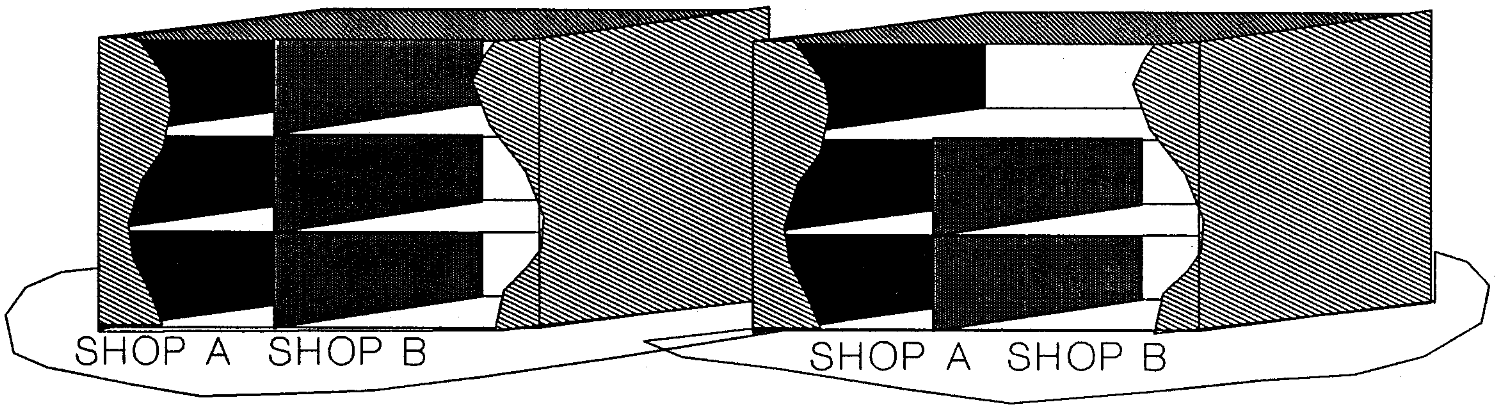
Fig.(3) Decision Tree, Provisions for Fire Fighting Subframe

FIRE ALARM REQUIREMENTS FOR VERTICALLY DIVIDED BUILDING

1 h fire separation extending through all storeys, no access openings through separation.

FIRE ALARM REQUIREMENTS FOR VERTICALLY DIVIDED BUILDING WITH ACCESS THROUGH SEPARATION

1 h fire separation with access opening on 3rd storey



Each shop considered as a separate building for the purpose of determining fire alarm requirements.

Shops A and B must be considered as one building for the purpose of determining fire alarm requirements.

Fig.(4) Help Screen Graphic

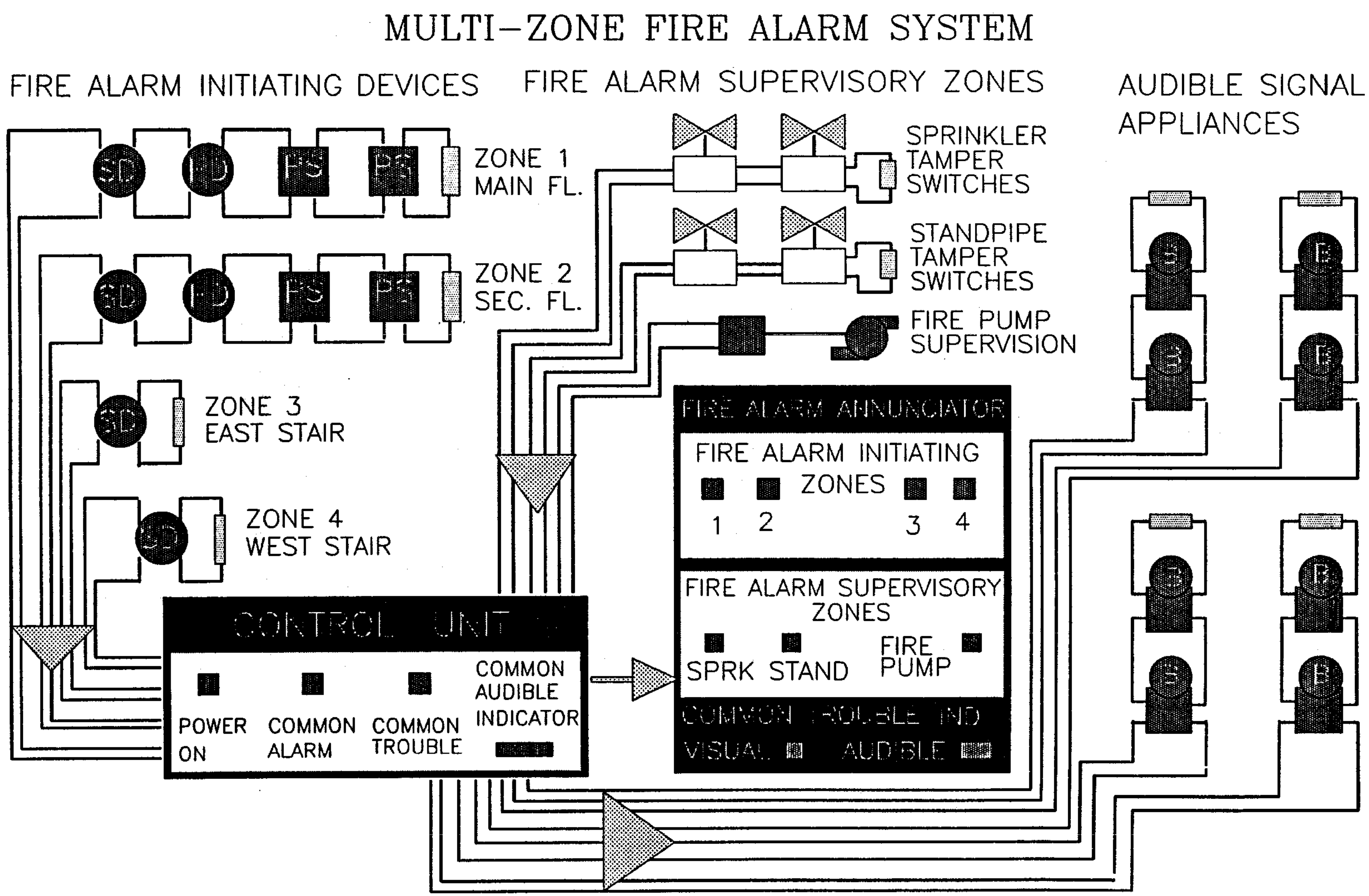


Fig.(5) Conclusion Screen Graphic