MiniCode generator: a practical research application for standards processing

MiniCode Generator: A Practical Research Application for Standards Processing

Générateur de minicodes: recherche pratique pour le traitement des normes

Ein "Bauvorschriften-Generator" für die praxisbezogene Normenverarbeitung

Dana J. VANIER
Senior Research Officer
National Research Council
Ottawa, ON, Canada

Dana Vanier has degrees in engineering, building science and architecture. For the past 15 years he has been a researcher at the National Research Council, Canada. He works currently on investigating information technology for the construction industry. His research includes electronic technical information and electronic building codes.

SUMMARY
MiniCode Generation permits users to extract project-specific building codes based on a user-assisted selection of building attributes. A rule-based pre-processor queries the user for details on the construction type, building size, and occupancies, as well as features such as sprinklers, fire alarms, and combustibility rating. The MiniCode Engine then extracts the relevant provisions from the building code in less than 2 seconds on a standard personal computer. The user is then free to browse the resulting MiniCode or modify the attributes and generate a new MiniCode.

RÉSUMÉ
Le générateur de minicodes permet au concepteur de consulter une norme de construction d'après un ensemble de caractéristiques du bâtiment. Un pré-processeur demande à l'utilisateur des précisions concernant le type de construction, les dimensions et l'usage du bâtiment, ainsi que certaines particularités, telles que la présence d'extinction automatique à eau, le système avertisseur d'incendie ou la classe de combustibilité. Le système extrait alors de la norme de construction, en moins de 2 secondes, les dispositions applicables, et cela à l'aide d'un ordinateur personnel conventionnel. L'utilisateur peut dès lors parcourir le minicode ainsi obtenu ou changer les caractéristiques de départ pour en obtenir un autre.

ZUSAMMENFASSUNG
Dem Planer wird die Möglichkeit geboten, projektbezogene Bauvorschriften durch benutzergesteuerte Eingabe von Gebäude-Kenndaten abzurufen. Ein regelgestützter Vorverarbeitungsrechner erfragt vom Benutzer Einzelheiten über Bauart, Abmessungen, Höhe und Nutzungsart des Gebäudes sowie über spezielle Bauelemente wie Sprinkleranlage, Brandmelder und Brennbarkeitsklasse. Das Computersystem wählt dann über einen Standard-PC in weniger als 2 Sekunden die einschlägigen Bestimmungen aus den Bauvorschriften aus. Der Benutzer kann in dem auf diese Weise zusammengestellten Bauvorschriftenauszug beliebig recherchieren oder die Ausgangskenndaten ändern und dadurch einen neuen Vorschriftenauszug erstellen.

1. INTRODUCTION

Many research papers describe projects using expert systems, databases, neural networks and other information technologies to process building standards. The facts indicate that, in spite of the theoretical discussions of the advantages of such systems, there are few applications for building code users [1]. This could be attributed to the difficulty in creating knowledge bases for complex technical and legal documents, to the relative inexperience of building code writing bodies in the area of KBES, to the uniqueness of each and every building code or standard, and to the slow-growing marketplace for Information Technology (IT) tools [1]. The term "building code" in this paper means building code or standard and "electronic code" identifies an IT tool providing access to building codes.

It is the view of the author that research endeavours in this field need not apply the most advanced information technologies, some research can use existing and stable techniques and still create innovative, useful applications. The MiniCode Generator [2, 3] is one research project that can have a significant impact on the building code user community. This paper describes past research activities, the current status of the project and future considerations and possibilities for the MiniCode technique at the Institute for Research in Construction (IRC).

1.1 Past Research

The MiniCode Generator was developed in conjunction with a number of code writing bodies to meet their individual needs [3]. Each partner in the research consortium had expressed an interest to include the MiniCode attributes and features as a search option in their proprietary electronic codes. The first phase of the MiniCode project is now complete and this product has been delivered to the research consortium for testing, over forty copies of the software are in use across Canada.

The first phase concentrated on the needs of building plan examiners, - those municipal or regional authorities responsible for building plan verification prior to the issuance of building permits. The MiniCode Generator identifies significant building attributes or features, such as occupancy of the building or height of the building, it contains attribute tags for each of the building code provisions, and it generates abridged versions of building codes based on the associated attributes.

1.2 Current status

The Plan Examiner MiniCode consists of three discrete software components. (1) The National Building Code of Canada [4] Building Classifier is an expert system preprocessor that assists the users to classify their building; (2) the MiniCode Engine culls the 4000 tagged provisions of the National Building Code of Canada (NBCC) using the Classification Description provided by the user, and (3) the MiniCode Viewer enables the user to browse the resulting document in a hypertext environment, and to record the state compliancy of a building for the resulting provisions. The MiniCode Generator works with Windows™ 3.1 or the Macintosh™ operating system.

The NBCC Building Classifier is a set of 20 classification trees using a rule-based expert system to assist the user classify the building. For example, knowing the building is used for residential occupancy and is less than 4 storeys in building height and is less than 600 square metres in area, the NBCC Building Classifier informs the user that this type of building must conform to the NBCC provisions dealing with small buildings. Some classification trees are obviously more complex, consisting of dozens of interrelated rules. In addition, any number of approximately 50 building features such as noncombustible construction, wood shingles, or sprinklers can be selected and, depending on this selection, the relevant provisions are included or excluded. The output of the NBCC Building Classifier is called the Classification Description.

The MiniCode Engine consists of a bitmap of the attribute tags for provisions of the building code and a parsing engine. The Building Code Bitmap contains the provision number and the applicable attribute for each tagged provision. This static file is searched and compared with the Classification Description provided by the user. Generating of the MiniCode is transparent to the user and take approximately 2 seconds to create a MiniCode on a 50 MHertz, 486 personal computer [2]. The exclusion principle, whereby only irrelevant provisions are excluded from the MiniCode, forms the basis for the search mechanism. The exclusion principle assists the speed of the search and reduces liability for the user and the software developer alike. The MiniCode Engine's output is a sparse description of the classification attributes along with a listing of the provisions for the MiniCode. This file can be saved in ASCII format or can be printed.

The MiniCode Viewer is a WinHELP™ application that reads the file generated by the MiniCode Engine and allows the user to view a listing of the provision numbers and headings for the resulting sparse building code. Double-clicking the appropriate provision number or heading displays the full text of that regulation. Although the MiniCode Viewer can only access provisions that appear in the MiniCode, the hypertext environment of WinHELP™ permits the user to view all linked references and all defined terms, interactively.

2. THE DESIGNER MINICODE GENERATOR

The next anticipated phase of the MiniCode project builds on the past experience and creates a more robust Generator to address the needs of building designers. This extension encompasses a different set of problems and opportunities. For example, the plan examiner already knows if a building is sprinklered: the sprinklers are shown on the drawings. A designer, on the other hand, wants to know if the building must be sprinklered or the advantages of sprinklering when it is not mandatory.

The steps required to develop the Designer MiniCode are included in this paper. These steps should serve as examples for others intending similar MiniCode projects with their building codes.

2.1 Prerequisites

The prerequisites for starting a similar project include an in-depth knowledge of the building code or direct access to a building code expert. In our case, an engineer with 35 years of code writing experience was contracted to perform this function. Encoding the MiniCode Engine also requires someone with "C" programming skills, but this skill set is readily available in both the academic and industrial communities. Access to an expert system shell to encode the classification rules is essential, as is familiarity with WinHELP™. The remainder of the tasks can be handled by a knowledge engineer familiar with building codes and architectural engineering.

The task are broken to four discrete areas. The NBCC Building Classifier contains the rule base and creates the Classification Description. The Document Tagging provides the Building Code Bitmaps. The MiniCode Engine compares the Classification Description to the Building Code Bitmaps and creates the sparse list of the MiniCode provisions. And finally, the MiniCode Viewer allows the user to browse the MiniCode and record the state of compliancy of a building for each provision.

2.2 NBCC Building Classifier

The initial selection of which attributes or features are included for classification is somewhat subjective, and dependant on building code document and the knowledge of the domain expert. It is recommended to start with a small number of classification attributes, twenty is a workable number, and with a limited number of features, approximately 50 are easy to maintain. The numbers can always be increased as and when required.

Generally the building attributes relate to different types of buildings or to a faceted classification for the building type. Faceted classification involves attributes such as major occupancy where the building may be any one of a selection of values such as A1, A2, A3, A4, B1 or B2. Building features, on the other hand, are normally a type of construction or component that is contained in the building. These include items such as elevators or favourable soil conditions. An additional distinction between the two is that building attributes require detailed classification trees to assist the user, whereas features may not. In some instances features can be suppressed if they are not applicable for a type of building. This, in itself, is a simple logic rule: if the building is a residence then do not display potential features such as escalator or electrical vault in that dialog box.

The classification attributes and features for the NBCC MiniCode include the following:

Building Attributes: Major occupancies, subsidiary occupancies, building height, firewall, building area, standpipe and hose, occupant load, number of facing streets, interconnected floor space, high building, sprinklers, noncombustible, fire alarm, barrier-free access, etc.

Building Features: Stucco, wood shingles, house, dwelling units, garage, carport, elevator, escalator, electrical vault, air-supported structure, farm building, etc.

Detailed classification trees must be developed for each building attribute, these are developed in conjunction with the building code expert. Examples of a classification tree for small buildings and the associated rule base are shown in Figs. 1 and 2, respectively.

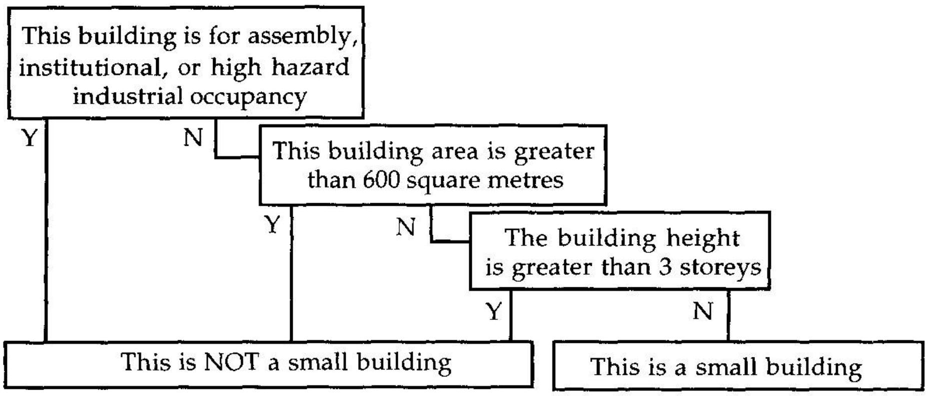

Fig. 1 Classification Tree for Small Buildings

Part9:-p9Occ(All),p9area(All),p9storeys(All).
"Small Buildings (Part 9) must meet occupancy AND area AND storey height requirements"

p9Occ(c):-MO(c).
p9Occ(d):-MO(d).
p9Occ(e):-MO(e).
p9Occ(f2):-MO(f2).
p9Occ(f3):-MO(f3).
"C, D, E, F2, or F3 are valid occupancies"

p9area(600):-setarea(b600).
"600 square metres or less is a valid area"

p9storeys(1):-NumberofStoreys(1).
p9storeys(2):-NumberofStoreys(2).
p9storeys(3):-NumberofStoreys(3).
"1, 2, or 3 are valid storeys heights"

Fig. 2 Rule base for Classification Tree for Small Buildings

In some instances these classification trees should be augmented with additional rules that are above and beyond the scope of the building code. For example, if the building is only one storey in building height there is no need to query the user about whether or not the building is a high building.

Entering the classification trees and the validation of the rules requires considerable attention to detail. Typically one day of work is required for the building code expert to establish classification trees for each building attribute. An additional day is required for each attribute to encode this information and to validate the rule base. In total, this could entail four person-weeks of work.

2.3 Document Tagging

The tagging of the building code document can commence once the building attributes and features have been identified, and the tagging can be done in parallel to the encoding of the rule base.

For the original Plan Examiner MiniCode the tagging was done manually on sheets prepared for the operation, and transcribed to the electronic format. For the Designer MiniCode the existing tagging must be updated to include additional building attributes and features. Typically the information could be entered in the format shown in Fig. 3

Provision Number	Major Occ.	No. Stry.	High Bldg.	Bldg. Area	Part 3	Sprk.	Occ. Load	Fac. Str.	Fire Alr.	Fire Wall	Features
1.1.1.1.	C,D										
1.1.1.2.		GT2									
1.1.1.3.	C+D				T			EQ2			
1.1.1.4.					F						Farm
1.1.1.5.	EX B2							1		T	
1.1.1.5.	EX B2	GT6						2		T	
1.1.2.1	C	LE3						GT1			

Fig. 3 Tabular Layout for Data Entry of Document Tagging

The building code expert enters the building attribute information in the proper columns, each row contains information for one provision. Keep in mind that multiple rows can be required for the same provision, as with provision 1.1.1.5. Features are entered in the last column in free text format. Any input mechanism can be used to record this information. In the original MiniCode, a word processor with tabular cells provided a user-friendly interface. This file can be easily exported to any format such as tab delimited ASCII or Rich Text Format (RTF) and can be parsed to the proper format for the MiniCode Engine. The parsing entails decoding the shorthand data entry required for the format presented above. For example "LE3" means less than or equal to 3 storeys in building height. This would be parsed into three tags for provision 1.1.2.1.: a tag is required for one, two or three storey buildings. Other shorthand notation includes "C,D,E" meaning C or D or E major occupancy, "C+D" for C and D major occupancies, "EX C" for all major occupancies except C, "EQ2" meaning equals 2, and "GT2" meaning greater than 2. The resulting file is called the Building Code Bitmap.

The Document Tagging is a time-consuming, labour-intensive task. Depending on the complexity of the building code and the knowledge level of the individuals involved, the time required to tag the document can average at 3 minutes per provision. In the case of the NBCC with 4000 provisions this equals over 5 person-weeks of work. Although this may seem to be a considerable amount of time, remember it is probably the first attempt to classify and structure the information contained in that building code. The writing of the parser may require a day or two.

2.4 MiniCode Engine

The MiniCode Engine is described in detail in Cornick and Thomas [2]. Basically, the Engine compares the Classification Description to the Building Code Bitmap and excludes those provisions that do not met the criteria.

The output from the MiniCode Engine includes the classification of the building along with the list of the remaining building code provisions, as shown in Fig. 4.

Project Name: **IABSE Demo**	Major Occupancies: **C**
Minor Occupancies: **none**	Number of Storeys: **1**
Fire Wall: **Not present**	Fire Alarm: **Not present**
Building Area: **500**	Building Code: **Part 9**

1.1 1.1.1 1.1.1.1 1.1.2 1.1.2.1 1.1.3 1.1.3.1 1.1.3.2 . . . . .

Fig. 4 MiniCode Listing

As mentioned earlier, the MiniCode generation takes approximately 2 seconds on a standard personal computer. In actual fact there are 183 bits required to encode the information in the Building Code

Bitmaps; so the Engine compares 4000 x 183 bits to extract each MiniCode. The writing of the Engine may involve a day or two.

2.5 MiniCode Viewer

The MiniCode Viewer reads the output of the MiniCode Engine and presents these to the user in the form shown in Fig. 5.

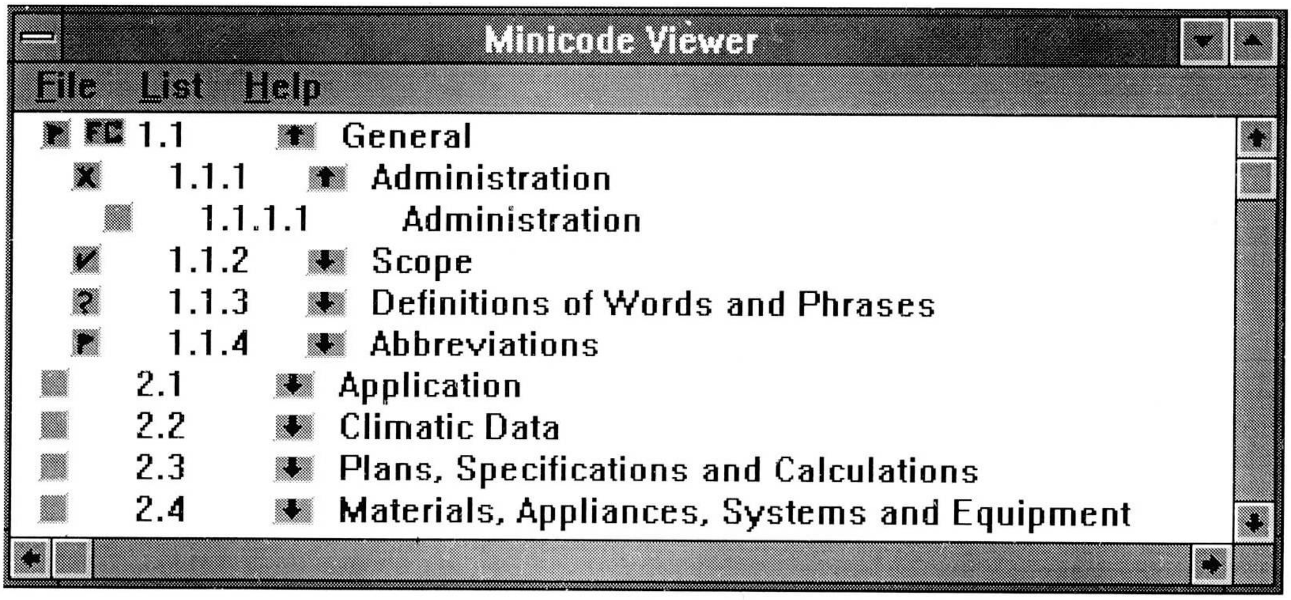

Fig. 5 MiniCode Viewer

The toggle boxes to the left of the Fig. 5 allow the user to record the status of compliancy for each of the building code provisions in the MiniCode. These toggles include an check mark, question mark, an x, or a flag, as shown in Fig. 5. The arrow immediately to the right allows the user to expand the contents as shown in Fig. 5.

The NBCC Document Viewer uses WinHELP™ to view a hypertext version of the building code. Fig. 6 demonstrates some of the possibilities of the environment. This viewer is similar to other hypertext environments available on personal computers [5, 1]. There are many tools now available to assist software developers create WinHELP™ applications. It is difficult to estimate the time required to develop such a tool, but less than 2 person-weeks were required for the NBCC Viewer.

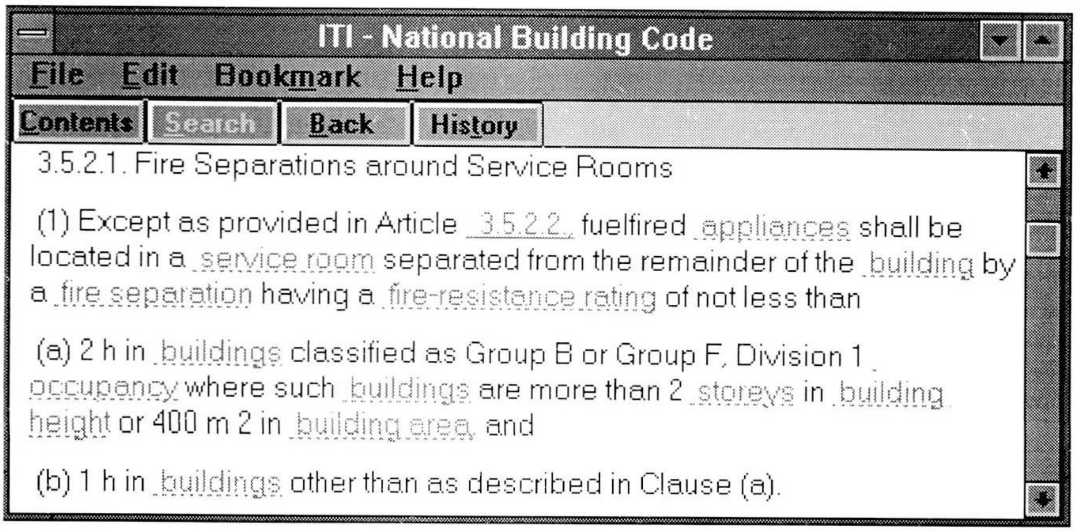

Fig. 6 NBCC Document Viewer

3. DISCUSSION

The purpose of this paper is to describe a product resulting from research into the generation of sparse building codes, and to encourage other researchers to develop similar tools. There are numerous advantages to this type of technology for researchers, software developers and users involved with building codes.

3.1 Researchers

Although this paper may resemble a cookbook for creating an electronic code software product, there is significant research required to identify the building attributes and features for individual building codes. In addition, there is knowledge engineering required to decipher the classification trees for the NBCC Building Classifier. This cannot normally be handled by a building code expert, as the expert understands the document too thoroughly and may not be able to relate the information to strict "if_then_else" rules. The researcher, *cum* knowledge engineer, must direct the activities of the expert to create a clean, concise, and accurate representation of the classification process.

On completion of any new NBCC Building Classifier, the code writing body must also be part of the feedback loop. If a specific building attribute cannot be accurately classified in the rule base by the knowledge engineer and building code expert, this normally implies that it is difficult for ordinary users of the building code to understand. This was encountered a few times in the Plan Examiner MiniCode when it was impossible to decipher when a specific provision could be excluded, or when it couldn't.

Unanswered problems still exist for the researcher. For example, how deep and how broad can the building attributes and features become? Can the MiniCode classification be extended to hundreds of features, or will a limitation be reached at 256 bits or 2 Kbits in the Bitmap?

3.2 Software Developers

Profit is <u>the</u> driving force in the development of software products, and there are few available electronic code tools. Typically the potential client for software developers is a code writing body, as most building code users desire some stamp of approval from the jurisdictional authority [6]. Code writing bodies are then the logical initiator for these software products because they have a vested interest and they have the expertise on staff to validate the rule base and create the tags for the provisions. To defend this position, the code writing bodies have been selling their products for decades and they provide updates, errata, supplemental instructions and other services: a software product is almost identical. It is strongly recommended that electronic products be part of the product offerings from code writing bodies [7]. It will not be long before most similar information is available electronically, and before users start demanding that type of service and product.

3.3 Building Code Users

The users are the main beneficiaries of electronic codes. Building codes are complex documents [8], and have remained so for decades. As construction techniques become more complex and as construction regulations increase because of increased liability and risk, free-trade, new products, etc., there is an ever-increasing need to assist the users in this building code quagmire [8].

The MiniCode Generator is one of the few attempts to codify the information in building codes [9, 10, 11, 12]. The technique offers advantages to novice and experienced users alike. For example, the NBCC Building Classifier serves as a tutorial for many of the factors concerning building classification. Help messages are available at every step in the classification tasks to explain the signification of specific selections, as shown in Fig. 7.

Novice users can learn as they browse, whereas experienced users can quickly gloss over the supporting literature and concentrate on data entry. Without sounding too much like promotional literature, using the NBCC Building Classifier is like having a building code expert as a colleague.

Generating sparse building codes in seconds reduces the amount of work for the plan examiner and the building designer alike. MiniCodes considerably reduce the amount of information to peruse, in some case MiniCodes are 20 % of the original document length. This is an added benefit to the users: they are guaranteed that the excluded provisions are not applicable, and need not be reviewed.

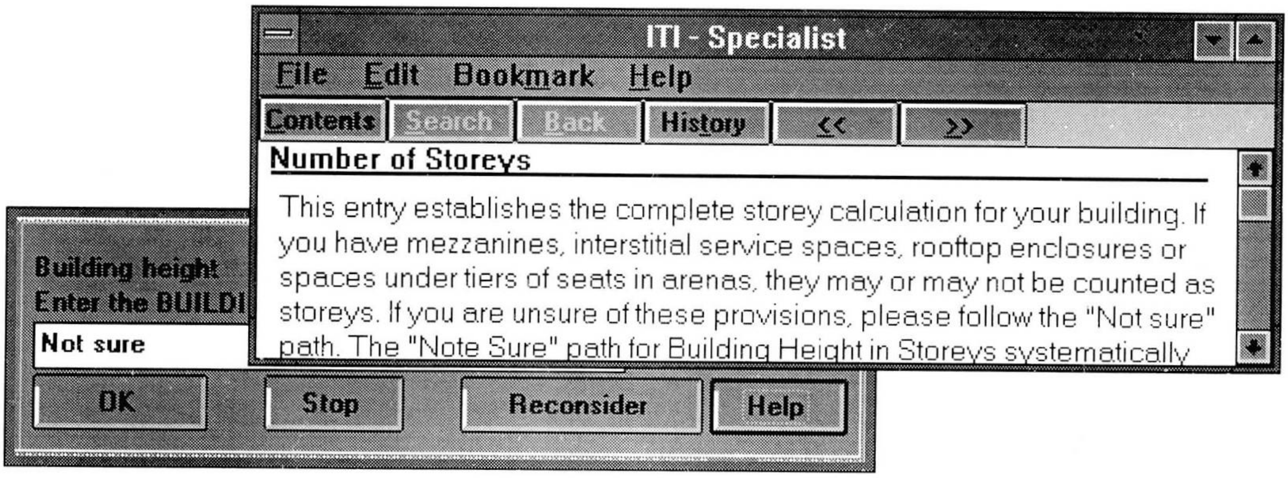

Fig. 7 NBCC Building Classifier Help System

The WinHELP™ version of the MiniCode Viewer is a simple-to-use, effective, browsing interface to building codes. It provides hot-links to related provisions and interactive defined terms.

3.4 Problems

Although the MiniCode Generator appears to be a robust, useable tool, the author feels that the significant contribution to the community is the development and structuring of the building code information for the NBCC Building Classifier. The generation of MiniCodes is interesting as a concept; however, even if the building code is reduced considerably, even 20 % of the 400 pages of the NBCC is a formidable building code.

The Document Tagging is labour-intensive, requiring a high level of expertise from the building code expert. In addition, there is need for constant validation of the data entry in the Document Tagging. For example, although one provision definitely applies to sprinklered buildings, the provision may identify a solution that is of interest to designers contemplating no sprinklers. Therefore the sprinkler tag should not appear for that provision.

Updates to building codes may be a problem. That is, changes to a building code have to be reviewed and retagged, and the NBCC Building Classifier may have to be restructured. However, experience has shown at IRC that changes to the NBCC, with respect to the classification tags, do not occur in significant numbers to warrant concern. In some instances changes in building codes may even alter the rule base: the current debate in Canada concerning sprinklers for all buildings would alter the NBCC Building Classifier significantly.

The NBCC is a model code for Canada that is modified by three provincial counterparts to suit regional requirements [4]. As such, the MiniCode technology can be easily transferred to these documents. However, it is extremely difficult to transfer the NBCC MiniCode application to other National codes. It would be equally difficult to transfer this application directly to other Canadian codes such as the National Fire Code of Canada.

3.5 Future Extensions

Although the Plan Examiner is still under *Beta* review, the current users have been very specific in their demands for future extensions. As a result, a number of modifications have been suggested: the Designer MiniCode could include "What if?" analysis and subject index comparisons.

These two features basically work on the same principle: the user can generate a number of MiniCodes and then compare any two at a time. In the "What if?" scenario the user might compare a two storey with a three storey variation. The "logical difference" between these alternatives would identify additional provisions that have to be investigated.

The subject index comparison can also be a valuable tool for designers: (1) create a MiniCode for a standard three storey townhouse; (2) create a MiniCode of provisions dealing with basements, and (3) the logical difference is a listing of all provisions for basements in three storey townhouses. This can have numerous applications for designers such as finding provisions dealing with sprinklers for commercial buildings, identifying mezzanine provisions for low-rise residential construction, or locating all provisions about fire alarms and two storey offices under 2000 square metres

Additional functionality in the Designer MiniCode could include validation of input by the user, the ability to append user-specified notation, the addition of more building attributes and features, and increased depth in the classification of building attributes. The first two of these features must be explained. The validation of input addresses potential erroneous or trade-off decisions by the designer. For example, a building may not require sprinklers; however, the designer may decide that sprinklers may permit combustible construction, a cheaper construction type. The NBCC Building Classifier informs the user that sprinklers are not mandatory, but still recognizes that all sprinkler provisions cannot be excluded. On the other hand, if novice users select this option incorrectly, they are informed of their error. The user-specified notation permits national, regional or local agencies to append additional notes or interpretations to specific provisions. For example, the Fire Commissioner of Canada might have a specific ruling on NBCC Section 1.1. In this case a special icon would appear in the NBCC Viewer beside that provision. This feature has been implemented for one government agency, as shown with the [FC] icon in the upper portion of Fig. 5. To access the relevant literature, the user clicks on the [FC] icon.

Other future extensions include interfaces to any version of electronic code, such as the provincial building codes. In the case of the NBCC MiniCode Generator, it could be linked to commercial packages currently available or to future products such as the upcoming Compact Disk (CD) version of the NBCC [13]. There is also the obvious possibility of including Building Code Bitmaps for provincial building codes in one MiniCode Generator, thus allowing users to select the appropriate provincial MiniCode.

4. CONCLUSIONS

The MiniCode Generator described in the paper is a simple implementation of existing technologies. The paper details a sequence of steps in the development of an information technology tool for building designers. These steps can be used by code writing bodies or software developers to create their own versions of the MiniCode Generator for their national or regional building codes.

The general goal of researchers in the area of electronic codes, or at least the envisioned panacea, is an interface to Computer Aided Design (CAD) systems [14, 1]. This is only possible through the proper classification and structuring of building code information. The MiniCode Generator is a step in the right direction, but falls short of the CAD interface envisioned by designers. Future developments such as detailed classification systems [1] could provide additional steps towards this goal. In any case, the classification or structuring of building code information can only assist the user community in the long term.

This paper describes the second stage [3] in the development of tools to assist architects, engineerings and building officials access building codes. The *Beta* trials generally indicate that users are pleased with the tool and its capabilities. However, the users have indicated the need for more functionality. More research is envisioned for the creation of the Designers MiniCode.

REFERENCES

1. VANIER, DJ, *A Parsimonious Classification System to Extract Project-Specific Building Codes*, PhD Thesis, Université de Montréal, Montréal, Québec, October 1994.

2. CORNICK SM, and THOMAS JR, HyperCode, MiniCode and ExperCode: Evolution of a Code Users Environment, Paper presented at *Management of Information Technology for Construction*, eds. Krishan S. Mathur, Martin P. Betts, and Kwok Wai Tham, Singapore, (n. pag.), August 1993.

3. VANIER DJ, Minicode Generation: A Methodology to Extract Generic Building Codes, *CAAD Futures '93: Proceedings of the Fifth International Conference on Computer-Aided Design Futures*, 7-10 July, Pittsburg, PA, pp. 225-239, 1993.

4. NBCC, Associate Committee on the National Building Code, *National Building Code of Canada*, 10th ed., National Research Council Canada, Ottawa, Ontario, 1990.

5. YABUKI N, and LAW K, Hyperdocument Model for Design Standards Documentation, *Journal of Computing in Civil Engineering*, Vol 7, No. 2, pp. 218-237, 1993.

6. Market Study, Institute for Research in Construction, *CD-ROM Market Study: Characteristics of the Market for Methods of High Density Electronic Storage and Delivery of Code and Standards Information*, by Opinion Search, Inc., National Research Council Canada, Ottawa, Ontario, 27 p., 1993.

7. VANIER DJ, THOMAS JR, and WORLING JL, Standards Processing 2000, In *Proceedings of ASCE First Congress on Computing in Civil Engineering*, Washington, DC, 20–22 June 1994.

8. FENVES SJ, RANKIN K, and TEJUJA HK, *The Structure of Building Specifications*, NBS Building Science Series, No. 90, Center for Building Technology, National Bureau of Standards, Washington, DC, 77 p., 1976.

9. CRONEMBOLD JR, and LAW KH, Automated Processing of Design Standards, *Journal of Computing in Civil Engineering*, Vol. 2, No. 3, pp. 255-273, 1988.

10. GARRETT JH Jr., Object-Oriented Representation of Design Standards, In *Proceedings of International Association of Bridge and Structural Engineering (IABSE) Colloquium*, Bergamo 1989, IABSE-AIPC-IVBH, ETH-Honggerberg, Zurich, pp. 373-382, 1989.

11. DE WAARD, M, *Computer Aided Conformance Checking*, Self Published Ph.D Thesis, Melis Stokezijde 11, 2543 CA, The Hague, The Netherlands, 203 p., 1992.

12. SHARPE R, OAKES S, HASELDEN P, and DAVIDSON I, Automation of the Building Code of Australia, Proceeding of the Third World Congress of Building Officials, New Orleans, LA, Session 5, 1-6 May (n. pag.), 1993.

13. VANIER DJ, MELLON BS, WORLING JL, and THOMAS JR, Management of Construction Information Technology, In *Proceedings of Management of Information Technology for Construction*, eds. Krishan S. Mathur, Martin P. Betts, and Kwok Wai Tham, Singapore, , pp. 75-84, August 1993.

14. DYM CL, HENCHEY RP, DELIS EA, and GORNICK S, A Knowledge-Based System for Automated Architectural Code Checking, *Computer-Aided Design Journal*, Vol. 20, No. 3, pp. 137-145, 1988.