SERIES BUI

NO.66
(1987)

BUILDING RESEARCH ASSOCIATION OF NEW ZEALAND

# REPRINT

CI/SfB
| | | (Agm)
UDC 681.3.06

# *Expert systems for regulations and codes*

**J.G. Hosking, W.B. Mugridge and M. Buis**

This report originated in work commissioned by BRANZ as part of its ongoing research into knowledge based systems. The views represented are not necessarily those of the Association.

Reprinted from ***Proceedings of the Second New Zealand Conference on Expert Systems,*** Auckland, 2-4 February 1987.



BRANZ
BUILDING RESEARCH ASSOCIATION OF NEW ZEALAND

# Expert Systems for Regulations and Codes

J. G. Hosking W. B. Mugridge and M. Buis†
Department of Computer Science
University of Auckland

† current affiliation Expert Software Ltd, Newmarket, Auckland

## Abstract

A good application area for expert systems is in the interpretation of regulations and codes of practice. In this paper we examine how an expert system can serve as a design tool to help ensure that code requirements are met, using the *FireCode* system, based on *DZ4226 Code of Practice for Design for Fire Safety*, for illustration. Not all codes are suitable for encoding as an expert system. The attributes of DZ4226 which makes it suitable for such an approach are discussed.

## Introduction

Designers in the building and construction industry must ensure that their designs comply with a wide range of standards and codes of practice, covering aspects such as requirements for fire safety, seismic and wind loading, and electrical supply. Many of these codes are complicated, requiring considerable effort on the part of the designer to ensure that the requirements are met in a design. Hence, it is desirable that design tools be developed to assist in this process, freeing the designer to consider the more creative aspects of the design process.

One method of developing code-based design tools, which is receiving increasing attention, is to use an expert-systems approach. For example, Sharpe et al (1986) describe two expert systems, one of which assists designers to "navigate a correct path" through the Australian Wind Loading Code and the other assists designers in interpreting the SAA Timber Structures Code. Rosenman et al (1986) describe an expert system based loosely on the Australian Uniform Building Code.

In this paper we look at a design tool for a fire safety code, *FireCode*, developed as a joint project between the authors and the Building Research Association of New Zealand (BRANZ). One of the aims of this project was to assess the suitability of an expert-systems approach to code-based design tools (Whitney and Dechapunya, 1985). The development of this tool has clarified the characteristics a code should possess to make it suitable for implementation as an expert system. We discuss these characteristics and examine some of the issues that arose during the development of *FireCode*.

## DZ4226

*DZ4226 Code of Practice for Design for Fire Safety* (Hay, 1984) is a draft New Zealand standard which lays out building design requirements to ensure public safety in the event of a fire. The code is reasonably complex because one of its principal aims is to allow designers flexibility in meeting its requirements. In the existing standard (SANZ, 1983) constraints are applied to the building as a whole without considering that different functions within the building may have different safety requirements. Flexibility is achieved in DZ4226 by constraining the design of individual building components, such as a set of rooms with a common function, and then considering the interaction of those components. Considerable cost savings can be made using this approach as areas of high fire hazard can be "contained", allowing less expensive materials to be used in the non-hazardous areas.

However, checking a building design against the new code is a non-trivial exercise, as many clauses are conditional on the results of applying other clauses. Thus much of the effort of a designer consists of determining those clauses that are relevant to a particular design. This trade off between flexibility, with its associated cost savings, and complexity, with its increased effort, is a common one for code and regulation authors.

| Use Group | Use Class | Criteria for function or purpose of space | Fire severity grading criteria | Life safety hazard grading criteria<br>Occupant response grade | Occupant mobility grade | Access to safety grade |
|---|---|---|---|---|---|---|
| A | | Spaces used for assembly purposes | | | | |
| | AS | enclosed rooms or suites containing not more than 100 occupants, but not less than 10 children or 30 adults | low | good | good | good to limited |
| | AM | enclosed rooms or suites containing not more than 500 occupants | low | good | good | good to poor |
| | AL | enclosed rooms or suites containing more than 500 occupants | low | limited | good | poor |
| | AO | structures providing open-air seating or standing spaces, with or without shelter | extra low | limited | good | limited to poor |
| S | | Spaces used for sleeping | | | | |
| | SC | enclosed rooms or suites used by occupants under institutional care because of mental or physical disability or illness | low | poor | poor | limited |

**Table 1: Part of Scedule 2.3A: Schedule of use groups and classes, with criteria for functions and fire hazard criteria (from Hay, 1984).**

2.4.3.1 Grading: The mobility which occupants are presumed to have in moving to escape from fire is based on the following assessments:

(a) Good: where occupants are fit, able, and free to move to exitways without assistance

(b) Limited, where:
(i) more than 5% of occupants are physically disabled but able to move of their own volition;
or (ii) more than 30% of occupants are between 2 and 3 years old.

(c) Poor, where:
(i) any occupant is involuntarily restrained from moving;
or (ii) more than 30% of occupants are bedridden;
or (iii) more than 30% of occupants are less than 2 years old.

**Fig. 1: Clause 2.4.3.1. from DZ4226 (Hay, 1984)**

*Part 6, Means of Escape*, is especially complex and, together with *Part 2, Use Classifications*, forms a relatively self contained unit of the code. Part 6 is concerned with ensuring occupants are able to react quickly to a fire and that adequate escape paths are provided for them. For example, the length and width of escape paths from a space are prescribed, based on attributes such as the density and physical ability of occupants, and the inflammability of materials stored in the space. Many of these attributes are summarised by the *use classification* given to each *space* in the building. The various *use classifications* are specified in Part 2 of the code; some of these classifications are shown in Table 1.

Some examples of clauses from Parts 2 and 6 are shown in Figs 1 - 3. The commentary sections and diagrammatic explanation are particularly useful when interpreting the requirements in the formal clauses.

## FireCode: an example

The *FireCode* system is a tool to assist building designers to check their plans against the requirements of Parts 2 and 6 of DZ4226. A dialogue illustrating the use of the *FireCode* system is shown in Appendix 1. The designer is checking an auditorium consisting of one main hall and two part storeys.

Attention initially focuses on the main hall. The *use classification* of this space is not known, so the designer is asked a series of questions about attributes of the space relevant to the classification. At this stage the designer is informed of assumptions made about the attributes of the space, such as the *fire severity*. If these assumptions are incorrect or doubtful, the designer must reconsider these attributes.

Once the *use classification* of the main hall is known, *FireCode* asks the designer about other important attributes, such as whether the space is a dead end. Proceeding in this way the other parts of the auditorium and their interactions are examined and a summary of the session is produced, in which key information provided by the user and determined by the system is displayed. This summary pinpoints problem areas in the design and may also suggest possible solutions to these problems. For example, the length of escape paths from the auditorium is found to be longer than allowed by DZ4226.

## FireCode as an expert design tool

For a design tool to be useful the designer must be able to save time during design by using the tool and/or make cost and quality improvements by developing a better model of the design than would be possible manually. Having looked briefly at how the *FireCode* system is used, we now consider what facilities a design tool for this task should provide to be acceptable to a designer. Some of these facilities can best be provided using an expert systems approach, while others are relevant to a design tool regardless of the methodology used and hence must be incorporated into an expert system solution.

### *Limiting detail*

An important principle in the design of any interactive tool is to minimise the amount of interaction required by the user to obtain the desired results. The major task of *FireCode* is to assist the designer to navigate through the code. *FireCode* assists both by determining the clauses in the code that are applicable or likely to be applicable to a particular design, and by helping the designer deal with those clauses in a sensible order. A number of methods are used to minimise the amount of information required from the user to perform these tasks.

The requirements specified by some of the clauses in the code are completely checked by the system. However, other clauses are left to the user to check because their requirements are very easy for a designer to consider but would require much spatial information to be tediously entered for *FireCode* to perform the check. Another major reason for this approach is that in the early phases of design some details may not have been decided upon. In this case it is better for the designer to know the requirement rather than be forced into an early, and possibly undesirable, 6.5.3.3 Multiple exitways: In an auditorium in use class AL at least half ot the **exitways** from any **part-storey**, other than the lowest **part-storey**, shall be in the form of **safe paths**, directly serving the **part-storey**, and may not be combined with any other **safe path** before reaching the **final exit**.

C6.5.3.3. See D 6.5.3.3. An auditorium may have a balcony in the form of a single mezzanine, restricted in size as required for mezzanines and with no more restrictions on the paths of travel than apply to mezzanines. However if the balcony (or circle or gallery) exceeds the mezzanine area, or if there is more than one tier of seating, then these upper tiers all become classed part-storeys and the provisions in Part 3, Part 5, and this Part apply. (see 6.5.2) But, because of the particular problem of crowd panic, the requirement of this clause (and the one that follows) ensures that the audience on a second or higher tier of seating can escape directly to a final exit without being caught up in a jam of people using exitways from lower levels.

D 6.5.3.3

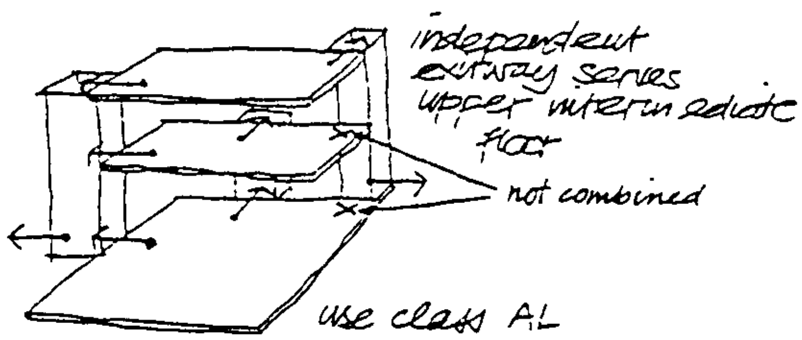


**Fig. 2: Clause 6.5.3.3. from DZ4226 (Hay, 1984). Note the commentary and explanatory diagram**

6.3.1.1 General method: Except where reduced for fixed seating, the widths of **paths of travel** in each **storey** shall be determined by:

(a) calculating the **occupant load** of each **storey** as specified in 6.3.2;
then (b) determining the minimum widths as specified in 6.3.3;
then (c) applying the minimum widths to determine the total capacity of **exit-ways** serving that **storey** as specified in 6.5.1;
and (d) adding the minimum widths of any exitways combined under the conditions specified in 6.5.5;
but (e) observing the special requirements specified in 6.3.3; 6.4.1; 6.4.7; 6.5.1 to 6.5.4.

**Fig. 3: Clause 6.3.1.1. from DZ4226 (Hay, 1984). Parts (a) - (d) illustrate procedural knowledge encoded in the clause. Part (e) indicates constraints that must be taken into account when applying the procedural knowledge.**

design decision.

The dialogue illustrates the above method when *FireCode* is dealing with Clause 6.5.3.3, as shown in Fig. 2. This clause specifies a complicated requirement relating to the interconnection of exitways from part storeys in auditoria. *FireCode* tests if the clause is *likely* to be relevant by using easily determined information such as the *use classification* and the space being a part storey. In this case the clause is found to be potentially relevant to the *circle* part storey so the designer is informed of the requirements of the clause by listing them in the summary for *circle*.

Two other methods are used to limit the amount of information needed in an average session. *FireCode* makes some default assumptions about the design being considered, the hazard ratings in *use classification* are an example. Such assumptions are explicitly stated, and the user is informed how to alter them. Also, features which can be considered relatively independently and may not be of interest in the majority of cases, such as seating requirements, are made optional.

*Allowance for iterative design*

*FireCode* permits the designer to iterate to a solution, allowing potential designs to be created, evaluated and modified. The designer is easily able to make changes at any point. Problems encountered in the design are isolated and explained with possible solutions given. This is illustrated in the dialogue in Appendix 1; the designer is informed of a problem with the length of escape paths from the auditorium, and makes a change to the design by altering the answer to an earlier question.

The ability to make changes means that the designer is encouraged to consider various possible designs and assess their correctness using the tool. The tool frees the designer from the drudgery of detailed assessment of a possible design and allows concentration on the more creative aspects. Any useful design tool must provide the opportunity to pose "what if?" questions in the development of the design.

To be truly useful, the tool must be able to retain as much information as possible after the modification of an answer. In typical menu-based systems backing up several menus to modify a selection usually involves the loss of all subsequent answers. In *FireCode*, only those answers which are inconsistent with a change that has been made are discarded, and the designer questioned again on those points that are relevant to the current design.

On a longer time scale the ability to save details of a session for later resumption is important, particularly when coupled with the ability to modify prior answers. The designer can temporarily exit *FireCode* to consider the affect of changes, such as the addition of sprinklers to solve the auditorium path length problem. Also, as the design is developed and more details become known, requirements that were ignored in the early stages, such as seating, can be considered without reentering the original design information.

*Explanation and justification*

Professionals are loath to accept unjustified advice, particularly if liability issues are involved. Therefore a design tool such as *FireCode* must be able to provide clarification, explanation and justification to the designer. Explanation of reasoning has long been an important component of expert systems methodology (Buchanan and Shortliffe, 1984). *FireCode* has a number of explanation facilities available; for example, the dialogue shows the system clarifying the term *dead end* and explaining the reason for the question about sprinklers.

A useful enhancement to *FireCode* would be the on-line availability of the complete text of DZ4226 accessible from within a *FireCode* session. When dealing with legal material, it is often important to have the complete context of a regulation or statute available to allow interpretation of possible ambiguities (Greenleaf et al, 1986).

*Non textual interaction*

The intended use of *FireCode* is via dial-in connections to the BRANZ computer using a variety of terminals. Hence, only textual interaction with the user is possible. Other modes of interaction are highly desirable to improve explanation and minimise the interface barrier. For example, as shown in Fig. 2, diagrams form an important component of the commentary in DZ4226. In *FireCode* such diagrammatic explanation is only available indirectly by referring the user to the appropriate diagram in the printed code.

Ideally, a tool such as *FireCode* should be integrated into a complete CAD package, allowing non-textual input devices, such as mice or graphics tablet, to be used and to minimise the multiple handling of sets of data when moving from one design or analysis tool to another. This is a fruitful area for further research.

## Knowledge Engineering of FireCode

*FireCode* was developed using *Class Language*, which provides rule-based and procedural constructs in an object-oriented setting in which classification plays an important role. The design of *Class Language*, with particular application to *FireCode*, is discussed in Mugridge et al (1987).

An expert systems approach provides major benefits in the development phase of a design tool such as *FireCode*. These benefits arise from using an expert system language, such as *Class Language*, which is more suited to encoding the sorts of knowledge that arise in the code than traditional programming languages.

Some sections of the code, such as parts (a) through (d) of Clause 6.3.1.1 as shown in Fig. 3 are procedural in nature, in which the steps involved in checking a design are clearly laid out. Many sections of the code are concerned with specifying interacting constraints, such as part (e) of Clause 6.3.1.1, in which order is not relevant. Still other sections of the code are concerned with classifying building components, such as the *use classifications* shown in Table 1.

To use a procedural language like Fortran would unnaturally force these different forms of knowledge into a procedural formalism. The logical structure of the code would be hidden and considerable translation between the written code and the programming language would be required. This would consequently lead to difficulties in understanding and modifying the design tool and in providing adequate explanation.

The acquisition of knowledge is usually considered the most difficult part of expert system development (Waterman, 1985). Using the written code as a basis meant that knowledge acquisition was less of a problem in *FireCode*, particularly because of the thorough job of code development performed by the authors of DZ4226. However expert assistance, provided by Mr Grant Coupland one of the authors of DZ4226, was still required during the development of *FireCode*.

The expert provided the following types of information: clarification of terminology, sensible default information (including the level of expertise that could be assumed of users), clarification of ambiguous and incomplete sections of the code, changes to the wording of clauses that made them more sensible, and ways of interpreting tables. He also played an important role in debugging the knowledge base. Hosking et al (1987) discuss more fully the need for expert assistance.

One result of formalising DZ4226 in *FireCode* was that problems with the written code, such as ambiguous or incomplete requirements,were highlighted. This provided useful feedback on the written code to both the expert and to BRANZ.

## Why DZ4226 was suitable as a basis for an expert design system

The previous sections have shown, by example, that Parts 2 and 6 of DZ4226 are suitable for encoding as a design tool using an expert-systems approach. Not all codes of practice are likely to be so suitable. Hence, it is useful to examine the attributes of the code that made an expert-systems approach successful in the *FireCode* project.

The complex conditional nature of DZ4226 is the principal factor. This complexity is partly due to the number of clauses, but mostly results from the interaction of clauses. As discussed earlier, many clauses specify their requirements completely or partially in terms of meeting the requirements of a number of other clauses. This means that it is not possible to work sequentially through the written code to check a building. Other complexity results from the need to consider groups of components. For example the minimum combined width of all exitways from a storey is dependent on the combined exit requirements of each of the distinct spaces within that storey.

Because of the complexity, manual approaches require a considerable amount of work on the part of the designer. Sharpe et al (1986) have also noted this difficulty with the SAA timber code; even experienced users require a period of re-learning to "navigate" the code after a break from using it. The conditional and non-procedural aspects of the code makes an expert systems approach more suitable than traditional approaches in developing a design tool.

DZ4226 includes considerable commentary which was invaluable in providing explanation in *FireCode*. The commentary was also useful in interpreting the code so reducing the reliance on the expert during the early stages of development of *FireCode*. The authors of DZ4226 have gone to considerable trouble to spell out the rationale behind decisions. To quote the authors (Hay, 1984, page PC-3):

> Therefore , if this code appears more complex, at first sight, than its overseas cousins, it should be realised that it sets out to clearly explain "why"; and that another is that it also sets out to carefully explain the "need", ....

It is tempting to generalize further from our experience with DZ4226 to other codes. However, it is likely that other, as yet unrecognized, characteristics of codes are significant in the development of an expert design tool. We are examining other codes with this in mind.

## Conclusions

The *FireCode* project has provided a number of guidelines for the development of expert design tools based on codes of practice, both with respect to the facilities that should be provided and the nature of the code itself. A design tool must be easy to use and allow the designer to remain in control of the overall design process. This implies that iterative design should be encouraged and adequate explanation of the tool's functioning be provided. An expert system approach is particularly appropriate for the latter. There is a need for such tools to be integrated with other analysis and design tools, particularly more traditional CAD tools which make use of non-textual input and output.

The development of *FireCode* highlighted problems in the written code. The quality of a code of practice would be enhanced by the code and an expert system being developed simultaneously. The formalisation of the evolving code as an expert system focuses on problems such as ambiguity and incompleteness which may not otherwise be addressed. An expert system allows difficult design examples to be tried out, and the code altered if problems are found. A set of case examples can be collected to be run automatically testing any changes to the developing code.

We are planning to develop several new systems based on codes of practice. Possibilities include a system for the design of wall bracing, based on section 6.3 of *NZS 3604*, and a system for checking seismic loading, based on Part C of *DZ4203, General Structural Design and Design Loadings for Buildings*. The development of these systems will enable a better understanding of the suitability of codes as a basis for expert design tools.

## Acknowledgements

The authors gratefully acknowledge the Building Research Association of New Zealand for financial assistance, Grant Coupland for expert assistance, and John Hamer for implementing *Class Language*.

## References


Buchanan B G, Shortliffe E H, 1984, *Rule-Based Expert Systems*, Addison-Wesley, Reading, Massachusetts.

Greenleaf G, Mowbray M, Tyree A, 1986, "Legal expert systems: words, words, words ...?", *Proc. 1st Australian Artificial Intelligence Congress* (Melbourne).

Mugridge W B, Hamer J, Hosking J G, 1987, "Class Language - a language for building expert systems", *Proc. NZES'87*, University of Auckland, Auckland.

Hay R D (editor), 1984, *DZ4226 Code of Practice for Design for Fire Safety* , Standards Association of New Zealand, Wellington , New Zealand.

Hosking J G, Mugridge W B, Buis M, 1987, "FireCode: a case study in the application of expert systems techniques to a design code", *Environment Planning and Design B*, to be published.

Rosenman M A, Coyne R D, Gero J S, "Expert systems for design applications", *The Second Australian Conference on Applications of Expert Systems*, (New South Wales Institute of Technology, Sydney) p 74-91.

SANZ, 1983, *NZS 1900, Chapter 5: 1963 Fire -resisting construction and means of egress*, Standards Association of New Zealand, Wellington , New Zealand.

Sharpe R, Marksjo B S, Thomson J V, Wilson D, 1986, "Expert systems applications in building and construction", *Proc. 1st Australian Artificial Intelligence Congress* (Melbourne).

Waterman D A, 1986, *A Guide to Expert Systems*, Addison-Wesley, Reading, Massachusetts.

Whitney R, Dechapunya A H, 1985, "The building of a system", *New Zealand Interface* (September) 77-82.

BUILDING RESEARCH ASSOCIATION OF NEW ZEALAND INC.
HEAD OFFICE AND LIBRARY, MOONSHINE ROAD, JUDGEFORD.

The Building Research Association of New Zealand is an industry-backed, independent research and testing organisation set up to acquire, apply and distribute knowledge about building which will benefit the industry and through it the community at large.

Postal Address BRANZ, Private Bag, Porirua

branz

ISSN 0111–7459