

**SEITRON
SEIsmic ElecTRONic Handbook**

A Knowledge Based Systems Approach to Structural Steel Design

By

Lillian S.F. Siu

B.A.Sc. (Civil) The University of British Columbia 1990

**A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
MASTERS OF APPLIED SCIENCE**

Department of Civil Engineering

We accept this thesis as conforming to the required standard




THE UNIVERSITY OF BRITISH COLUMBIA

November 1993

© Lillian S.F. Siu, 1993

In presenting this thesis in partial fulfilment of the requirements for an advanced degree at the University of British Columbia, I agree that the Library shall make it freely available for reference and study. I further agree that permission for extensive copying of this thesis for scholarly purposes may be granted by the head of my department or by his or her representatives. It is understood that copying or publication of this thesis for financial gain shall not be allowed without my written permission.

(Signature) 

Department of CIVIL ENGINEERING

The University of British Columbia
Vancouver, Canada

Date 27 SEPT. 93

Abstract

The Canadian steel code, CAN3-S16.1-M89, has become increasingly complex with the introduction of new seismic provisions, Clause 27, in its 1989 edition of the Handbook of Steel Design. The implementation of these provisions is meant to provide the necessary ductility and redundancy requirements for the basic types of steel frame construction. Prior to the 1989 changes, it was reasoned within the industry that the inherent mechanical properties of steel and the standard design practices provided adequate ductile frame behaviour. However, tests performed in the 1970's and 80's have shown successful behaviour of some frames and less desirable characteristics in other.

The purpose of Clause 27 is to quantify design practices such that an adequate level of ductility and redundancy is reached for the basic types of steel frames.

Further to the concept of steel member design, there are numerous software packages available for structural analysis using finite elements, static and dynamic elastic theory; however, there are very few design packages which transpose the forces from the analysis software into quantifiable results of beam, columns and connections.

The objectives of this thesis is to clarify and illustrate the design principles as stated in Clause 27 of the HSC Edition 91 and to expand the engineer's toolkit with elemental design tools. This is done in three ways:

- 1 Provide an electronic reference system for Clause 27, along with electronic links to commentary and references

- 2 Quantify the critical elements of each frame system and provide the corresponding code design requirements through the use of graphics.
- 3 Provide design templates as engineering design tools which utilize Clause 27 other CAN3-S16.1-M89 design requirements.

The approach taken in the design of SEITRON, (from SEIsmic ElecTRONic Handbook), is to utilize the present state-of-the-art computer technology in order to provide the qualifying relationships and links that are inherent in the new code. Also, the creation of a WINDOWS environment coupled with the use of graphics is intended to effectively illustrate the application of the code with respect to the particular frame element and frame type.

In terms of expanding the engineer's toolkit, SEITRON bridges this gap with structured spreadsheet templates which incorporate the CAN3-S16.1-M91 steel design code. The resultant forces from structural analysis can be placed in the templates for automatic or user defined member selection and elemental strength and stability checks. The structured templates increase the engineers efficiency and decreases the time spent on design.

The development of SEITRON promises many benefits to the steel design industry as an engineering tool or as a teaching aid. Productivity and efficiency of design is inherent with the use of the structured templates, especially when member selection or member verification is desired for structures that are composed of hundreds to thousands of elements. SEITRON also provides greater comprehension as to the application and rationale of the seismic provisions with respect to ductility design through its electronic referencing system.

SEITRON can play an important role for the education of students as well as practicing engineers who would like to become familiar with the new seismic code. It can provide a practical tool for elemental design.

Table of Contents

Title Page	i
Abstract	ii
Table of Contents	v
List of Figures	viii
List of Tables	xiii
Acknowledgments	xv

CHAPTER 1

INTRODUCTION TO AN ELECTRONIC DESIGN HANDBOOK

A. Introduction	1
B. Purpose and Justification	3

CHAPTER 2

A CONCEPT FOR AN ELECTRONIC DESIGN HANDBOOK: THE HYPERTEXT SYSTEM

A. Definition	6
B. Historical Highlights	8
C. Advantages of Hypertext	10
D. Disadvantages of Hypertext	14
E. Applications of Hypertext	17

F. Anatomy an Expert System	18
i. Inference Engine	20
ii. User Interface	20
iii. Knowledgebase	22
a. Knowledge Acquisition	24
b. Knowledgebase versus Database	27

CHAPTER 3

OPERATIONS

A. Components of SEITRON	30
B. Clause 27 Knowledge Base	32
i. Level 1	35
a. Table of Contents	37
b. Index of Topics	38
c. Map Organization	39
ii. Level 2 and 3	40
iii. Example of Clause 27 Knowledgebase	41
C. Ductility and Redundancy Requirements	50
i. Ductile Moment Frame	53
ii. Nominally Ductile Moment Frame	55
iii. Ductile Braced Frame	57
iv. Nominally Ductile Braced Frame	60
v. R-Factor	62
vi. Ductility Requirements	63
D. Spreadsheet Design Templates	
i. General	65
ii. Formatted Spreadsheets	66
iii. Structured Spreadsheets	67
iv. Batch Mode	70
v. Spreadsheet Advantages	71

E. Template Design Component	
i. General	72
ii. Interface and Communication	74
iii. Overview of Template Design Modules	76
iv. Features of Design Templates	79
v. Modification of Templates	80
F. Design Template Example	80
i. Column Example	83

CHAPTER 4

SUPPORTING INFRASTRUCTURE

A. Cursor Appearance	94
B. Types of Buttons	97
C. Menu Line	100
D. Use of Help System	102
E. Knowledgebase Button	104
F. Information Library	105
G. Shape Library	110

CHAPTER 5

CONCLUSIONS	115
BIBLIOGRAPHY	120
BIOGRAPHICAL INFORMATION	123

List of Figures

Figure 1	
Hypertext System of Reference Manual	13
Figure 2	
Hypertext System for Tutorial	13
Figure 3	
Hypertext System of Brochure	13
Figure 4	
Expert System Software Structure	19
Figure 5	
Framework of Knowledge	26
Figure 6	
SEITRON Title Page	30
Figure 7	
SEITRON Main Index	31
Figure 8	
Structure of Clause 27 Knowledgebase	33
Figure 9	
Clause 27 Knowledgebase Relationship between Levels	35
Figure 10	
Access to Clause 27 Knowledgebase via the Table of Contents or Index	36
Figure 11	
Table of Contents	37
Figure 12	
Index of Topics	38
Figure 13	
Map Organization	39
Figure 14	
SEITRON Title Page	41

Figure 15 SEITRON Main Index	42
Figure 16 Access to Clause 27 Knowledgebase via the Table of Contents or Index	43
Figure 17 Table of Contents Revealing Ductile Braced Frames Topics	44
Figure 18 Investigation of Diagonal Bracing Members within Ductile Braced Frames	45
Figure 19 Diagonal Bracing Members Code Specification to Clause 27.4.3.1	46
Figure 20 Diagonal Bracing Member Commentary to Clause 27.4.3.1	47
Figure 21 References to Clause 27, Liu	48
Figure 22 Map of Seismic Clauses Illustrating Code Requirements for DBF's	49
Figure 23 Frame Ductility Requirements Title Page	51
Figure 24 Access to Frame Ductility Requirement via the Index of Topics	52
Figure 25 Frame Index	52
Figure 26 Ductile Moment Frame	53
Figure 27 Ductile Moment Frame with Beam as Critical Element and Design Requirements for Beams	54

Figure 28	
Nominally Ductile Moment Frame	55
Figure 29	
Nominally Ductile Moment Frame with Panel Zone as Critical Element and Design Requirement for Panel Zone	56
Figure 30	
Ductile Braced Frames	58
Figure 31	
Ductile Braced Frames with Design Requirement for Factored Compressive Resistance	59
Figure 32	
Nominally Ductile Braced Frames	60
Figure 33	
Nominally Ductile Braced Frames with Design Requirements for Bracing Members	61
Figure 34	
Effective Earthquake Base Shear Reduction Factor	62
Figure 35	
Ductility and Redundancy	63
Figure 36	
Frame Configuration and Component Design Force Comparison	64
Figure 37	
Design Module Index	76
Figure 38	
Element Design Modules	77
Figure 39	
General Design Modules	77
Figure 40	
Frame & Composite Design Modules	78

Figure 41	
Compressive Resistance of Column, Class C	85
Figure 42	
Compressive Resistance of Column, Class C with Help Box	86
Figure 43	
Results for the Compressive Resistance of the Design Example Column showing the Deformed Shape	87
Figure 44	
Activation of the W Shape Library for the Compressive Resistance of Column, Class C	88
Figure 45	
W Shape Index	89
Figure 46	
W250 Selection Index	90
Figure 47	
W250x49 Member Geometric Properties	91
Figure 48	
Definition of "kx" from the Info Book	92
Figure 49	
Effective Length Factor Data from the Info Book	93
Figure 50	
Menu Line for Clause 27 Knowledgebase	101
Figure 51	
Menu Line for Ductility and Redundancy Design	101
Figure 52	
Menu Line for Template Design	101
Figure 53	
Bending Resistance of I-Sections with Help Box	103
Figure 54	
Help Info Library Activated to Show Info on " $(KL/r)_x$ limit"	106

Figure 55 Cover Page of Seismic Info Book	107
Figure 56 Title Page of Seismic Info Book	107
Figure 57 Index of Variables from the Info Book	108
Figure 58 Beta Information from Info Book 109	
Figure 59 W Shape Title Page	110
Figure 60 Access to W Shapes via Index	111
Figure 61 Shape Table Selection	111
Figure 62 W Shape Index	112
Figure 63 W250 Selection Index	113
Figure 64 W250x49 Geometric Properties	113
Figure 65 Good-bye!	119

List of Tables

Table 1 Principles of User Interface Design	21
Table 2 Nodal Relationships	23
Table 3 Navigation of Nodes	24
Table 4 Knowledgebase versus Database	28
Table 5 Template Features	79
Table 6 Column Design Example	87
Table 7 Cursor Appearance	95
Table 8 Button Appearance	97
Table 9 Description of Menu Line Button	101

Acknowledgments

"It is better to know some of the questions than all of the answers."

James Turber
Journalist

This thesis is dedicated to those people who have shown unfailing support in me over the years.

Of especial thanks I give to the Natural Science and Engineering Research Council (NSERC), and the Steel Structures Group of British Columbia for their scholarships which funded my research. With respect to industrial support, David Halliday of Coast Steel Fabricators is instrumental in providing me with the practical aspects of engineering design.

I am particularly grateful to Dr. Siegfried Fritz Stierner, my graduate advisor, who saw potential in me and encouraged my development.

Eternal thanks and love to my parents, Christine and Joseph who believed "engineering is no place for a woman" when I started my undergraduate work but have always been my most ardent fans. They will forever brag of my profession to friends and strangers alike.

Finally, I would also like to thank Vincent Latendresse for providing computer support with SEITRON and for putting up with my seemingly endless and inane questions on why the program does not do what I programmed it to do.

CHAPTER 1

INTRODUCTION TO SEITRON

A. INTRODUCTION

The mass production of silicone chip technology has created inexpensive computing power. These hardware advancements have created a vast and diverse industry of software development. The computer is no longer limited to scientists and business personnel as analysis tool but has reached a level of everyday dependency. This dependency may be witnessed in all facets of North American life from the elimination of commuting to work while using the home office with modem and fax communication to the use of computers to sign books out of the library.

This information explosion is benefiting the user through the presentation of information via a plethora of graphics, video clips, animation, music, audio - multi-media. The user is no longer restricted to a dull monochromatic screen containing rows and columns of data. Psychologists agree that the use of graphics, colors, sound, video, enhance the perception of information and increase its retention value.

With the availability of the personal computer, PC, and its powerful computational and information management advantages, it is no wonder, that the PC has traded places with the "traditional" engineering tool of the slide rule. It may even ambitiously replace engineering design textbooks.

Within the structural engineering community, the greatest use of the PC is for structural analysis. The multitude of commercial software created for the structural engineer allows the designer to easily specify the degree of high order analysis required when given an array of boundary conditions and constraints. Software created for this purpose include those that perform static

and dynamic as well as finite element analysis on structures. The desired output may include resultant stresses, forces, reactions and deflections.

Very often, the role of the software is finished once the forces are determined; however, the design process is only partially complete. The engineer is left on his own to make use of these forces by comparing these values with resistances of beams, columns, bracing and connections derived from the code guidelines.

Another aspect of the design is in the sheer size of the structure which may be reflected in the large number of elements which must be sized. Depending on the dimension and complexity of a structure, the designing engineer may have to consider hundreds to thousands of elements with forces, displacements and rotations. For example, such a structure may be a telescope enclosure.

During the design process, the actual computing time spent on analysis is minimal. However, due to the large volume of variables and the sophisticated calculations required to meet code strength and stability criteria, the iterative design of each element may take considerable time using traditional calculator and paper methods.

Whereas the software market seems to be saturated with analysis programs, there are very few packages that design and check the appropriateness of the element with the structural design code, whether that be the concrete or steel design handbook. A tool which combines computational skill with organization ability is beneficial to constrain the iterative member selection cycle to hours and days instead of weeks and months.

Further to the concept of a code oriented design tool, it would be beneficial to increase the toolkit repertoire with electronic access to the structural design code. In SEITRON's case, the design code is that of the CAN3-S16.1. Advantages of an electronic code would be that code

requirements and theoretical background are linked such that the engineer may better comprehend the rational of the design principles. Linked ideas should not be limited to text but may encompass graphics, animation and other forms of multi-media.

B. PURPOSE AND JUSTIFICATION OF SEITRON

SEITRON attempts to address steel design needs within the academic and practicing engineering context. The objective of this thesis is to expand the engineer's information network electronically and to expand the instruments in a design toolkit.

SEITRON stands for SEIsmic elecTRONic Handbook and tries to emulate an electronic version of a traditional steel design handbook. SEITRON helps to clarify the new Clause 27 Seismic Provisions as set out in the Handbook of Steel Design, CAN3-S16.1-M91 and provides design templates for member selection.

Part of the motivation of this thesis is the fact that as seismic research progresses, the codes, which make use of these new findings, become progressively more complex and difficult to follow. The Canadian steel code has become increasingly elaborate with the introduction of the new Clause 27.

The implementation of these provisions is meant to provide the necessary ductility and redundancy requirements for the basic types of steel frame construction. Prior to the 1989 changes, it was reasoned within the industry that the inherent mechanical properties of steel and the standard design practices provided adequate ductile frame behaviour. However, tests performed in the 1970's and 80's have shown successful behaviour of some frames and less desirable characteristics in other. The purpose of Clause 27 is to quantify design practices such

that an adequate level of ductility and redundancy is reached for the principle types of steel frames.

However, two factors limit its practical use: the inter-relativity of the clauses and the sometimes limited theoretical seismic background of new practicing engineers.

Part of the confusion of the new seismic provisions is in its inter-relational design requirements. Investigation of the clauses reveals a referral or a designation to other clauses, commentary or references. Although the objective of the linkages between clauses and other material is meant to interpret or to avoid repetition of the constraints, the format of the clauses tends to confuse rather than to clarify.

Furthermore, each structural component of each frame must be verified as a critical or non-critical element. Based on the condition of criticality, certain rules must be followed. Unfortunately, the code is written in such a fashion that it allows for confusion of when to apply the clauses. That is, the conditional environment of "if A of frame X is critical then follow the design practices of C from frame Y when C is critical" is evident.

Also, sometimes the comprehension of practicing engineers with respect to seismic theory is relatively basic. Therefore, an inexperienced engineer who combines the new design standards with a basic understanding of the fundamentals is faced with elaborate and complex design practices instead of simplified procedures.

The objective of this thesis is to attempt to clarify and illustrate the design principles as stated in Clause 27, CSA structural steel design standard, CAN3-S16.1-M89, that comprises part of the HSC Edition 91, and provide a practical instrument for design. This is done in three ways:

1. Provide an electronic reference of Clause 27, along with electronic links to commentary and references.
2. Quantify the critical elements of each frame system and provide the corresponding code design requirements through the use of graphics.
3. Provide design templates as engineering design tools which utilize Clause 27 other CAN3-S16.1-M91 design requirements.

The approach taken in the design of SEITRON is to utilize the present state-of-the-art computer technology in order to provide the qualifying relationship links that are inherent in the new code. SEITRON is operated through a WINDOWS environment of object manipulations and icon commands which provide ease of use. Also, the utilization of graphics effectively illustrates the application of the code with respect to the basic frame type and particular frame elements of beams, columns, bracing or connection.

In regards of the template design, the user has the option of working on individual elements or using a batch mode process. During the latter, the computer automatically chooses members to optimize according to strength and stability conditions.

The development of SEITRON represents a promising start to improve and ease the steel design process. It can be used as an engineering tool as well as a teaching aid. It provides the possibility of gaining greater comprehension of the application and rationale of the seismic provisions with respect to ductility design. In the future, SEITRON may play in the future a role in the education of students as well as practicing engineers who are unfamiliar with the seismic code.

CHAPTER 2

A CONCEPT FOR AN ELECTRONIC DESIGN HANDBOOK: THE HYPERTEXT SYSTEM

A. DEFINITION

Hypertext is the term used to describe a system of associations of informational nodes with relational links. The focus of this method is the control given to the user to explore a complex web of, logically interlinked information according to his needs and interest. That is, the purpose of hypertext is to allow the user to choose his own course of reading and browsing with the aid of an electronic information medium.

This interaction between user and medium goes beyond that of conventional question/query or computer-user alternation of turns that is found in traditionally structured flowchart techniques. The progression of hypertext can be unpredictable and is dictated by the desires of the user. Traditional programming techniques follow a flowchart style of advancement where the programmer predefines the next course of action. However, in hypertext, there is no fixed route computed in advance and thus knowledge or information from a database is accessed "on the move". The defining feature of hypertext is the sensation of close involvement of the user when such interactivity is inherent.

Information in a hypertext system is stored as *nodes* and the relationships that connect the nodes are termed *links*. Nodes containing purely textual information define the "text" part of "hypertext". However, information is not restricted to text but may contain other types of media such as images, graphics, animation, video, and sound recordings. Thus the linkage of multiple forms of media is called *hypermedia*. The aspiration of hypertext and hypermedia is the electronic transfer of knowledge from the computer to the user.

At present, hypertext is still in a state of technological infancy. Nonetheless, there has been extensive development in the software and hardware technologies which advanced hypertext technology today. The growing success in hypertext comes with the use of the personal computer, (**PC**), for information management retrieval. In terms of hardware developments, the falling cost of computing power coupled with the increased capacity in storage and optical disk technology (or **CD**) have led the way to high quality graphics and imaging. Developments in networking further fuel the information generation.

Advances in hardware are not sufficient without the advancements in software. Both are continuing to allow for applications which strive to be user friendly. There are two main software advances in PC technology which have made hypertext technology possible today: the graphical user interface (or **GUI**) and object-oriented programming (or **OOP**). [2][34]

The graphical user interface makes use of graphical representations to perform tasks. As a result, these interfaces make computers easier to use through the manipulation of metaphors and icons. Graphical icons provide more powerful, flexible and direct representation of tasks and data than previous character-based systems.

Object oriented programming allows a hypertext application to be divided up into discrete modules of objects. For example, within a hypertext application, there are objects consisting of fields, buttons and graphics. Each module or object is coded with object specific characteristics and functions. The program operates through manipulation of objects. Hypertext is created through a composition of such objects. The advantage of **OOP** is that it reduces redundancy and repetition of code from traditional programming.

A more detailed discussion of these advances as they apply to **SEITRON** is presented in successive chapters.

B. HISTORICAL HIGHLIGHTS

Although the concept of information retrieval via the process of organization and linkage is not new, it has not been realized until recent technological developments occurred. In the 1940's one of the pioneers of information management, Theodore Nelson, predicted that books would be obsolete by the year 1967. However, with software technology only recently having reached the ability to manipulate some of these ideas, it will be a while before hypertext replaces books, if ever.

One of the articles most cited as the birthplace of hypertext is Vannevar Bush's "As we may think" (Bush, 1945). Bush was appointed the first director of America's Office of Scientific Research and Development by President Roosevelt in 1941. Even at this time, Bush was experiencing the beginning of the information explosion and saw problems associated with the rapid growth of information: [25]

"There is a growing mountain of research. But there is increased evidence that we are being bogged down today as specialization extends. The investigator is staggered by the findings and conclusions of thousands of other workers - conclusions which he cannot find time to grasp, much less to remember, as they appear. Yet, specialization becomes increasingly necessary for progress, and the effort to bridge between disciplines is correspondingly superficial."

With this plethora of information, Bush had a vision of a device which he called the *memex*, which like "an individual stores his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility". The *Memex* was to simulate the human brain in that it would use associative indexing, and not be a simple repository, "the basic idea of which is a provision whereby any item may be caused at will to be selected immediately and automatically after another". Hence the non-linearity of the system lends itself as an effective teaching medium. However, the microfilm technology of the time did not lend itself to the level and complexity of indexing, retrieval and storage capacity required by the *Memex*.

Although Bush defined the principles of hypertext, he did not name this field of endeavor. The term *hypertext* was created by Theodor Nelson in the late 1960's. Since that time, Nelson has taken on the ambitious role of designing a network-based hypertext system capable of storing and providing access to the world's complete stock of textual material, the *Docuverse*. Nelson has developed principles that will ensure that the system is capable of uniquely referencing, billions of times, textual material and their interrelationships and the copyright issues arising from their use and distribution. One key feature is that a piece of text is never written twice, in that each document contains links to the original document rather than copies of its parts. Not surprisingly, the Docuverse is presently incomplete.

One of the first serviceable computer facilities which utilized a hypertext environment was invented by Doug Engelbart. His H-LAN/T system, *Human using Language, Artifacts and Methodology*, was used as an environment to support collaborative work at Engelbart's *Augmented Human Intellect Research Centre at Stanford Research Institute*. The system supported documents, memos, notes, reports, planning, debugging and communication. Engelbart's emphasis on the system was on augmenting or amplifying human intellect. As such his system was a support environment, in that the user was able to achieve more than would be possible without it.

Since then, interest and activity in hypertext and hypermedia has grown steadily. Numerous research prototypes and commercial hypermedia systems have been developed. As awareness of this technology grows, there is a trend for the user to not go through the laborious routine of programming his own interface but towards using commercially available interfaces such as *HyperCard* to create his own application.

C. ADVANTAGES OF HYPERTEXT

The present use of hypertext is analogous to that of books. With hypertext as well as with books, the user is gaining information by browsing and reading. However, the difference between a hypertext system and that of a book is that the user traverses through the electronic system by choosing the subjects that are of interest to him. He is not restricted, as with a book, to linearly follow information page by page but may jump from topic to topic as they are semantically related. Such a system with its high degree of user interactions and organizational aspects has much potential in the freedom of accessing information. [25]

The concept of an electronic encyclopedia with direct access between interrelated subjects defines a topic-oriented system. Furthermore, the value of a topic-oriented system is strongly related to the completeness and variety of its information content. The greater the variety of form, style, media and point of view is, the greater is the potential for the user to explore, penetrate and understand a topic.

The advantages of hypertext systems are expressed in terms of

- ♦ nonlinear organization,
- ♦ modeling of associative thinking,
- ♦ multiple information paths, and
- ♦ linkage of large libraries.

Nonlinear Organization

Freedom in organizing documents is achieved. The writer/programmer may express the nonlinear aspects of a body of information through a web network. Each topic may be associated with a similar topic with related links. The organizational effect is that of a large database with nodes of information connected with links which represent the relations among the nodes.

The result is that one topic does not necessarily follow another as in a book format but hypertext allows the user to express non-linear aspects of information that are currently expressed by cross-references, side base articles, digressions, bibliography citations, and footnotes. Thus, in creating the hypertext system, the writer shapes the document or system to fit the information instead of forcing the information to an arbitrary structure. Similarly, the user may navigate through the system by accessing information that is pertinent to him.

Hypertext Models Associative thinking

A hypertext system is similar to the methods used by the human brain. Associative, non-sequential linkages are comparable to the semantic networks of human memory. Psychologists claim that the manner of thought in a human brain is pluralistic rather than sequential. The undertakings of the mind explore multiple thoughts in a seemingly random manner instead of in a directed, sequential undertaking. That is, one thought may trigger another idea that may not be totally related to the initial thought in the conventional sense. By incorporating the parallel path workings of the human brain into the hypertext system, this suggests that information in hypertext form is easier to learn, understand and remember.

More Paths to Information

The same information may be organized in several different ways at once. Hypertext puts the organizational flexibility and power of a document knowledgebase to use by stringing together common modules of information to create many different types of documents. As a result, the user may produce personalized documentation for audience-specific documentation. For example, a hypertext system may be navigated in such a way to produce a reference manual, traversed with other nodes to produce a brochure or directed with particular subjects to produce a tutorial.

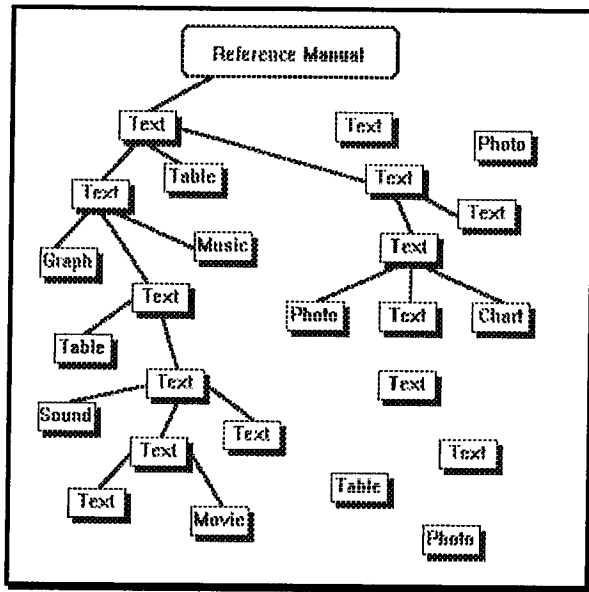


Figure 1, Hypertext System of Reference Manual

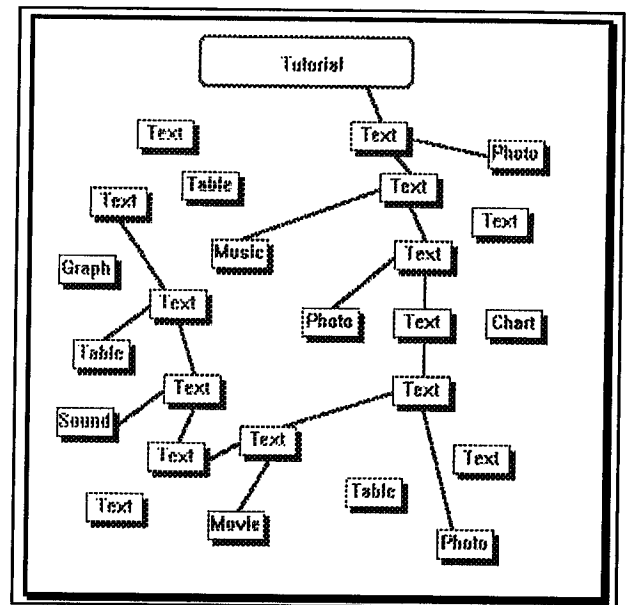


Figure 2, Hypertext System of Tutorial

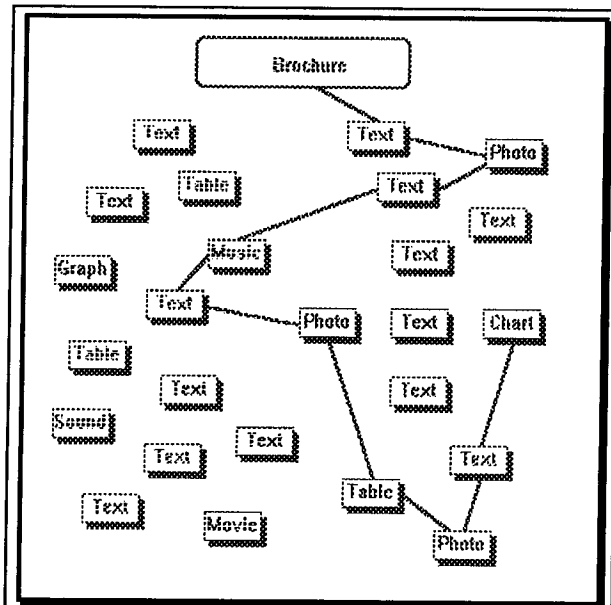


Figure 3, Hypertext System of Brochure

Linkage of Large Libraries

Hypertext allows users to link information comprising of knowledge bases and media types in various ways while leaving the individual parts modified. Each node of information is a separate entity which the programmer may change or edit. The links are based on attributes of the nodes. The nodes and links are distinctive entities and exist independently of each other. In order for the system to operate, the attributes which the links are dependent on must be present in the node. Thus it is feasible to connect unlimited arrays of databases to the system.

This seemingly simple concept of linkage of various media types was only recently accomplished with advances to software technology. Much of the difficulty in the linkage of text and graphics occurs due to the variation and incompatibility between interfaces. The challenge for software engineers is to create media integration interfaces and database linkage techniques which would provide easy and powerful access to information of all media types.

D. DISADVANTAGES OF HYPERTEXT

The practicality in a hypertext systems lies in its ability to:

- ♦ handle a large amount and variety of information,
- ♦ support many types of users with different skills and information needs,
and
- ♦ incorporate the vast quantity of existing information available in traditional
linear form.

However, the problems associated with such as system arise as the system expands in size. The use of a hypertext-style database of linking relevant information to each other poses three main problems:

- ♦ Information may be difficult to find,
- ♦ Users may become *Lost in Hyperspace*, and
- ♦ Hypertexts can be tedious to create and maintain.

Finding Information

As the number of items in a hypertext grows, so does the number of potential connections. The respective paths from a given item quickly exceed the optimal "seven plus or minus two"[4] choices to present to a user at a given time. The "seven plus or minus two" programming rule restricts the paths available such that the user is constrained to follow a directed path.

When too many choices are made available to the user, he is left with a dilemma or search sub-problem in which he must determine the optimal next choice. Consequently, as the number of paths increase, so does the complexity in organizing relationships and creating the ability to return to a specific topic or locate a specific topic.

Users traditionally rely on the programmer to determine the sequence of topics and to employ conventional cues that signal relationships among topics. Such organizational cues may be information management classified by relative importance or by chronology. When an organization network is not obvious, information becomes hidden and thus useless.

Although the impression of non-sequentiality is apparent to the user, the writer/programmer must focus on information management of schemes or informational hierarchies to organize data. The

links between ideas must also be discrete and explicit in order to lead the user from one subject to another and allow him to see the direction or progression of the link.

Lost in Hypertext

Hypertext systems may contain a variety of information management schemes. However, within its various hierarchies and networks a clearly defined organization is required such that users may be able to follow the rational of relationships. The combined effects of organization uncertainty with the large number of topics or items creates disorientation problems for the user. Users may get lost in the network. A rapid succession of jumps between items may appear confusing to the inexperienced user. Also, forcing users to backtrack adds to anxiety and confusion.

A possible solution to getting lost is the use of graphics, such as a map in order to structure information. The representation of each item of information with respect to its context in a map may be a basis for user reorientation. However, the breadth of information that may be represented in large knowledgebases can make such mapping schemes difficult to implement and maintain. Such a system would need constant upgrading when links and nodes are added.

Creation and Maintenance of Hypertext

The author decides the coordination and relationships between information points. If obscure paths are followed, it may be hard to locate the proper or any information. Likewise, if the author's associative powers are weak, meaningless links could be created and a worthless database might be generated. To ensure potentially successful use of the system, the programmer must be aware of the logic and organization behind the links and its impact on the user.

Also, since one of the advantages of hypertext lies in the use of large multi-media systems, the issue arises as to which author is given the task of pruning and editing the system. The complication created by multiple authors create the additional responsibility of assigning a single author or many to decide on the relevance and suitability of links. Needless to say, the system becomes difficult to maintain.

E. APPLICATIONS OF HYPERTEXT

The characteristics of hypertext and hypermedia make it well suited to applications involving a high degree of user interaction. While hypertext applications are presently used in computer-based training and education, its usage is rapidly expanding to cover such applications as:

- ♦ entertainment,
- ♦ electronic publishing,
- ♦ on-line information systems,
- ♦ computer-aided learning systems,
- ♦ diagnostic and repair manuals,
- ♦ group support systems, and
- ♦ multimedia authoring systems.

F. ANATOMY OF AN EXPERT SYSTEM

Hypertext systems are part of the computer science which deals with artificial intelligence, (*AI*). The concept of AI is the development of computer-supported techniques which could emulate some of the natural capacities of the human brain. AI encompasses many topics, from computer speech recognition and vision to different levels of problem solving. The development of AI programs which in some way imitate the behaviour of skilled human experts in problem solving is called expert systems. While expert systems are a sub-branch of AI, Hypertext is also part of the domain of expert systems. [3] [17][22]

While the scope of this thesis is not to cover the discussion of knowledge engineering design principles, a general overview of the basic concepts are useful for completeness.

The philosophy of artificial intelligence programming is to extract practical knowledge that is gained by experience and exposure and organize it into rules such that when a problem is presented, the system may replicate the problem-solving techniques of an expert and provide a rational conclusion.

The key characteristics of an expert system are

- ♦ the system makes inferences and deductions from information provided to it,
- ♦ knowledge is applied in order to solve the problem,
- ♦ the problem is narrow and specifically defined,
- ♦ the system is designed to cope with uncertainty,
- ♦ the knowledge base is used to guide and constrain the search for a solution, and

- ♦ the system questions the user while in a conventional program, the user interrogates the machine.

Although hypertext does not necessarily make use of rule-based logic and associated uncertainty of rules that are inherent in artificial intelligence systems, its principles of the linkage of knowledge adheres to the concept of information and knowledge transfer of expert systems.

The basic components of an expert system consist of the user interface, the inference engine and the knowledgebase. Figure 4 describes these components. [3]

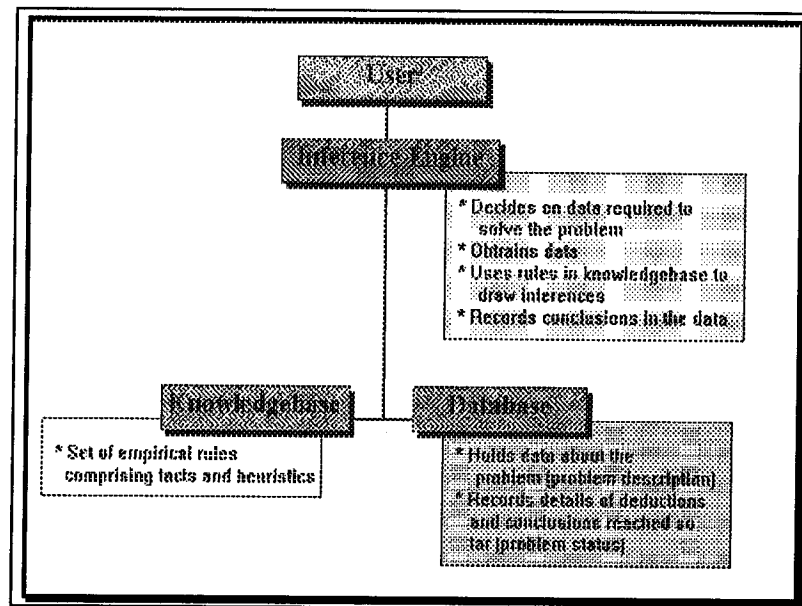


Figure 4, Expert System Software Structure

i. INFERENCE ENGINE

The inference engine is the protocol for which rules and data from a knowledgebase are processed. In conventional programs, its execution is dictated by the programmer who selects the sequence of operations. However, in an expert system, this predefined structure does not exist and the progression of the program is open to the user's needs. The task of the inference engine is to select and apply the appropriate rules that the user requires as he progresses through the system.

The inference engine is supposed to perform the following activities:

- ♦ decide on data required to solve the problem,
- ♦ obtain data,
- ♦ use rules in knowledge base to draw inferences when used in an expert system or implements linkages in a hypertext system,
- ♦ record conclusions in the data base.

ii. USER INTERFACE

The term *user interface* refers to the front-end environment which acquires and displays data and results. Thus for an interactive computer system, the focus is placed on the screen layouts and interaction scenarios. Generally, it is not sufficient to simply provide a menu of choices in which to guide the user. [3][6][16]

The principles of user-interface design determine both the amount of data transferred and the number of interactions between the computer system and the user's terminal in a manner to meet time and presentation guidelines.

When the interface is not discrete in its hierarchical menus or organization, time is wasted through navigation of menus and submenus which may in turn dominate the total transaction time. Careful design of the interface can reduce unnecessary waste of time. Also, the need for a clear, visual presentation is inherent to display data and also to show the position of the user within the system.

Some of the principles which must be incorporated into the design include [37]:

	Item	Maxim
1	Menu design for frequent actions	Use direct selection (icons + mouse, functions keys, etc.) to fix frequent actions to special buttons or function keys rather than through layers of menus.
2	Fix frequent actions and symbols early	Assign most frequent actions and symbols to mouse buttons, to function keys, or to first-level menus.
3	Fix cursor location to selected actions and symbols	predefined centers of gravity for quickly matching screen regions to single symbols or actions.

4	Reference information	Display and retain information that is likely to be referenced later. Do not cover results that are likely to be needed again with transient information.
5	Fix frequently used sequence of actions	For frequently used sequences of actions, automatically execute at the next action in sequence, and provide an exit to handle unlikely actions. Avoid asking question.
6	Fixing screens to programs	Minimize the number of screens to accomplish user actions, and group screens and programs to minimize overhead.
7	Defining screen contents	Cluster data that is needed within a short period of time in the same screen region; omit other data that is unlikely to be needed at the same time, and structure it hierarchically by likely frequency of use.
8	Match device capabilities to software uses	Use device intelligence, high resolution, color and windowing to decrease the number of user interactions and maximize perception of results.
9	Match amount of data needed to the amount displayed or entered	Use default values to minimize user interactions.
10	Focus on frequent user functions	Minimize the number of screen interactions and the processing required to accomplish frequent end-to-end user tasks.
11	Hierarchical results	Limit processing to most likely needed results; provide more by requests.

Table 1, Principles of User Interface Design

iii. KNOWLEDGEBASE

A knowledgebase is the entity which contains all the data, facts, rules and constraints which may provide a solution to a problem.

The knowledge within hypertext consists of information in the form of nodes and rules in the relation of links. To promote the purpose of hypertext as "browsable knowledge", the structure of the information and its navigational ideas are essential as there is a strong relationship between the structure of nodes and the types of links that may be used and ultimately in the presentation of material to the user. [13]

The concept of knowledgebase representation and acquisition fall in the domain of knowledge engineering and involves the manipulation of nodes, links and navigational devices. Generally, the main features of these concepts are discussed below.

A node consists of

- ♦ Name,
- ♦ Information Units,
- ♦ Features,
- ♦ Index terms, and
- ♦ Organizing Information.

The assignment of relationships between nodes is not restricted to similar ideas but may encompass other associations. [25]

Types of Links	Maxims	Example
Contrast	The meaning of one term contracts, opposes or contradict the other term.	buy-sell
Class inclusion	Terms whose denotative meaning are a part of a more broad classification.	game-chess
Similar	Terms that overlap in denotative meaning.	car-auto
Case relations	Relations involving attribution.	bark-dog
Part-whole relations	Inclusion that is pragmatic rather than necessary.	faculty-professor

Table 2, Nodal Relationships

The navigational aspects between nodes may be through the various transitions shown in Table 3.

Navigational Method	Maxim
moving by comparison	Searches in an information retrieval system.
moving by gateway or pathway	Follows a predefined path to assist the user in finding particular information.
moving by menu selection	The menu may refer to nodes directly or may refer to another menu of choices.
moving by question (direct values vs. inferred relation arguments)	Retrieval of information is by direct question.
moving by navigational link	Uninterrupted movement to directed link.

Table 3, Navigation of Nodes

a. KNOWLEDGEBASE ACQUISITION

Knowledge acquisition is the process of extracting and formalizing the knowledge of experts so that it can be included in the expert system knowledgebase. Knowledge is in the form of facts, description of objects, as well as the identification of relationships and the explanation of procedures.

The extraction of knowledge from a human specialist into the knowledgebase is a difficult process due to the heuristical nature, or rule of thumb, of the decision making process of the expert. Whereas the expert has proven his ability of applying sequence, timing and the use of combinations to maximum effect in solving specialized problems, the decision making process is often difficult to articulate and to replicate in terms of rules. Often, experts are not conscious of their decision-making processes or of the finely tuned heuristics that they apply to different problems.

The knowledge base acquisition method utilized for hypertext is a variation of the expert skill procured in expert systems and may remove some of the heuristical problems that are associated with it. The knowledgebase may be gathered from a variety of sources, ranging from human experts, to books, manuals, videos, animation.

The steps in this general knowledge engineering process include: [17][12][22][36]

- ♦ Identify knowledge,
- ♦ Elicit knowledge,
- ♦ Structure knowledge,
- ♦ Represent knowledge,
- ♦ Use knowledge, and

- ♦ Validate, refine and maintain knowledge.

The inherent nature of hypertext as nodes of information linked together diminishes the constraints of decision making rules. The "rules" of the knowledgebase come from the relational links between the nodes. Instead of translating the cognitive and heuristic rational of a human expert, logical relationships or links must be clearly defined. Such relationships are to ensure that the end user will be able to progress through the expert system by a system of rules or navigate through the hypertext through a series of links to arrive at a solution or knowledge.

The organization of knowledge within a node follows a structured guideline similar to that of nested frames. This is due to the strong relationship between the structure of nodes and the types of links that may be used. Figure 5 shows the nesting of knowledge. [22]

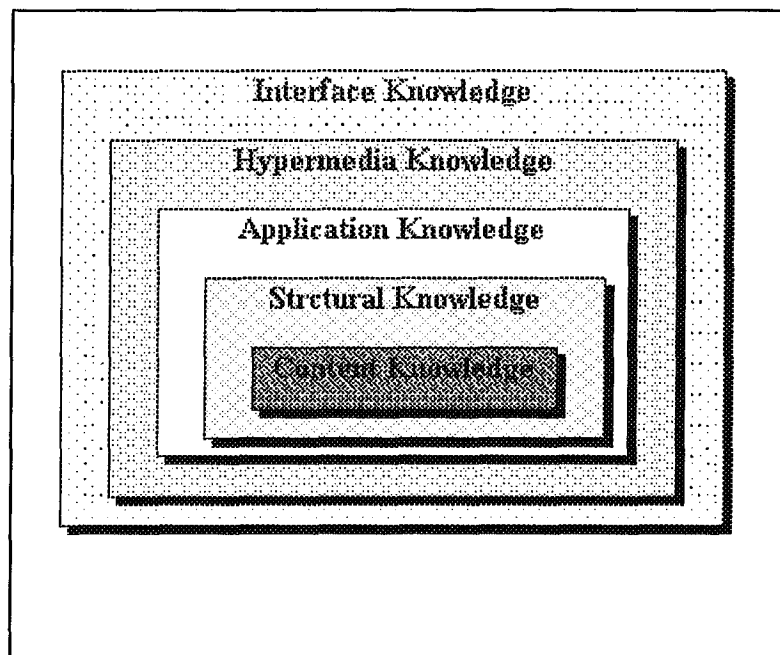


Figure 5, Framework of Knowledge

Content knowledge corresponds to the domain knowledge that serves as information resource for the hypertext application. In terms of the general knowledge engineering process, content knowledge will result from the identification and elicitation of knowledge. Organization of content knowledge consists of selecting relevant material and constructing it into nodes.

Structural knowledge refers to the links that connect the nodes that were identified in the content knowledge. The task of structural knowledge engineering is the specification of links of various types between nodes.

Application knowledge drives the usage of the hypertext. That is, it correlates the nodes and links defined from the content and structural knowledge to provide a direction in the interaction.

Hypertext knowledge is an elusive meta-knowledge that refers to the way the hypertext is structured and the information it contains. This concept is analogous to a good reference librarian who has the knowledge of how a library is organized and where things are. This concept is difficult to capture in knowledge representation and may currently only exist in the minds of skilled hypertext users.

Finally, interface knowledge consists of general information of the composition of the interface, as well as of the nodal information.

b. KNOWLEDGEBASE VERSUS DATA BASE

The knowledge-based systems, in which hypertext systems rely on, provide facilities beyond the databasing capabilities that we have become familiar with over the last thirty years. While the concept of hypertext resembles that of database management systems (*DBMSs*) with their relational links, the difference is in the intended use of the information. Where a relational database may resemble a hypertext database, the database systems usually emphasize *selection against criteria* and *reporting* rather than reading and browsing. Therefore, the efforts required to create and expand a hypertext knowledgebase with the intended purpose of perusing and learning is markedly different from that of relational databases. [14]

The difference in contents between a traditional database and that of a knowledgebase is distinct. The domain of a database consists entirely of facts and data. The domain within a knowledgebase contains both the domain facts and the associated rules governing the facts. Hence the knowledgebase is a superposition of the database and its rule structure. Data is not knowledge until it is applied.

The primary distinction between databases and knowledgebases is the fact that databases have a predetermined structure. Knowledgebanks or knowledgebases consist of the meaning of objects and the semantic relationships between objects. Knowledge is expressed by rules, facts and constraints in the form of object-oriented links. The paths by which facts are related in a knowledgebank are determined ad hoc, as needed to solve a particular problem. This characterizes a heuristic search in contrast to a structured approach.

The methodology of knowledgebanks, because of its rules, acquires a fourth dimension that is beyond that already provided in classical computing by databasing / data communications / data processing.

Knowledgebase	Database
Facts and states	Information elements subject to:
Relations between information elements	
Decision rules	Input
Methodologies	Updating
Consistency control	Retrieval
Dynamic extendibility	
Propagation actions	

Table 4, Knowledgebase versus Database

Furthermore, database inquiries imply quantitative references and precise queries. Knowledgebank queries are vague in the sense that the user has only an idea of what is required and may thus have many options of navigation. Thus queries to classical databases are crisp whereas queries in expert systems are vague with fuzzy sets, partial answers and perhaps unknown information and queries in hypertext systems are based on the logistics of the relational linkages.

CHAPTER 3

OPERATIONS

A. COMPONENTS OF SEITRON

SEITRON, synonym for SEIsmic elecTRONic handbook, is an interactive, knowledgebased design tool developed for the seismic design of multistorey steel construction. **SEITRON** incorporates the seismic provisions of Clause 27, as newly introduced in 1989, from the CAN3-S16.1-M89, with ductility parameters and template design modules in an interactive user interface for efficient and fast data retrieval.



Figure 6, SEITRON Title Page

There are three main components of **SEITRON**

- ♦ Clause 27 Knowledgebase,
- ♦ Ductility and Redundancy Requirements, and
- ♦ Design Template.

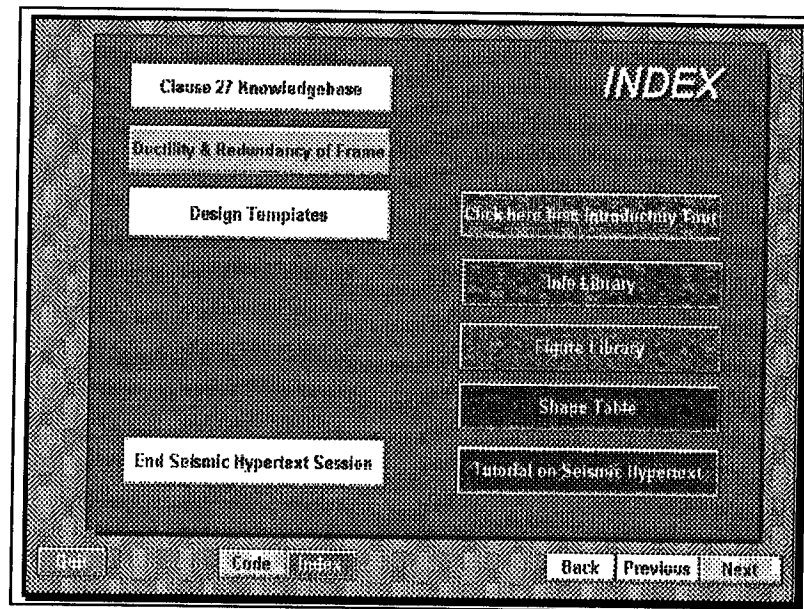


Figure 7, SEITRON Main Index

The *Clause 27 Knowledgebase* and the *Ductility and Redundancy Requirements* components acts as a hypertext which allows the user to access code information and theory of seismic steel frame design parameters. The *Design Template* component is a practical engineering design tool for the member selection of structural steel elements.

Complementing these components are sustaining knowledgebases such as the *Information Library*, *Figure Library* and *Shape Table* that clarify and support information presented from the main components of **SEITRON**.

B. CLAUSE 27 KNOWLEDGEBASE

The *SEITRON Knowledgebase* is based on the 1989 edition of the Handbook of Steel Construction, by the Canadian Institute of Steel Construction, with respect to the new seismic provisions, Clause 27. [7]

Prior to the 1989 revision, there was no code assistance in the design of steel elements with respect to seismic considerations. The National Building Code of Canada (*NBCC*) in its 1990 revision has modified the method of analysis of seismic loads acting on a structural system. The focal idea is to assign force modification factors, R , to various structural systems in relation to their capacity to absorb energy by undergoing inelastic deformations.

To be consistent with the R values assumed in the analysis, the Limit States Design of Steel Structures, CAN3-S16.1-M89, was modified to incorporate the R values into the seismic design of structures. This modification to the steel code embodied the ductility requirements into the four classes of frames. The result is the creation of a new clause, Clause 27, which attempts to standardize ductility requirements into steel frame design. Listed below are the four classes of steel frames and their governing clause numbers.

- ♦ 27.2 Ductile Moment Resisting Frames, DMF
- ♦ 27.3 Moment-Resisting Frames with Nominal Ductility, NDMF
- ♦ 27.4 Ductile Braced Frames, DBF
- ♦ 27.5 Braced Frames with Nominal Ductility, NDBF

Clause 27 relates the ductility requirements of structural components to the required strength parameters. Consequently, inspection of a clause may refer to other clauses and Appendices within the Seismic Design Requirements. The increased complexity of Clause 27 with respect to

the organization and to the cross referencing lead to the creation of an electronically referenced handbook, **SEITRON**.

The ***SEITRON Knowledgebase*** is an information network that exists in three levels of complexity and depth. The first level contains the individual clauses as stated directly from CAN3-S16.1. The second level contains the commentary with a discussion pertaining to the clauses. The third level is part of the reference or bibliography where the source of the clause is reviewed along with a brief extract or description of the author's paper.

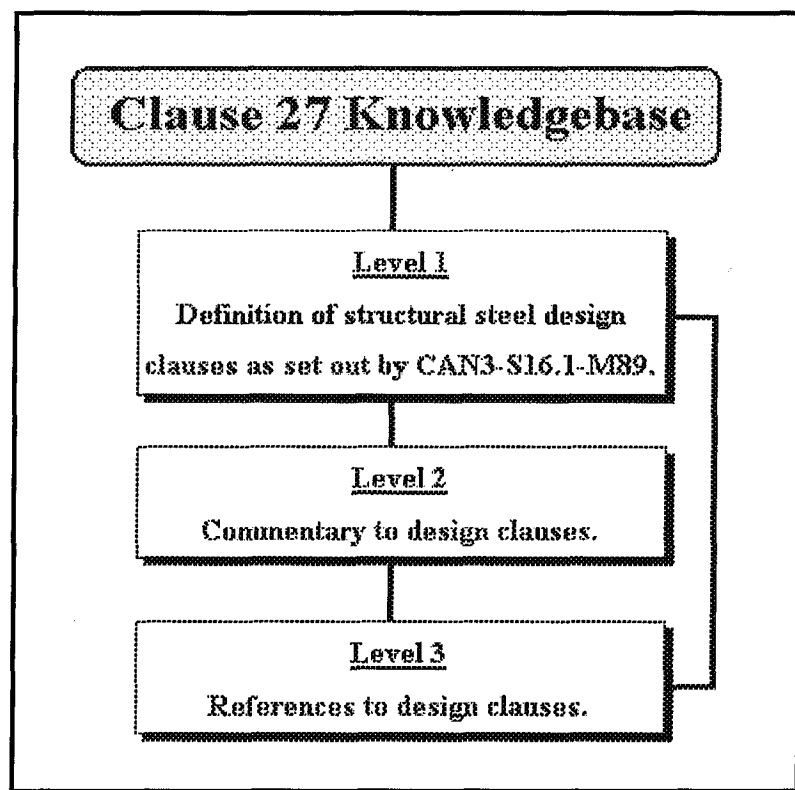


Figure 8, Structure of Clause 27 Knowledgebase

Level 1 DEFINITION OF CLAUSES

Clause 27 and its sub-clauses are located at this level and is cross-referenced between clauses on this level to that of the Commentary and the Reference levels (Level 2 & Level 3, respectively).

Level 2 COMMENTARY

Commentary regarding Clause 27 as discussed in CAN3-S16.1 is part of Level 2. Specific points as related to the clause in question or to any bibliographic development have been cross-referenced. The user may access Level 1 or Level 3 information while in Level 2.

Level 3 REFERENCES

Level 3 contains bibliographic research and development of particular clauses as background material. Bibliography of Clause 27 is cross-referenced to the corresponding clauses and to the commentary. A brief abstract has also been included in the references.

i. LEVEL 1, CLAUSES

Access to the knowledgebase is first through the clause level. When more information is required, linkage exists between the clause and its commentary or reference. Consequently, relationships also exist between the commentary and the reference itself. Thus a trilateral cohesion exists between the three levels of information.

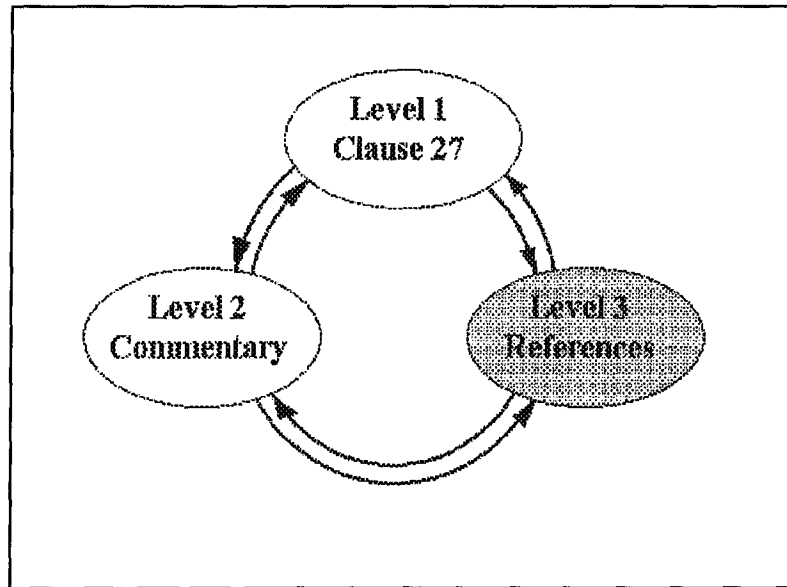


Figure 9, Clause 27 Knowledgebase Relationship between Levels

Investigation of the first level of **SEITRON** is to *Clause 27 Knowledgebase*. Organization is threefold. Access to the clauses may be through the *Table of Contents*, *Index*, or *Map*.

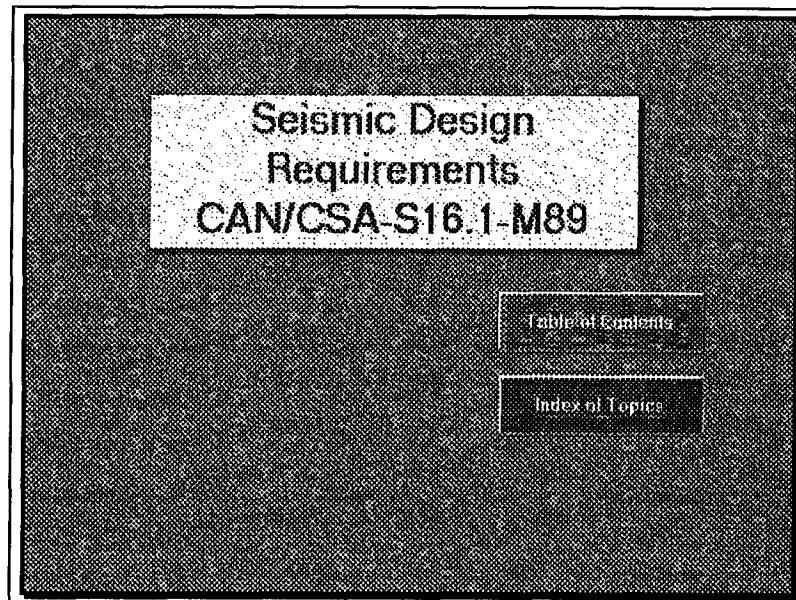


Figure 10, Acces to Clause 27 Knowledgebase via the *Table of Contents* or *Index*

a. TABLE OF CONTENTS

For users who are not familiar with the code, the *Table of Contents* will guide the user to the appropriate clause via the organization of frame systems. The division of Clause 27 is by frame systems and sub-organized by elemental ductility constraints. Elemental ductility constraints are applied to beams, columns, bracing or connection. When the *Table of Contents* is accessed, the four frame systems are listed. By clicking onto the desired frame, the sub-topics of elements guides the user to the ductility requirement in question. Depending on the frame and the sub-topic, there may be further sub-classifications to further assist the user in finding the correct clause.

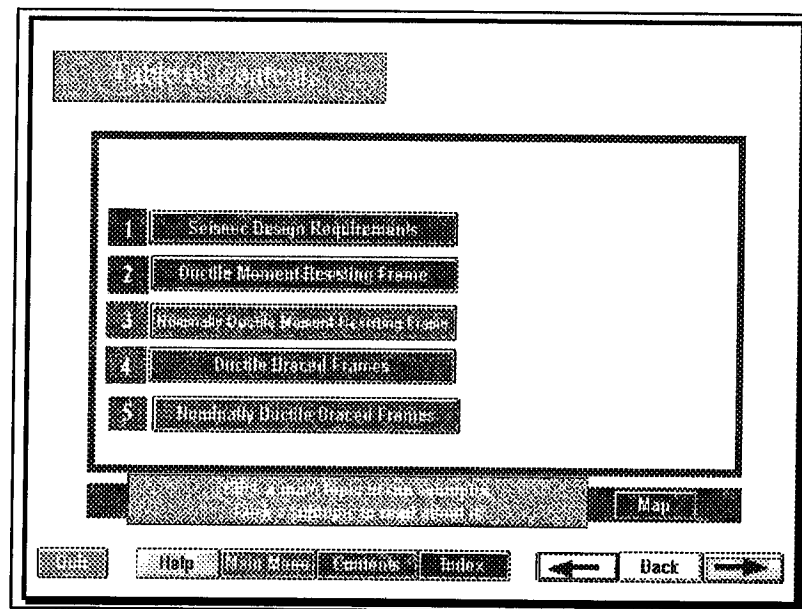


Figure 11, Table of Contents

b. INDEX OF TOPICS

Index of Topics reveals a listing of *Clause 27 Knowledgebase* in sequential order. Scrolling the list accesses all clauses as stated by the CAN3-S16.1. Also listed with the clauses are the commentary and references. Thus the *Index of Topics* shows all three levels of the knowledgebase.

The user can select a clause by scrolling down the list and clicking onto the desired topic. Similarly, the user may point to a letter in the alphabet list that is located below the scroll screen. Pointing to a letter will advance the list to the topics that start with that letter. The user simply clicks onto the topic in question to advance.

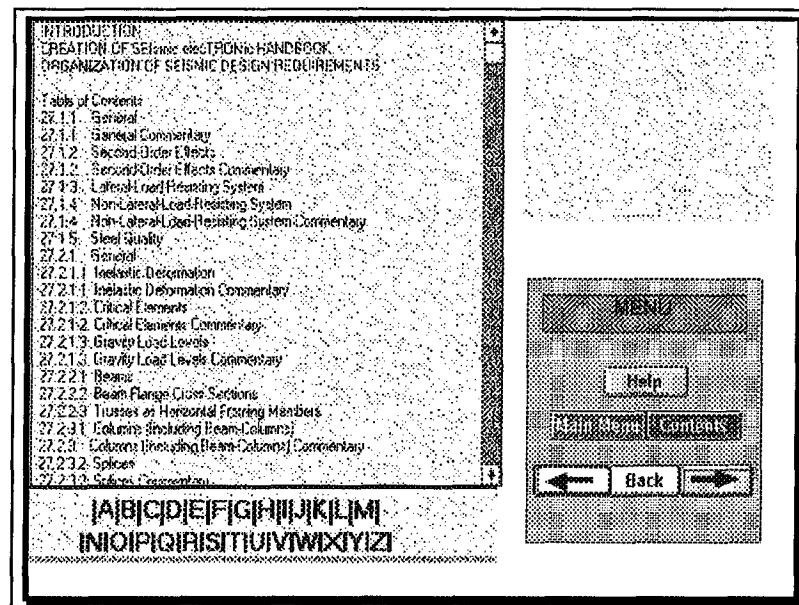


Figure 12, Index of Topics

c. MAP ORGANIZATION

It is essential that the user be aware of his position at any time during browsing or navigation in order to prevent being *lost*. A *Map* of the clauses reveals the linkage within each frame system of their elemental ductility requirements.

Clicking onto a frame system reveals the elemental components of the frames and a breakdown of sub-topics within each element. The advantage of a *Map* is that the user is able to see immediately all the code requirements for each frame system at once. The user may then determine which clause is relevant for his purpose and directly access the clause by clicking onto the topic from the *Map*.

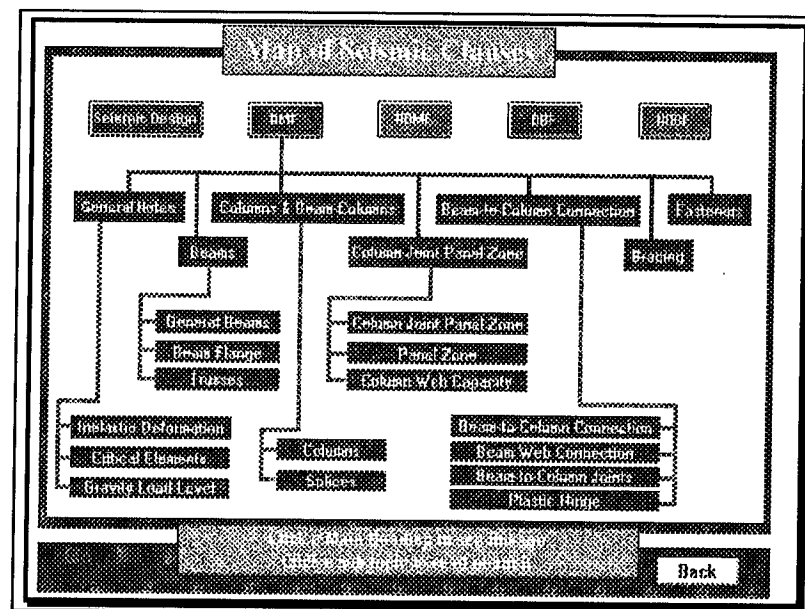


Figure 13, Map Organization

ii. LEVEL 2 AND LEVEL 3

Once the clause has been accessed, the relationship between clause, commentary and reference may be observed. While in the individual clause, linkages may be present to direct the user to common provisions in another clause or to further explain a concept through a commentary or reference link. Thus linkage may be between clauses, between clause and commentary, between clause and reference, or between commentary and reference.

The linkage between other levels is presented on the screen as buttons. Throughout the pages, guides or buttons have been implemented to provide navigational assistance by suggesting related information. These buttons and context sensitive navigational aids are discussed in Chapter 4, Supporting Infrastructure.

iii. EXAMPLE OF CLAUSE 27 KNOWLEDGEBASE

In order to illustrate the access of the three levels of Clause 27 Knowledgebase and the use of buttons as navigational tools, an example will be presented to show the steps involved in **SEITRON**. The presentation of *diagonal bracing members* from the *Ductile Braced Frame* classification will be investigated.

Access into **SEITRON** will show the title page of Figure 14.

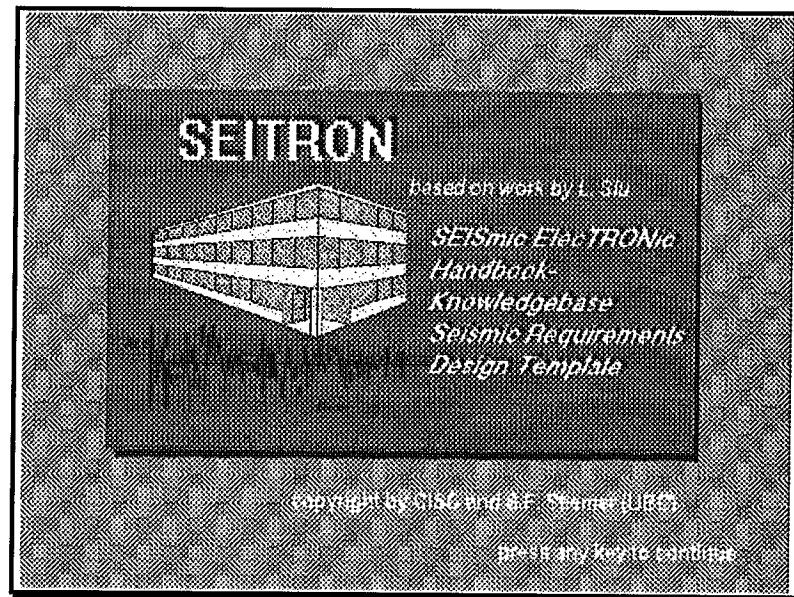


Figure 14, SEITRON title page

Clicking onto this screen will automatically present the *SEITRON Index* screen. All the topics presented here are in the form of buttons. By clicking onto any of the buttons, SEITRON directs the user to information regarding the chosen component.

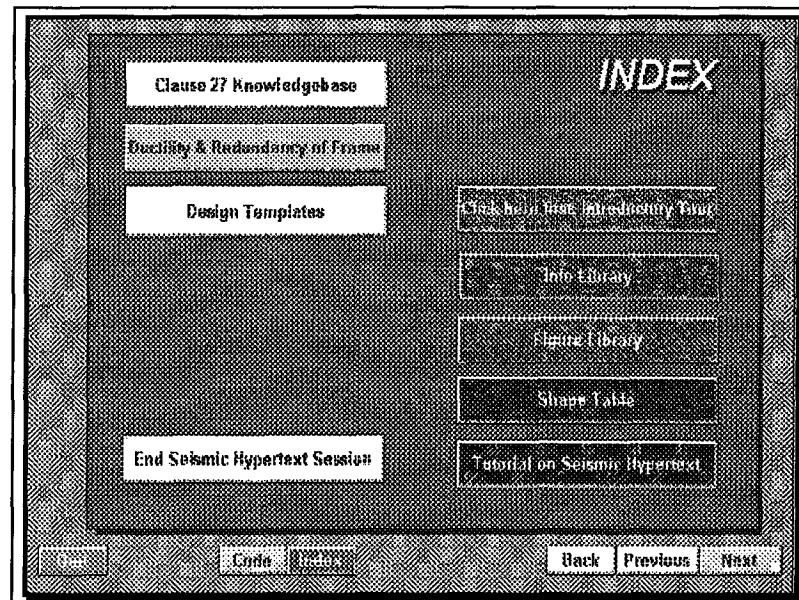


Figure 15, SEITRON Main Menu

Once the *Clause 27 Knowledgebase* button is clicked, the user is presented with the choice of information retrieval via the *Table of Contents* or through the *Index*. This example will focus on the use of the *Table of Content*.

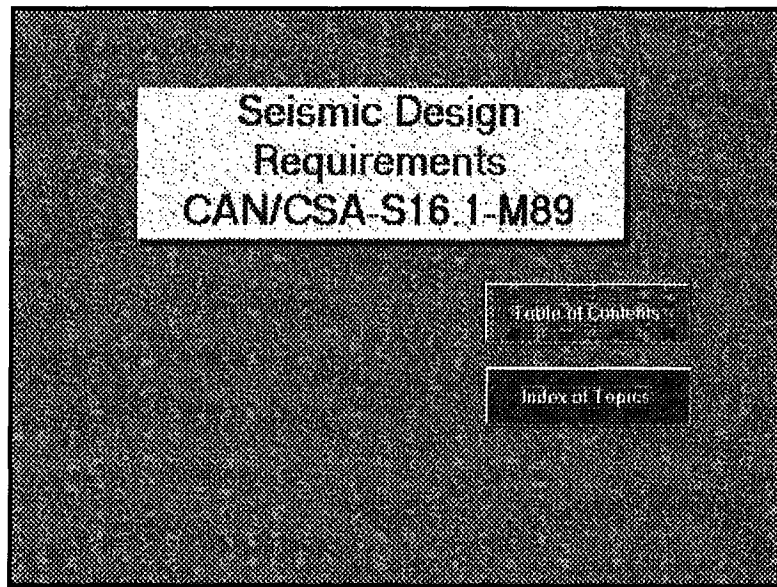


Figure 16, Access to Clause 27 Knowledgebase via the Table of Contents or Index

Clicking onto the *Ductile Braced Frame* presents a sub-menu of topics.

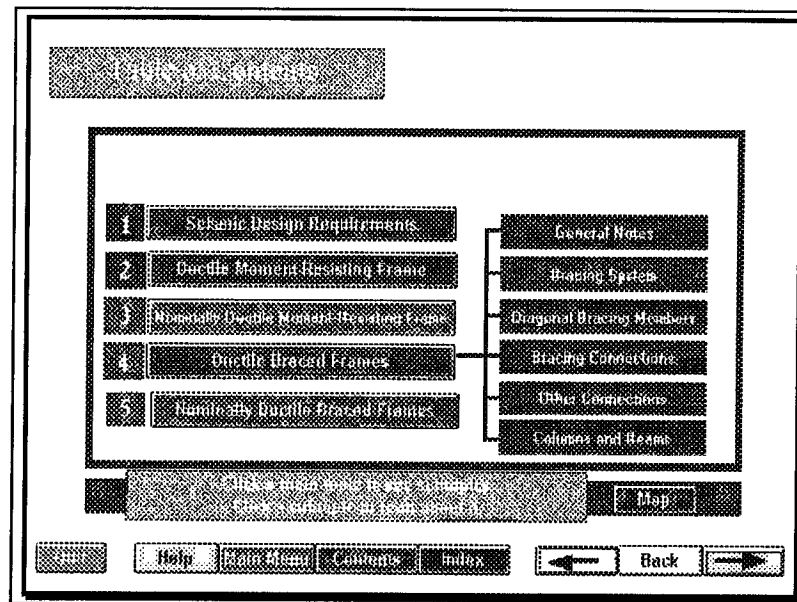


Figure 17, Table of Contents revealing Ductile Braced Frames Topics

From this sub-menu, the *Diagonal Bracing Members* button presents another sub-menu of choices. The *Diagonal Bracing Member* button is further segregated into two topics. Again clicking onto the *Diagonal Bracing Member* reveals the actual code specifications.

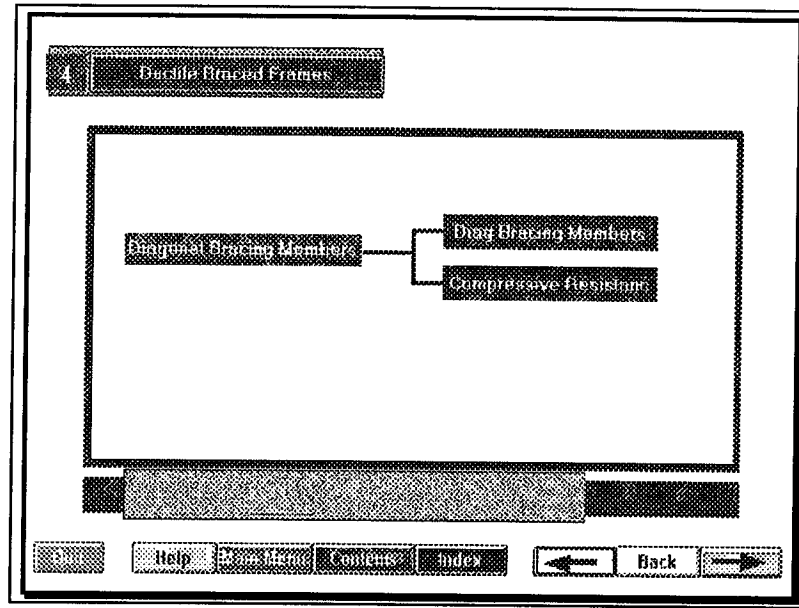


Figure 18, Investigation of Diagonal Bracing Members within Ductile Braced Frames

Note the title bar at the top left corner of the screen which reminds the user of the subject material that is being reviewed. In this example, the title pertains to the class of frames called *Ductile Braced Frames*.

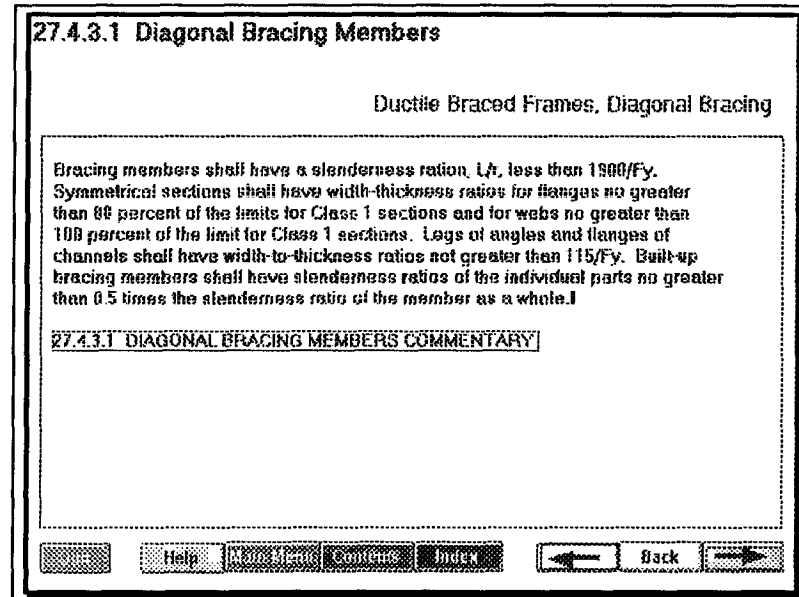


Figure 19, Diagonal Bracing Members Code Specifications for Clause 27.4.3.1

The clause in question is a direct rendition of the clause found in CAN3-S16.1.

Characteristics of the *Clause 27 Knowledgebase* screen include the main subject title with the associated clause number on the top left corner. Directly below the main title is the secondary title that references the frame classification. Information is presented directly below the titles in an information rectangle. The menu bar always appears on the bottom of the screen.

Within the information rectangle, there may be buttons which allow the user to navigate to other levels of the knowledgebase. The linkage to the *Level 2 Commentary* may be seen with the button, *27.4.3.1 Diagonal Bracing Member Commentary*. This linkage is evident to the user by

the change of appearance of the mouse cursor into a small window when he passes the mouse cursor over it. Clicking onto this button directs the user to another information screen.

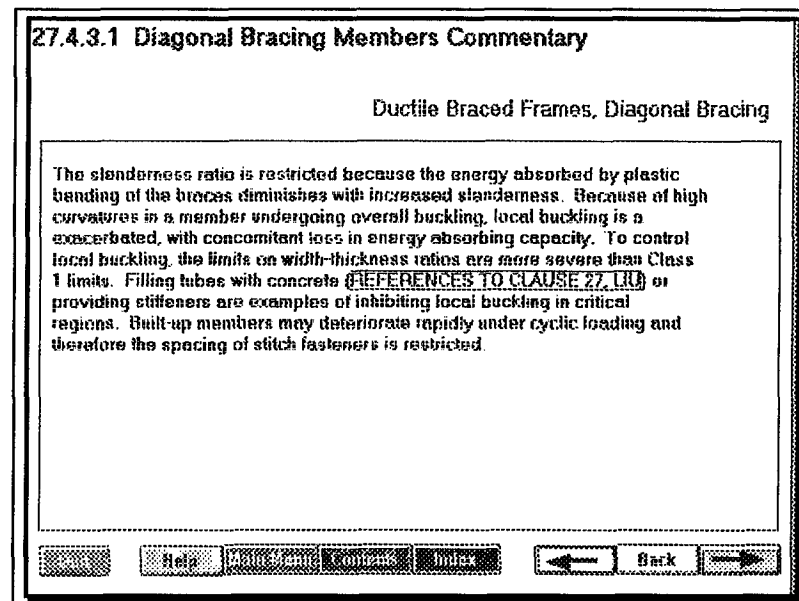


Figure 20, Diagonal Bracing Members Commentary for Clause 27.4.3.1

The main title reflects the subject and the level within the knowledgebase. In this case, we have accessed *Level 2, Commentary* and as such, the main title shows this. The sub-title continues to display the frame system of which this information screen is a part.

Access to other levels is again shown in the information rectangle in the form of buttons. The button, *Reference to Clause 27, Liu*, allows entrance to *Level 3 References*. The function of this button is apparent by its descriptive format. The first part of the button, *Reference to Clause 27*, refers to *Level 3 References* of the knowledgebase. The second part contains the name of the author/researcher who made the contribution to the clause. When more than one researcher was instrumental to the development of the clause, there are individual navigation buttons for each of the researchers such that the user has the option to investigate each author's contribution.

In the investigation of diagonal bracing members, the researcher, Dr. Liu, has provided much of the material that went into the formulation of this clause. Because of his contribution, the user may access this information by clicking onto this button.

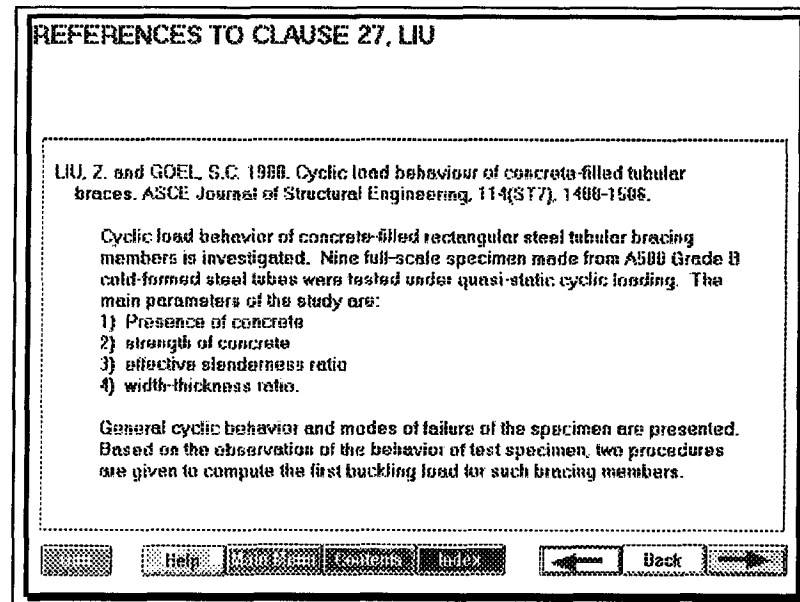


Figure 21, References to Clause 27, Liu

The main title indicates the knowledge level that is currently being investigated and may be seen to be of the same format as the *References to Clause 27, Liu* button that was previously discussed. The secondary title is left empty as the paper may refer to more than one idea and thus may affect more than one clause.

The information rectangle within the reference level of the knowledgebase contains the bibliographic reference and a short extract of the abstract from the paper that is pertinent to the development of the clause.

Finally, the progression of clauses and its hierarchical structure may be seen with the *Map* of seismic clauses. This may be accessed through a series of steps that begin with clicking onto the *Contents* button of any information screen. The *Contents* buttons navigates the user to the *Table of Contents* screen. Finally, clicking onto the *Map* button that is located just above the menu line reveals the *Map of Seismic Clauses*. In this example, information regarding *Ductile Braced Frames* was desired so by clicking onto the *DBF* button, the complete clause layout is presented.

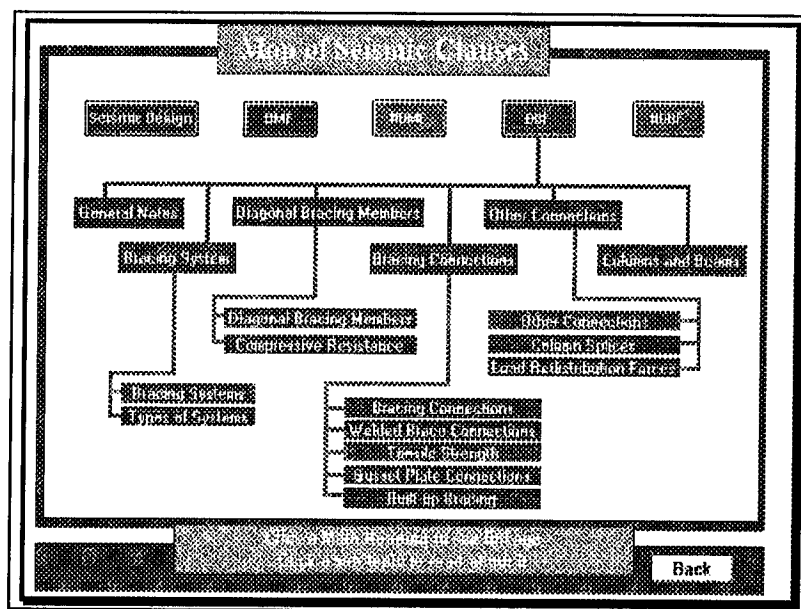


Figure 22, Map of Seismic Clauses illustrating Code Requirements to DBF's

C. DUCTILITY AND REDUNDANCY DESIGN

The **Ductility and Redundancy Design** component of **SEITRON** is a graphical representation of the ductility and redundancy requirements of frames and how the code provisions are applicable. [7][8][9][37]

Due to the complexity of Clause 27, it is often difficult for the user to apply and interpret the code. While a system of linkages relates to the individuals clauses, it does not provide full comprehension of code application. An explanation of how and when to apply the elemental ductility requirements was thus deemed necessary and a hypertext solution was the most worthy.

This second component of **SEITRON** employs the use of colors and graphics to enhance its explanation of frame ductility. Within each frame, there is a possibility of its components (beam, column, panel joint, or bracing) undergoing large plastic deformations and which are identified as critical elements. The ductility requirements are focused on the determination of critical elements. Some confusion exists in Clause 27 when it comes to determining which component is a critical element, and what constraints must be applied to both the critical element and to non-critical elements in order to ensure ductility.

Critical elements are assumed to undergo significant plastic deformation during an earthquake while "protecting" non-critical elements from begin overloaded. The latter are assumed to remain elastic.

Depending on which element is critical, certain code provisions must be met to ensure ductility. *Ductility and Redundancy Design* is arranged such that the user determines the critical element and **SEITRON** reveals which clause becomes relevant for all elements of the frame, and whether

it is critical or not. This component of **SEITRON** acts as an interpreter and teaching aid of how the seismic provisions are to be applied.

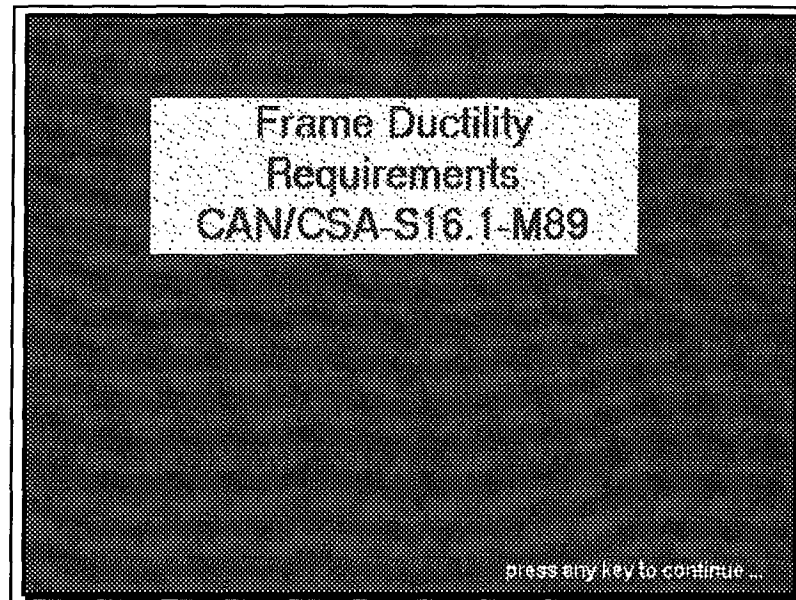


Figure 23, Frame and Ductility Requirements Title Page

When the *Ductility and Redundancy Design* component of **SEITRON** is first accessed, the *Frame Index* is shown. The *Frame Index* pertains to the four basic types of frames used in the steel industry and coincides with the code classification of the frames in Clause 27:

- ◆ Ductile Moment Frame, DMF
- ◆ Nominal Ductile Moment Frame, NDMF
- ◆ Ductile Braced Frame, DBF and
- ◆ Nominally Ductile Braced Frame, NDBF.

For each frame system, a graphical representation of its main elemental components along with its characteristics and design parameters are shown. In addition to the frame systems, two topics relating to ductility and redundancy are also available.

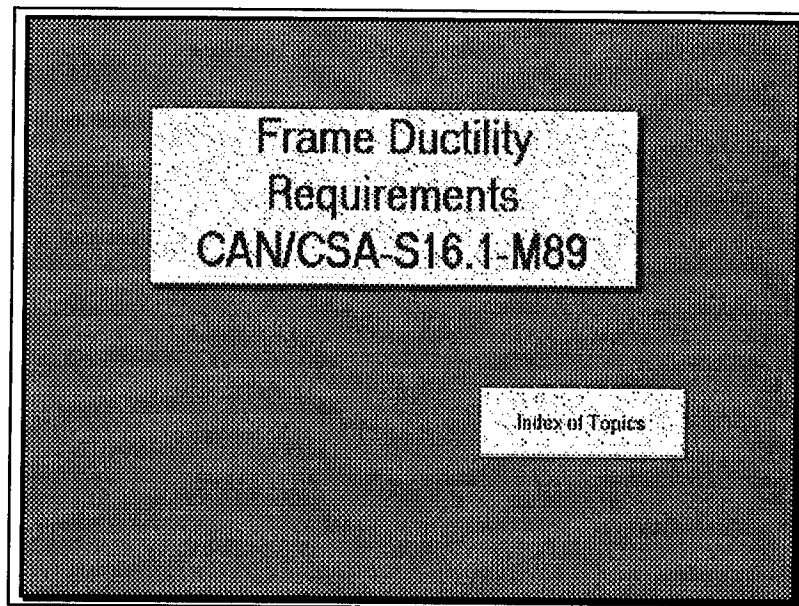


Figure 24, Access to Frame Ductility Requirements via the Index of Topics

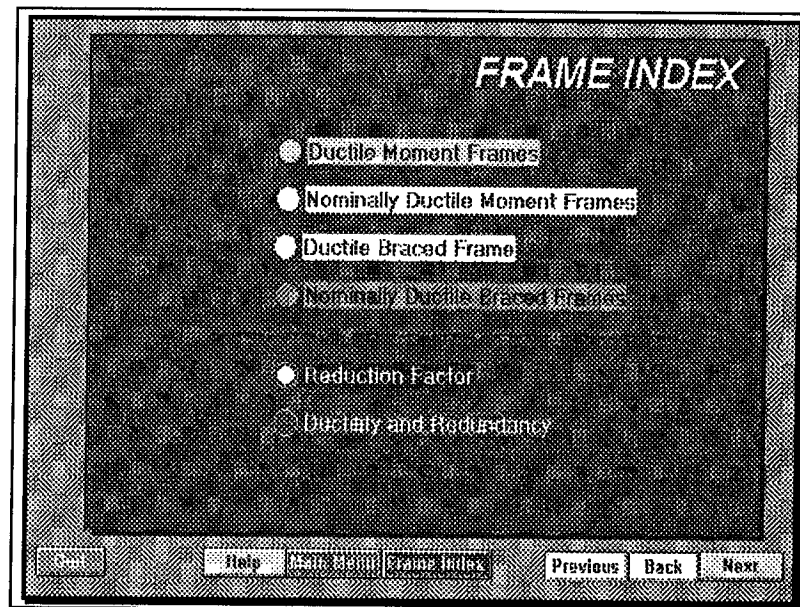


Figure 25, Frame Index

i. DUCTILE MOMENT FRAMES

In the case of *Ductile Moment Frames, (DMF)*, the critical element is the element with the potential to undergo large plastic deformations. The critical element could be either the beam, column or panel-zone. The force modification factor for *DMF*'s is 4.0 which demands the highest ductility requirements by reducing the seismic base shear.

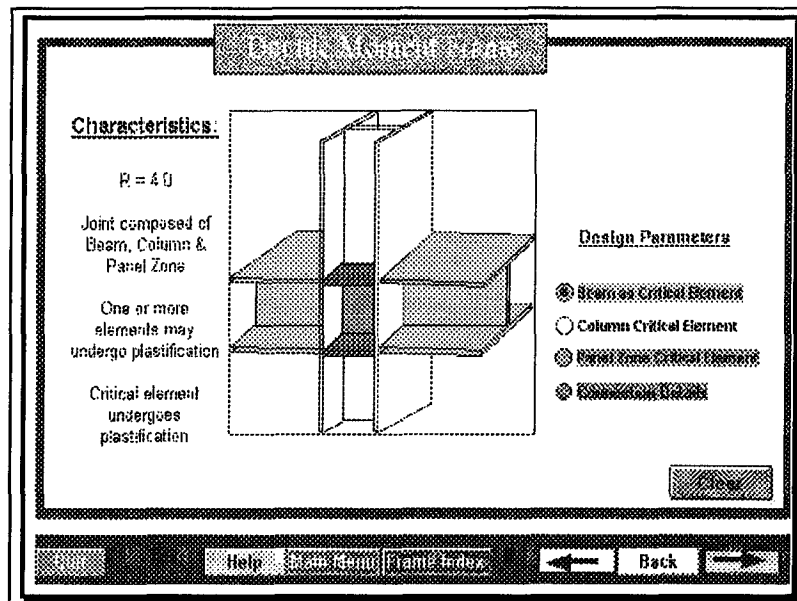


Figure 26, Ductile Moment Frame

Critical components have the capacity to form plastic hinges in beam, and, where necessary, in column or in panel zones. The critical element must be proportioned and braced in order to be able to undergo large plastic deformation. The proportioning is done through the assignment of Class 1 member sections such that redistribution of forces is possible. Non-critical elements behave elastically and are proportioned to be either Class 1 or 2 sections.

When using the *Ductility and Redundancy* component of SEITRON, the user selects the critical element by clicking onto one of the *Design Parameter* buttons. Then by passing the cursor over

the different elements comprising of the frame, the user notices the change in cursor appearance. This indicates further information may be revealed by clicking onto the various elements.

Clicking onto the elements displays a rectangular *information capsule* which gives a brief description of the design requirements and an option to review the clause in detail. This option is represented as a rectangular button with the clause number. A direct linkage has been established with the *Clause 27 Knowledgebase* for further explanations of the code requirements.

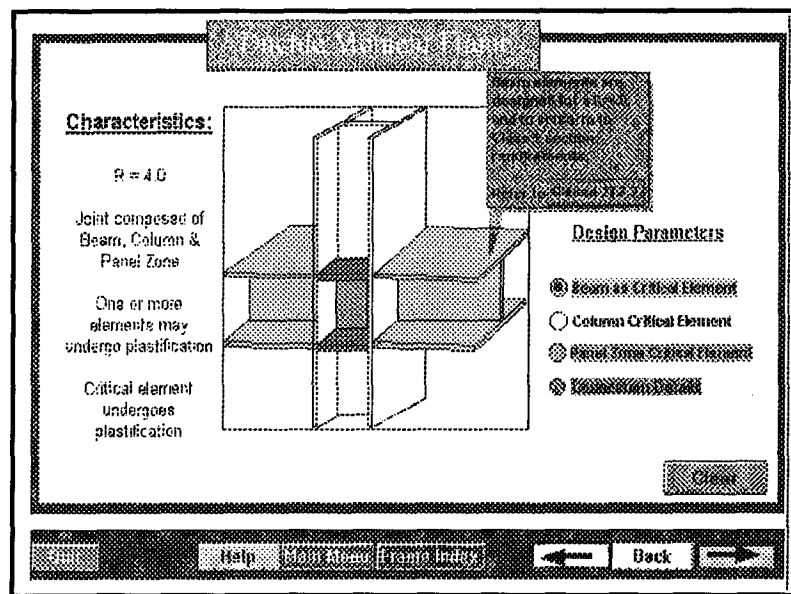


Figure 27, Ductile Moment Frame with Beam as Critical Element and Design Requirements for Beams

In this example, beams are assumed to be the critical element. Clicking onto the individual elements will show an information capsule which summarizes the design requirements based on the chosen design parameter. This system removes the confusion of when to apply which code condition. Note the rectangular button highlighting *Clause 27.2.2* that appears inside the popped-up capsule which directs the user to the desired *DMF* clause from the *Clause 27 Knowledgebase*.

ii. NOMINALLY DUCTILE MOMENT FRAME

In the case of *Nominally Ductile Moment Frames, (NDMF's)*, there is limited amount of inelastic deformation through flexural action, joint panel zone shearing or connection deformations. The critical element could be either the beam, column or panel-zone. The force modification factor for *DMF's* is 3.0 which demands the second highest ductility requirements through the reduction of the seismic base shear.

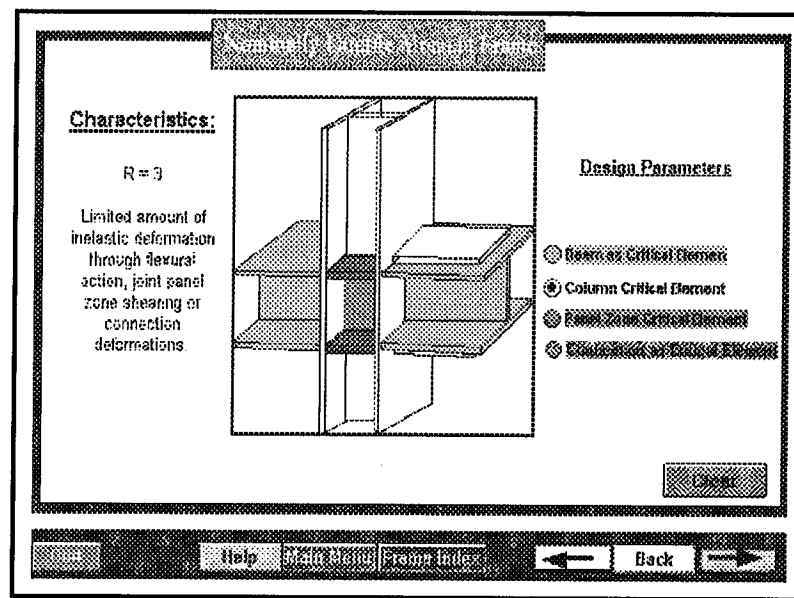


Figure 28, Nominally Ductile Moment Frame

Similar to the *DMF*, critical components have the limited capacity to form plastic hinges in beam, and where necessary in columns or in panel zones. The critical element must be proportioned and braced to attain large plastic deformation. The proportioning is done through the assignment of Class 1 or 2 sections to member sections such that limited redistribution of forces is possible.

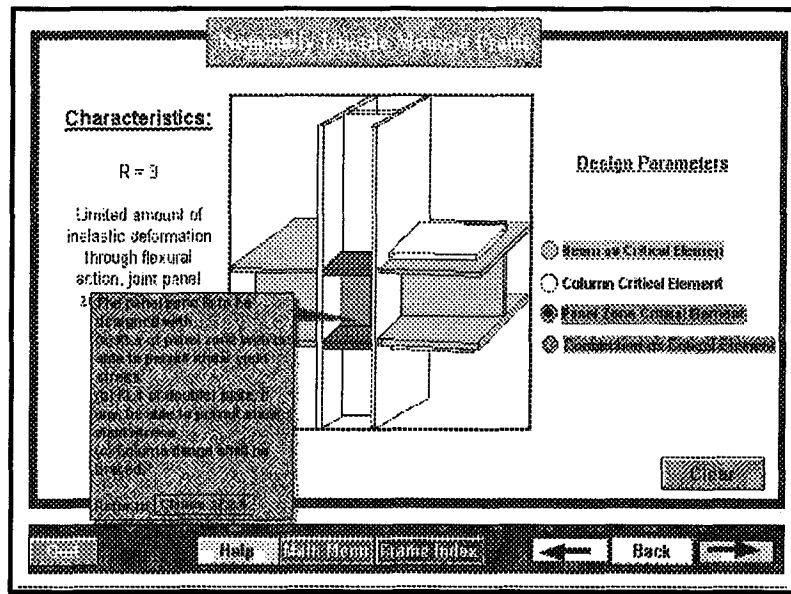


Figure 29, Nominally Ductile Moment Frame with Panel Zone as Critical Elements and Design Requirements for Panel Zone

The format for information retrieval is similar to that of the *DMF*. The user selects the critical element by clicking onto one of the *Design Parameter* buttons. In this case, the panel zone is chosen as a critical element and the button is checked. Then by passing the cursor over the different elements comprising of the frame, the user notices the change in cursor appearance. Clicking onto the elements shows a rectangular information capsule which gives a brief description of the design requirements and an option to review the clause in detail. From the example bubble above, a direct linkage has been established with *Clause 27.2.4* from the *Clause 27 Knowledgebase* for further code clarification.

iii. DUCTILE BRACED FRAMES

Ductile Braced Frames, (DBF's), follow stringent frame configurations and design criteria that do not exist for *Nominally Ductile Braced Frames*. These special configurations allow for redundancy in design and are shown on the *Ductile Braced Frames* screen. Also, braced frames do not follow the critical element criteria found for moment frames. Their energy absorption capabilities are achieved by yielding of the brace(s) and correspond to a force reduction factor of $R = 3.0$.

One of the main concepts of the braced frame system is the assurance of force redistribution from buckled or yielded braces. Because of this requirement, K, V or Chevron bracings are prohibited in *DBF* systems. Limits have been placed on the slenderness ratio and on the width-thickness ratio to control local buckling. The factored compressive resistance has been reduced to take into account the reduced load carrying capabilities under cyclic loading.

Especially stringent are the connection requirements. A reduction factor of 1.5 has been assigned to connection ductility design. This factor corresponds to the minimum ductility requirements for steel structures as specified in the NBCC90. This requirement is to ensure that the connection will always be stronger than the connecting elements and that the element will fail before the connection. Due to these ductility criteria, the code has specific clauses to ensure

- ♦ that the maximum compressive load to which the braced is subjected during initial cycles can be carried and
- ♦ that the other connections sustain the nominal braced resistance force and gravity loads

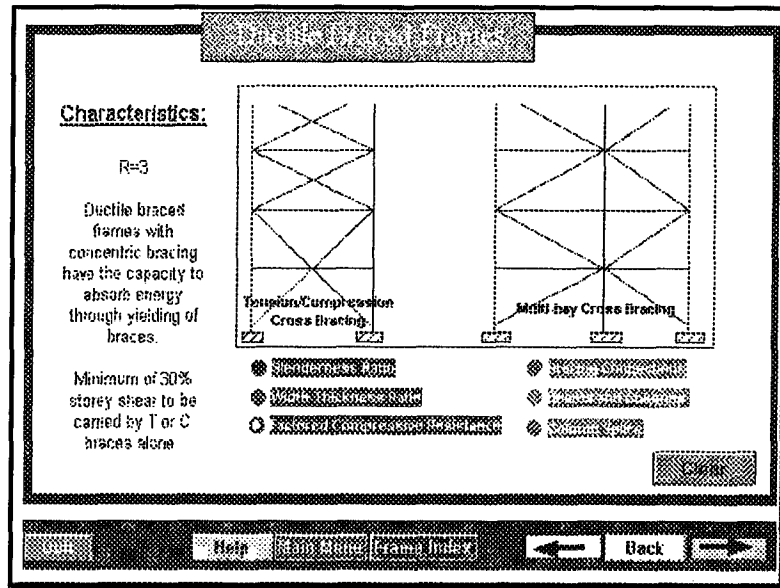


Figure 30, Ductile Braced Frames

When using the *Ductility and Redundancy Component* of SEITRON, information about each frame component is revealed by selecting one of the design parameter buttons. An information bubble appears to give a brief description of the design requirements. Each bubble contains the rectangular button which directs the user to the corresponding *Clause 27 Knowledgebase* for further code clarification. Concurrently directing arrows point to the design parameter in question.

If information regarding the *Factored Compressive Resistance* of the bracing is desired, the user simply clicks onto the corresponding button. Once the button is selected, the circle is coloured. A characteristic of this type of button, called the radio button, is that the user does not have to click on the circle to activate the button. Clicking on any of the text will activate the button properties.

Once the parameter button is activated, an information bubble appears with the corresponding clause button which again links the user to *Clause 27 Knowledgebase* and arrows point to the relevant element of the frame system.

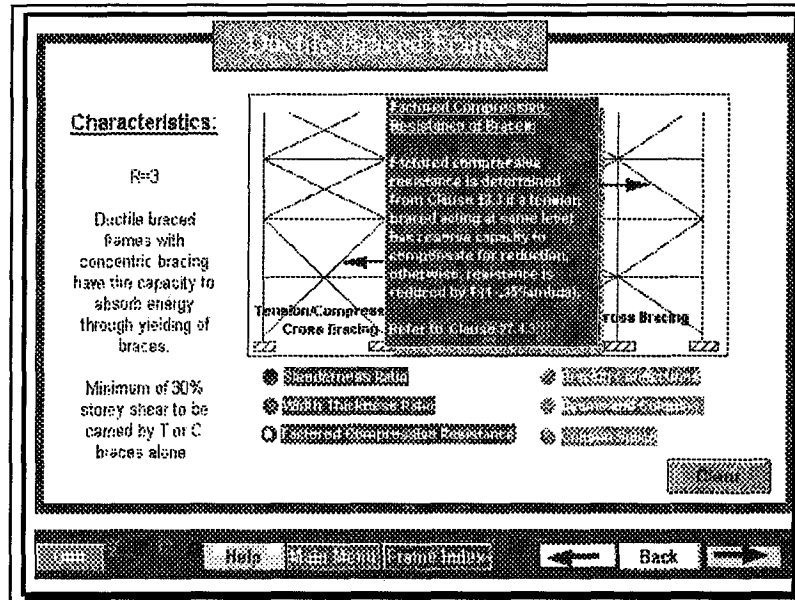


Figure 31, Ductile Braced Frames with Design Requirements for Factored Compressive Resistance

iv. NOMINALLY DUCTILE BRACED FRAMES

The *Nominally Ductile Braced Frames (NDBF)* screen shows suitable frame configurations and design criteria. Nominally braced frames do not follow the *critical element criteria* found in the moment frames. Their energy absorption capabilities lie in the limited amount of energy absorption through inelastic bending or extension of the braces which results in a force reduction factor of $R = 2.0$. Of the four basic frame systems, *NDBF*'s have the lowest reduction factor and thus the lowest ductility requirements.

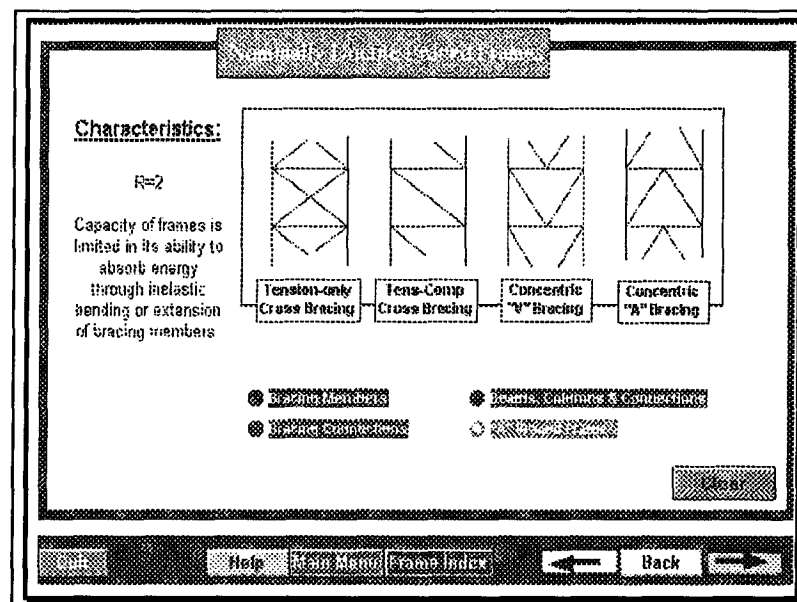


Figure 32, Nominally Ductile Braced Frames

Since yielding is not inherent in *NDBF*, the criteria for frame systems is not as stringent as that for *DBF*'s. Chevron, K, or V bracing are allowed. However, bracing members must conform to Class 2 section criteria to control local buckling.

Connection requirements for the *NDBF* are also assigned a force reduction factor of $R = 1.5$. This factor corresponds to the minimum ductility requirements for steel structures as specified in

the NBCC90. This requirement is to ensure that the connection will always be stronger than the connecting elements and that the element will fail before the connection. Due to these ductility criteria the bracing connection is designed for comparably high loads resulting from the low force reduction factor of $R = 1.5$. Also, the specific clauses are directed towards the use of A and V bracing to ensure that the beam can support the gravity load if the brace buckles.

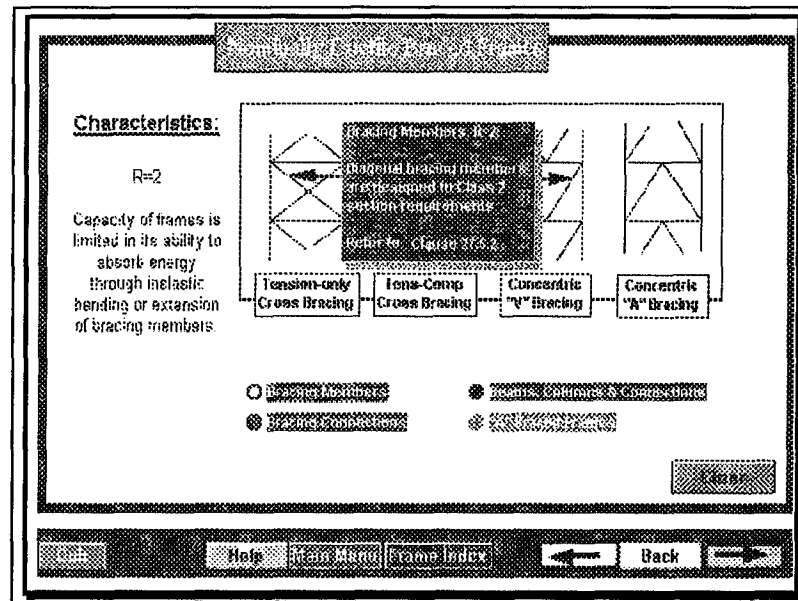


Figure 33, Nominally Ductile Braced Frames with Design Requirements for Bracing Member

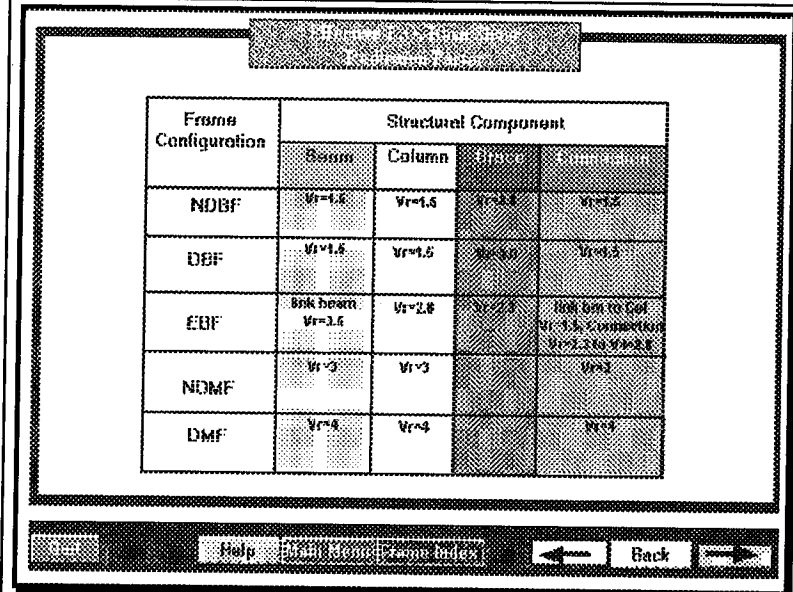
As with the *DBF* screen, information about each *NDBF* frame component is revealed by selecting one of the design parameter buttons. An information bubble appears to give a brief description of the design requirements. Each bubble contains the rectangular button which directs the user to the corresponding *Clause 27 Knowledgebase* for further code clarification, while arrows point to the design parameter in question.

In the example presented, the *Bracing Members* button was selected. The information bubble has arrows pointing directly to the bracing elements on the system and the rectangular button highlights access to *Clause 27.5.2*.

V. REDUCTION FACTORS

In addition to the frame ductility requirements, a summary table of reduction factors, (*R factors*), as it applies to each frame and to each component, is presented. Clicking onto any of the modification factors links the user directly to the clause and its supporting knowledgebase.

The ductility of the frame is governed by the maximum *R* value of the frame. It is worth observing that in the case of the braced frames, the *R* factor for the connection is lower than the *R* factor for the frame. This is due to the fact that the connection must always be stronger than the element to ensure no sudden failure. The connection ductility requirement of $R = 1.5$ is the same requirement if no ductility was applied and that the connection was to behave elastically.



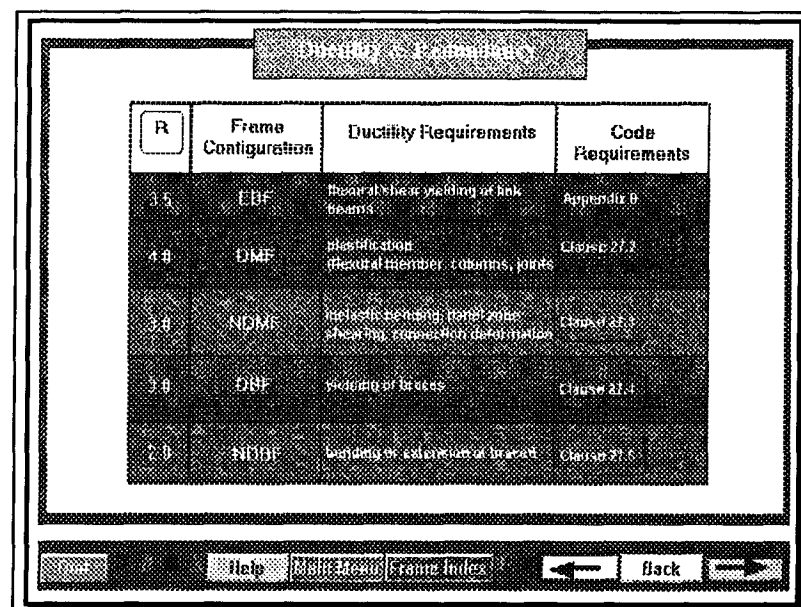
Frame Configuration	Structural Component			
	Beam	Column	Brace	Connection
NDBF	$R=1.5$	$R=1.5$	$R=1.5$	$R=1.5$
DBF	$R=1.5$	$R=1.5$	$R=3.0$	$R=1.5$
EBF	Brk beam $R=3.5$	$R=2.8$	$R=2.8$	Brk brk to Col $R=1.5$ Connection $R=1.5$ to $R=2.8$
NDMF	$R=3$	$R=3$		$R=3$
DMF	$R=4$	$R=4$		$R=4$

Figure 34, Effective Earthquake Base Shear Reduction Factors

vi. DUCTILITY REQUIREMENTS

A summary table of R factors for each frame along with a brief description of the frame's ductility requirements is shown. One accesses the *Clause 27 Knowledgebase* by clicking onto the *R* heading or clicking onto one of the topics below the *Code Requirements* column. These links are highlighted by rectangular buttons with rounded corners.

For example, if the user is constructing a building using nominally ductile moment frames, he may access the *Ductility and Redundancy* screen for the ideas regarding the main theoretical ductility constraints for design purposes. Looking at the table, *NDMF*'s have a R factor of 3.0 which the user must apply to reduce his base shear. Observing the *ductility requirements* for the *NDMF* consists of inelastic bending and panel zone shearing, he would make special consideration in his analysis and design procedure so that this kind of energy absorption is utilized. Finally, when specific code constraints are required, he clicks onto the *Clause 27.3* button to access CAN3-S16.1-M89 design specifications.



The screenshot shows a software interface titled "Ductility and Redundancy". It contains a table with four columns: R, Frame Configuration, Ductility Requirements, and Code Requirements. The table lists five frame types: EBF, DMF, NDMF, DBF, and NDMF. Each row provides specific ductility requirements and links to code requirements. At the bottom of the interface, there are navigation buttons: Home, Help, Ductility and Redundancy, Frame Index, and Back.

R	Frame Configuration	Ductility Requirements	Code Requirements
3.5	EBF	theoretical base yielding of link beams	Appendix B
4.0	DMF	plastication flexural member columns, joints	Clause 27.2
3.0	NDMF	inelastic bending panel zone shearing, connection deformation	Clause 27.3
3.0	DBF	yielding of braces	Clause 27.4
2.0	NDMF	bending or extension of brace	Clause 27.5

Figure 35, Ductility and Redundancy

When the *R* button is accessed, the *Frame Configuration and Component Design Force Comparisons* graph is shown. This diagram represents the seismic base shear force reduction as it applies to the four basic types of steels frame and are shown relative to each other.

The use of graphs and diagrams is illustrative in presenting concepts. This diagram as well as others are part of the *Figure Library*.

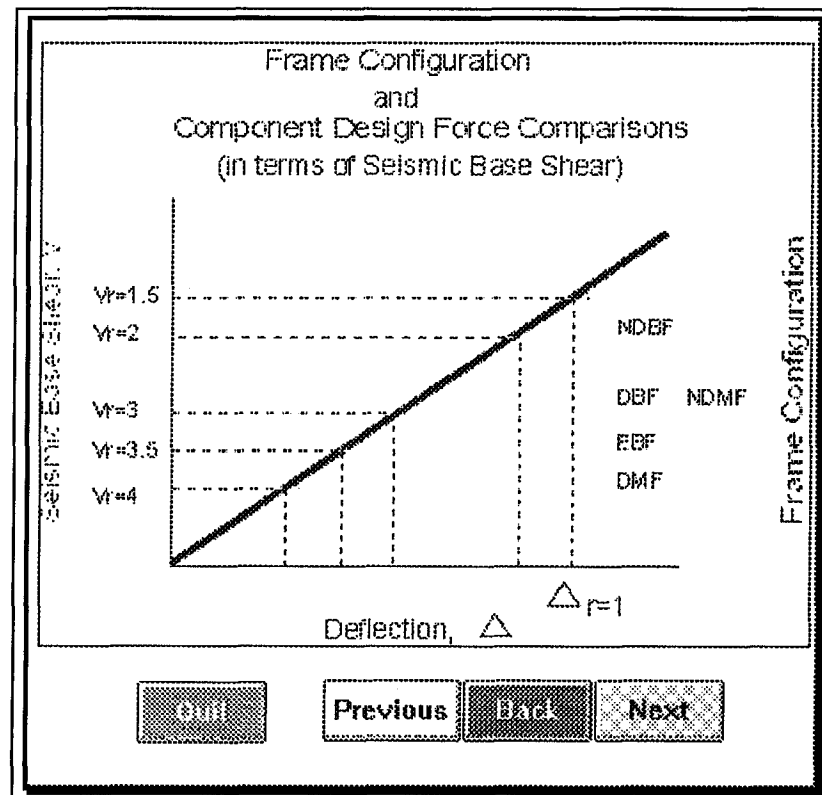


Figure 36, Frame Configuration and Component Design Force Comparison

D. SPREADSHEET DESIGN COMPONENT

i. GENERAL

In as much that slide rules were a part of the engineer's tools during 1930-70's, so is that of the spreadsheet for the 1980's engineer. What the spreadsheet entails is the flexibility and ease of calculation in a "visual" environment. Spreadsheets ease the traditional problem solving technique of programming. The user is able to see the calculations and thus troubleshooting or "debugging" is relatively easy.

Spreadsheets are adequate for data manipulation due to their inherent tabular form. However, to use this tool to its capacity involves the manipulation of range names for variable declaration and the assertion of formulae in symbolic form, much like that of traditional programming. The major diversion between these spreadsheets and traditional programming is in the presentation of inputs, calculations and output. The variables and formulae are always visual with spreadsheets, whereas in programming, they are hidden within the code. When coupled with the use of macros or programmed keystrokes for the operation of repetitious tasks or for the generation of menus, the spreadsheet may be more powerful than programming.

The use of spreadsheets as a structural engineering design tool has seen a few changes. The first generation method was coined *formatted* spreadsheet. Advances to multiple sheet technology along with **GUI** and **WINDOWS** technology have produced the second generation method which is termed *structured*. Finally, the most recent generation is that of the *batch* mode where the use of databases coupled with the structured spreadsheet and macro programming may automatically and optimally choose and design member sizes.

ii. FORMATTED SPREADSHEETS

One advancement of the spreadsheet environment is through the use of *formatted spreadsheets*. *Formatted spreadsheets* make use of macro programming to create single keystroke macros to perform repetitious tasks. The *formatted* term arises from the use of regulated column widths to change the computing environment to a working environment similar to that of traditional hand calculations and having the appearance much like that of an estimator's take-off sheet.

The replication of the design process in electronic form follows the same steps as the solution process in paper form. The engineer models the physical problem in a mathematical capacity through a succession of variables and formulae. These formulae would initially be written in variable form and numerical values attached later on. Using the regulated column widths of the formatted spreadsheet, the engineer uses each line as a variable or formulae declaration. Through the use of macros which incorporate the variable or formulae declaration into a single keystroke, the engineer may easily and quickly perform calculations.

In terms of structural engineering, one of the most time consuming activities is in the code verification of members. Standard hand and calculator operations would involve the calculation of the resistance of the member according to code specifications. This resistance value is checked with the applied loading and if the applied forces are greater than the resistance, a new member is chosen and its resistance is recalculated. This procedure would proceed until a suitable member is chosen to meet the applied forces and all the code constraints. More time and paper would be used in this repetitive process until all members of the structure are verified.

The iterative procedure of checking calculated resistance with the applied forces is easily performed in the electronic case through the simple change of value of the variables. Through the use of range names in the assignment of variables, a change in value of any variable will

automatically change the value of formulae which utilizes this variable. The result of these variable changes as they pertain to the output, or resistance, is immediate. In the case of the traditional paper calculation, iterations means that each formula must be performed again, utilizing more paper and more time. Thus the advantages to the *formatted spreadsheet* format is in the ability for the user to visually see and check the calculations and to quickly perform iterations.

iii. STRUCTURED SPREADSHEET

With the event of **WINDOWS**, the spreadsheet technology has advanced further in its increased potential of data manipulation and ease of use. The use of object manipulation and icon commands greatly reduces the need for the user to remember keystrokes as was once the case with traditional spreadsheets. The creation of multiple, stacked spreadsheets greatly enhances the concept of a finite work space. Finally, the use of graphics, fonts, and appearance manager effectively displays the results of the data.

In addition to the improvements in spreadsheets that were brought about with **WINDOWS** technology, the power of the *structured spreadsheet* comes from the utilization of data tables to provide auxiliary information. The manipulation of multiple files in conjunction with data base files allows *three dimensional data analysis and processing* which diverges into the realm of knowledge base design.

In the second generation of spreadsheet overlay development, multiple-sheet spreadsheets increase organization strength through the arrangement of information onto separate sheets instead of through the traditional staggered layout. Looking at i.e. **LOTUS 123W** spreadsheet application, a total of 256 separate sheets may be manipulated in a single file. As well, each sheet is the same size as the spreadsheets used in single sheet technology. In single spreadsheet

technology, blocks of distinct information or data must be arranged in a staggered manner such that when additions or deletions of columns or rows become necessary, other data is not unknowingly removed.

The three-dimensional arrangement of data or information is analogous to the stacking of paper in a file folder. However, the three-dimensionality comes into play with the knowledge that individual sheets may be inter-related to other sheets within the file. Computation is not restricted within each sheet but may encompass several sheets at a time. So instead of the single spreadsheet's two dimensional characteristics of rows and columns, the *structured spreadsheet* attains the third dimension of depth with the stacking of sheets.

An example of this arrangement may be such that calculations are on Sheet A, database on sheet B, data table on Sheet C, macros on sheet D and so on. The manipulation of calculations is not restricted to a single sheet but may encompass several sheets. Such a calculation would be in the usage of data table values from Sheet C in the resistance calculations on Sheet A. The effect is working with more than one file at one time and knowing where information is located.

The development of *formatted and structured spreadsheets*, with respect to structural steel design, at the University of British Columbia has led to the creation of steel design templates employing steel shape table databases. For elemental design templates, member characteristics must be input into the calculations. Member characteristics may be obtained from the Handbook of Steel Construction, (*HSC*), shape tables or they may be calculated for specialized built-up sections. However, instead of manually inputting the values and manually changing these values when new members are selected during iterative cycles, it becomes more efficient to create macros and sub-menus to allow the engineer to select desired members and have the computer automatically update the member properties from a data base.

Database files have been created for each of the structural shapes as listed in the HSC. Each line of a shape file contains all the pertinent sectional properties for a particular size. Such information includes the area, cross-sectional moment of inertia, section modulus, depth, width, thickness and other sectional properties. It is worth noting that all the information of a particular size of member is entered as a single cell of data. This data string is later broken down to different components. The purpose of a single string character versus many strings is for the ease of data manipulation.

When an element is chosen, the macro copies the database string to the *shape table* part of the template. The string is then delineated into its sectional properties and transferred to the input cells. When a new element is desired, the process is repeated.

1. Choose element shape
2. Macro combines database file of shape in question with template
3. Choose element size
4. Macro copies and delineates element string in *shape table*
5. Sectional properties added to input cells.

The entire process is fully automated and thus eliminates the user from manually finding the section related parameter values in the HSC and manually typing the data. Another benefit of this system is that engineers do not have to "blindly" guestimate the choice of element. For common design purposes, a W-section database exists that orders elements by moment resistance. Thus by desiring a certain moment, an "educated guess" selection of sectional size may be made.

iv. BATCH MODE

The design of multistorey structures can involve selecting and checking sizes of a large number of elements. It can become very tedious to manually select and verify strength and stability considerations for each element. This is usually the case when the member forces in a building have been determined in a structural analysis program. While many commercial software packages are available for the finite, static or dynamic analysis of structures, a particular tool is missing which adequately selects a member and verifies these elements according to the code conditions.

The *Batch Mode Design* involves the automatic computer selection or verification of members to meet strength and stability criteria. The user provides a list of elements with their corresponding geometrical properties, such as length, and applied forces. This information is usually the output from a structural analysis program. Through the use of macros, the computer goes through a pre-selected member database against the applied forces and checks for its adequacy. If the member in question is too small, the next larger size is checked. The process is repeated until all the strength and stability conditions are met.

Very often, however, especially in multistorey design, elements are not optimized for resistance. The wide range in forces and geometrical consideration of elements would result in a large variety of member shapes and sizes if the optimization to strength and stability was the governing criteria. This requirement does not lend itself to economy, member availability, connection design, or practicality. For such design conditions, a practical solution would entail the choice of a general type of shape and a solution would involve the user perusing through a condensed list of sizes within that shape. For example, in a multistorey building one would want to standardize its column size by using only W250 shapes and as such a restricted database of all W250 sizes would be utilized.

This batch procedure can save many hours of tedious design work which becomes inherent in large elemental structures such as multistorey buildings, domes and telescope enclosures.

v. SPREADSHEET ADVANTAGES

One of the major advantages of spreadsheet calculations is the ease of formulae changes. Unlike traditional programming where the formulation is hidden within the uncompiled code, the calculations in a spreadsheet are transparent and may be easily modified. This ability is quite important as revisions to the HSC and possibly some of the code design formulae are expected every four years. The interactivity of the spreadsheet eliminates the worry of input and output files as modifications to variables are readily seen.

The main advantages of using a spreadsheet environment for the design calculations are

- ♦ layout of the calculations is similar to traditional hand calculations,
- ♦ presentation of inputs, outputs, and calculations is clear, and
- ♦ ease of formulation updates in the event of code changes.

E. DESIGN TEMPLATE COMPONENT

i. GENERAL

The rationale of the *Design Template* component was to create specialized templates for structural design purposes which may be used repetitively, similar to a traditional computer program. These design templates usually encompass calculations where a steel element is checked for strength and stability parameters against the constraints of the applied forces. These design parameters and member characteristics are dictated by the HSC.

All the design templates are based on the latest code provisions of CAN3-S16.1-M89. In addition to its advantages as a design tool, the strength of the *Design Template* component lies in the ability to access *Clause 27 Knowledgebase* for explanation of the design principles.

The *Design Template* component of SEITRON utilizes the SEITRON interface as the input and output while a spreadsheet application in the background performs the calculations. The current spreadsheet application used with SEITRON is LOTUS 123W. Templates have been created for various design purposes, such as beams, beam-columns and tension members, which SEITRON automatically activates according to the design needs of the user. The user only sees the input and output screen and does not concern himself with the calculations.

The benefit of this system eliminates the need for the user to be familiar with spreadsheets while at the same time the author takes advantage of the spreadsheet as a "visual" programming tool. Thus for each design screen, the user simply inputs the elemental characteristics and applied forces and chooses the desired member size. By activating the *CALC* button at the menu line, SEITRON, through a system of dynamic data exchange and dynamic link libraries, transfers the inputs given by the user to the desired template and then transfers the output back to the interface.

(Dynamic data exchange and dynamic link libraries will be discussed later.) The user does not have to concern himself with formulae as all calculations are performed out of the user's sight in the background.

However, the spreadsheet system linked with a user interface is even more beneficial to users with some spreadsheet background. This is especially true for engineers who often request code modifications. When certain clauses or formulae need to be modified, the spreadsheet may be easily revised. This is due to the fact that the calculation module of **SEITRON** is independent of the interface. If traditional programming were to be used, the calculation programming code would have to be modified and then recompiled. Moreover, if the programming code is part of the interface code then any code changes would be vulnerable to errors. The user would have to be fluent in the programming language as well as in the architectural structure of the program. On the other hand, if formulae changes are needed but the user does not have the expertise to modify the calculation module, the program might become useless and obsolete.

The templates used in this component are a modification of the *structured spreadsheets* previously discussed. The difference between the *structured spreadsheets* and the *design templates* is in the capacity of the database. With *structured spreadsheets*, the database is contained as a spreadsheet file for the ease of combining the database to the design templates. Such spreadsheets act as stand alone programs that may be activated with macros.

The shape database for use within the *Design Template Component* is currently organized in the same configuration as **SEITRON**. That is, each member and its associated geometrical properties occupy one page of the *Shape Table Book*.

Although shape databases exist in spreadsheet files for macro activation within the spreadsheet mode, DLL technology has not advanced far enough to be able to activate macros while in the

SEITRON interface. Current technology dictates that a more efficient method of storage would be to create external database files within an application program such as **DBase®**. Data transfer and retrieval between the database, spreadsheet and interface would be through a system of **DDE's** and **DLL's** much like that used for input/output retrieval between the spreadsheet and **SEITRON** interface.

ii. INTERFACE AND COMMUNICATION

The interface in the *Design Template* serves two purposes: a means of presenting engaging data with the use of graphics and presentation managers and a coordination of the inputs and outputs required for the design. The advantage of the interface is in the ease of use. The user simply inputs the elemental properties, applied forces and desired member, and **SEITRON** summarizes the resistances with the calculations performed out of sight of the user.

The **SEITRON** interface is based on **WINDOWS®** technology. To utilize the data transfer capabilities from this technology, the spreadsheet must also be a **WINDOWS** application. Applications such as **LOTUS 123W®**, **EXCEL®**, and **QUATTRO PRO®** are suitable spreadsheets. Communication files between each of these spreadsheets and with **SEITRON** are slightly different. Templates used within this thesis make use of **LOTUS 123W** spreadsheets.

Communication between **WINDOWS** applications relies on *Dynamic Data Exchange, (DDE)*. The *DDE* protocol defines a strict set of rules that allows an application to send commands to, or get information from another application. The use of the *DDE* allows the processing of several sets of data from several applications which may be running simultaneously. In the *Design Template* component, **SEITRON** and **123W** are both running simultaneously, with the former in the foreground and the latter in the background.

In conjunction with the *DDE*, *dynamic link libraries*, (*DLL's*) are also employed. *DLL's* are separate files that contain functions which **WINDOWS** applications can use to share code and resources.

All communication of input and output data between **123W** and **SEITRON** are performed through *DLL* files. Creation of *DLL* files requires experience in **WINDOWS** programming. A copy of a programming language such as **Microsoft C** or **Pascal**, along with an assembler is required to write the code. With respect to the *Design Template* component, control of the inputs is from the *DLL* file which transfers input values from **SEITRON** to **123W** and then transfers the output from **123W** back to the interface.

iii. OVERVIEW OF DESIGN TEMPLATE MODULES

Organization of the *Design Template* component is broken down into three main modules:

- ♦ General Design,
- ♦ Element Design, and
- ♦ Frame and Composite Design.

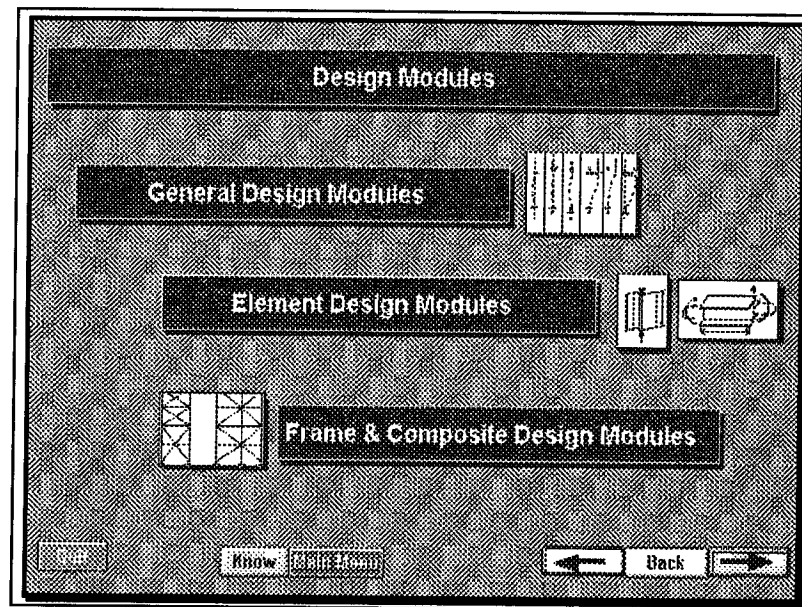


Figure 37, Design Modules Index

Element Design are those template dealing with individual frame elements of columns, beams and beam-columns.

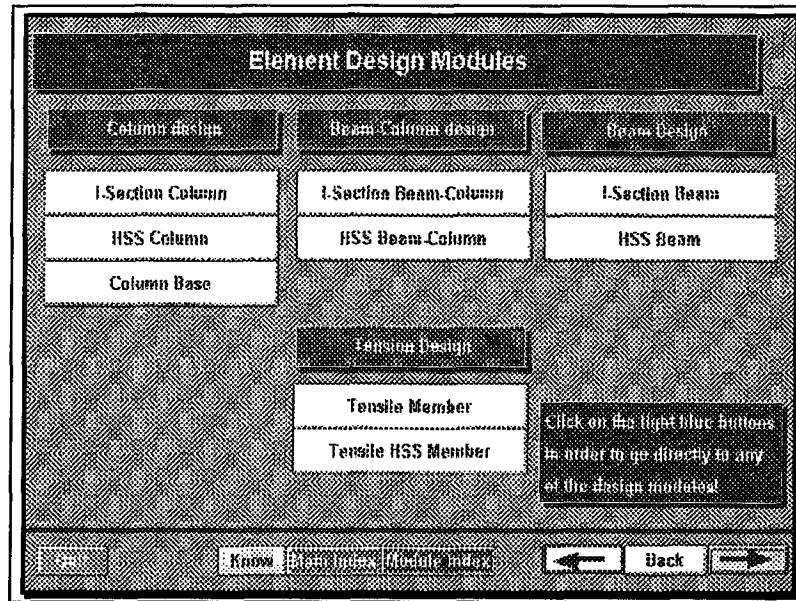


Figure 38, Element Design Modules

General Design template cover such templates as conceptual, load and reliability design.

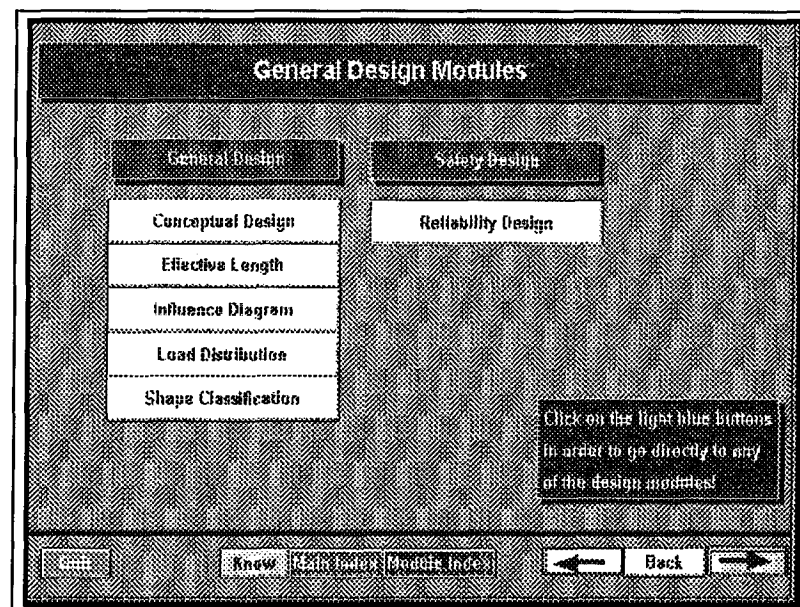


Figure 39, General Design Modules

Frame and composite design templates deal with moment frame, braced frame, plate girder design, connection design, story drift, composite beam design requirements.

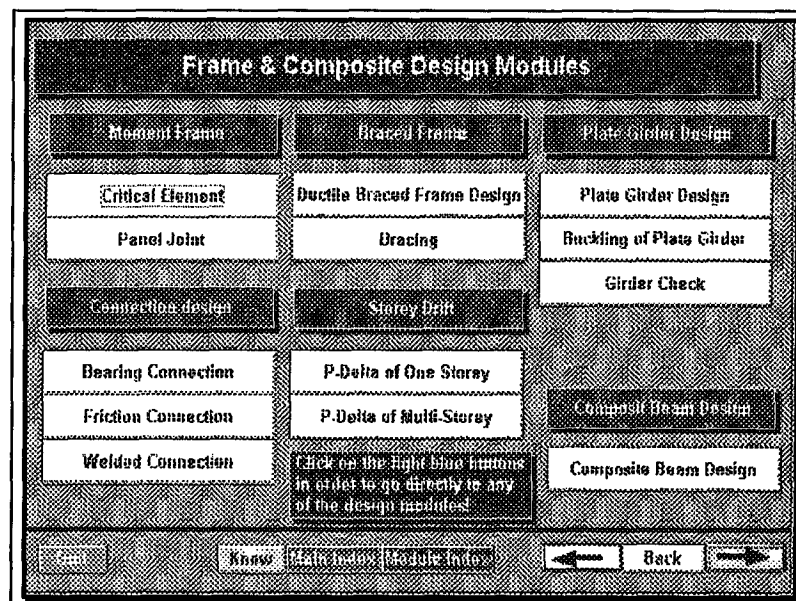


Figure 40, Frame & Composite Design Modules

iv. FEATURES OF DESIGN TEMPLATES

The design templates utilizing the **SEITRON** interface are standard in their format. Each template may contain a graphical representation of the element and comprise of **Input** and **Summary** sections.

Common to all templates are other features worth mentioning. These are briefly listed in tabular form and are discussed in more detail in later sections.

Features	Purpose
Fields cannot be overwritten in Summary Section	Prevent unnecessary inputs
Cursor Changes over certain fields	Information may be revealed by clicking onto the field
Menu Line at the bottom of the template	Command buttons for navigation or calculation
Cursor Changes within screen	The difference in cursor appearance indicates navigational links or various informational links
Graphical elemental changes at loaded and failed condition	Graphics effectively illustrates the stable and failed modes of a design

Table 5, Template Features

Also, throughout all design templates, there are various information support channels. As with traditional design handbooks, additional information such as a glossary of terms, member shape tables, various data tables and the like may be found. With **SEITRON**, support comes in the form of supporting knowledge banks of the *Info Library*, *Clause 27 Knowledgebase*, *Shape Table* and *Figure Library*.

v. MODIFICATION OF TEMPLATES

The **SEITRON** interface and the design templates are separate entities. The **SEITRON** interface need not be modified as the user input and desired output sections will undoubtedly remain unchanged. However, the flexibility to modify the calculations without affecting the interface is essential in the success of **SEITRON** as a design tool. Since the **SEITRON** interface simply creates a link between the interface environment and the spreadsheet, modification of the calculations is fast and simple. This is done by accessing the spreadsheet through **123W**, modifying the formulae and saving the file for future use.

In traditional programming, the coding must be modified every time there are changes to the clauses. This results in repeated transformations, debugging and recompiling of the program. Also, the program code containing the calculations is usually nested within the code for the interface. Thus when modifications are required of the calculations code, the code for the interface becomes vulnerable to accidental changes.

Proponents to traditional programming may argue in favour of increased calculation speed when traditional programming is used versus the reliance of *DLL*'s to transfer data with the use of external spreadsheet applications. However, increased computing speed does not outweigh the adverse modification considerations of programming.

F. DESIGN TEMPLATE EXAMPLE

The purpose of the *Design Templates* component was an attempt to illustrate possible future enhancements of structural design tools with the available software technology of the present time. This component was not created to provide a complete set of templates to cover all steel design situations.

At the present state of computer technology and within the constraints of this thesis, the practicality of the interface with the design templates is limited by the speed of data transfer and applications communication and by the available memory of the computing system. There is a tradeoff between multimedia enhancements and functionality of spreadsheets. Because of this, there may be some templates that are satisfactory without an interface and function better in the spreadsheet format. Nevertheless, future developments of multimedia may increase with advances in software and hardware and in turn enhance the use of multimedia interfaces as design tools.

The concept of an interface in front of a spreadsheet design tool has many advantages. With an interface, the programmer may control or restrict access to the formulation of the calculations so that users are not able to unknowingly or carelessly modify the formulae. In the case of **SEITRON**, the design templates may be easily modified by experienced spreadsheet users, however, the use of an interface forces the user to think twice about any changes as they may affect the operation of the interface.

Another benefit of an independent user interface is in the use of multimedia, pictures, graphics, and animation, which may more effectively illustrate the design concepts than pure numbers. The interface provides a user friendly environment that are often not found in analysis applications. Analysis programs often require the user to be fluent in the "application language". Through the

creation of a user friendly interface, the programmer may create an environment which is easy to use but at the same time retain the computational powers of the analysis program.

A common misconception of the user friendliness of the interface may be that any layman may design structural entities. The argument is, only the experienced engineer would understand the concepts and theory of the analysis and would be able to determine if the numerical results were reasonable or not. The interface allows the engineer to concentrate on the engineering and not to worry about correct syntax of the application.

The selection of templates used in conjunction with the **SEITRON** interface was based on the ideas of:

- ♦ functionality of the spreadsheet,
- ♦ possible graphical or animation enhancements to further illustrate the concept,
- ♦ usefulness and practicality of template,
- ♦ speed of data transfer template, and
- ♦ educational advantage.

The design templates presented in **SEITRON** are for the elements most commonly encountered in design and satisfy the conditions above. A current list of templates that are used with **SEITRON** include:

- ♦ Column Design of I-Sections,
- ♦ Column Design of HSS-Sections,
- ♦ Tension Design of I-Sections,
- ♦ Tension Design of HSS-Sections,
- ♦ Beam Design of I-Sections,
- ♦ Beam Design of HSS-Sections,
- ♦ Beam-Column Design of I-Sections,

- ♦ Beam-Column Design of HSS-Sections,
- ♦ Bearing Connection,
- ♦ Friction Connection,
- ♦ Shape Classification,
- ♦ P-Delta Displacement, Single Storey,
- ♦ P-Delta Displacement, Multi-Storey,
- ♦ Bracing Design of HSS-Sections .

This list is just a sample of the design possibilities with **SEITRON** and may be expanded at a future date.

While most templates may be enhanced with an interface, there may be some instances where it is not desirable to do so. Such an instance would be with the *Batch* mode templates. The *Batch* mode requires the arrangement of data in predefined columns and positions because operation of macros is through the relative movement of cells. Using an interface necessitates the transfer of data from a structural analysis program to the interface and then further data transfer from the interface to the spreadsheet. In the templates interfaced with SEITRON, the user inputs data directly to the interface. The *Batch* mode adds an extra step which reduces time and efficiency and takes away the primary concept of the *Batch* mode as a fast and efficient tool.

The column design template will be used as an example to illustrate the operation and show the benefits of the **SEITRON** interface.

i. COLUMN EXAMPLE

The design of columns follows the code guidelines from Clause 13.3 of CAN3-S16.1-M89. Steel columns are conveniently classified as short, intermediate, or long members and each category has an associated characteristic type of behaviour. A short column can resist a load equal to the yield load. A long column fails by elastic buckling and the maximum load depends only on the bending stiffness and the length of the member. Most columns common to steel buildings are in the intermediate range and failure is characterized by inelastic buckling and is greatly influenced by the magnitude and pattern of residual stresses and the magnitude and shape of the initial imperfections or out-of-straightness. The code constraints are to ensure stability and strength of the member.

Because the resistance calculation of columns is complex and is dependent on many variables, a simplified table of compressive resistances does not exist. The column design template bridges this gap as a useful design tool.

Characteristic of most design screens are an ***Input*** section, ***Summary*** section, graphics and menu line. The ***Input*** section contains all the variable inputs and is usually located at the top section of the screen. The ***Summary*** section contains purely results which is read-only information and is located on the bottom section of the screen. The graphics is located to the left or top of the screen and may change according to the results of the calculation. Finally, the menu line is always located on the bottom of the screen.

When the *I-Section Column* button is chosen from the *Element Design Modules* menu, the design screen as in Figure 41 appears.

Inputs

Member	W260 x 43	kx	1	Fy	360	MPa
		ky	1	E	200000	MPa
		Plate No.	930			
		Plate No.	904			

Summary

Classification		Resistance	
b/t ratio		Strength	
h/w ratio		Stability	
Class of Member		Efficiency	

Resistance

Comp. Resistance		kN
------------------	--	----

Status

Strength	
Stability	
Efficiency	

Menu Bar: CALC, Know, [Navigation Arrows]

Figure 41, Compressive Resistance of Column, Class C Template

The *Column Design I-Section* template has all the typical screen characteristics discussed above, including a picture of a column. One unique characteristic of this design screen is that the graphic may change with the results. During the input stage, the column is straight. However, depending on the desired slenderness constraints or calculated resistance, the column may start to buckle and a picture of the undeformed column may be replaced with a deformed column.

The menu line is the same for every template. However, investigation of the menu line reveals that evermore important for a design screen is a *Help* button. For users that are unfamiliar with how the design screen works, the *Help* button should always be accessed first. Other information provided from the *Help* button include hints about the layout or the format of the inputs and outputs, or an outline of special features of the screen.

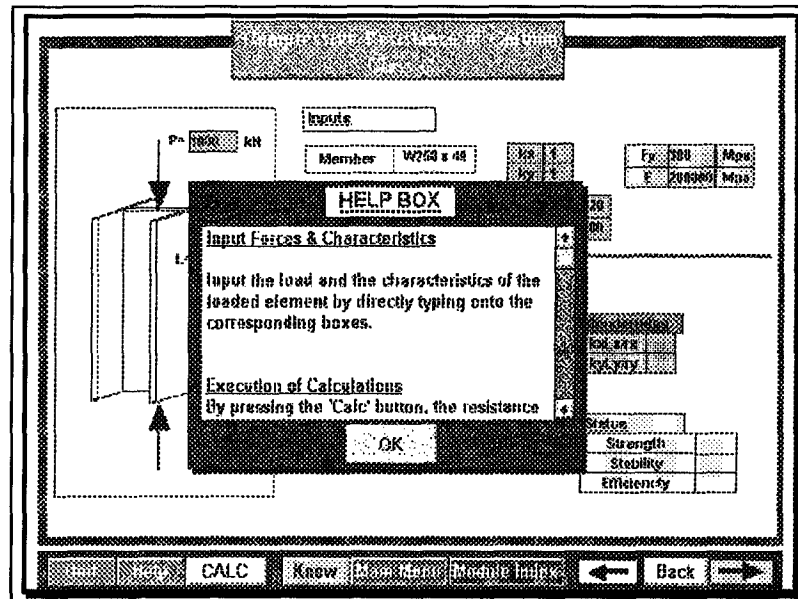


Figure 42, Compressive Resistance of Column, Class C with the Help Box

Depending on the complexity of the clause which may influence the complexity of the design screen, the *Help Box* may be very basic in its instructions or very complex. In this example, the *Help Box* contains an information scroll bar which gives insight as to

- ♦ how to input the member applied forces and characteristics,
- ♦ how execution of the design screen works, and
- ♦ how to get the definition of variables.

Scrolling through the *Help Box* suggests to the user that data may be entered by typing directly onto the corresponding field. Other information discussed include the execution of the design screen by clicking onto the *CALC* button and the definition of variables. This is achieved by passing the cursor over the field in question and noticing the change in cursor appearance to a question mark with a heavy arrow. Clicking onto the variable will then access its definition from the *Info Library* knowledgebase.

A column design example using the following inputs will be shown:

INPUT	Value	Unit
Member Size	W250x49	
Compression Force	1000	kN
Length of column	5000	mm
Effective length factor about X-Axis	1	
Effective length factor about Y-Axis	1	
slenderness limit about X-Axis	120	
slenderness limit about Y-Axis	200	
Yield strength of steel	300	MPa
Elastic Modulus of steel	200,000	MPa

Table 6, Column Design Example

These values are typed over the corresponding cells. Once these inputs are entered, the *CALC* button is clicked to perform calculations. The output appears as shown. The advantages to multimedia are shown in that given the inputs of the system, the column fails and the screen shows a deformed column.

Column Design Example of Column
Example 6.1

Inputs

Member	W250 x 49	r_x	110	mm
		r_y	110	mm
		E	200000	MPa
		ϕ_x	120	
		ϕ_y	200	

Diagram

$P = 1000 \text{ kN}$
 $L = 5000 \text{ mm}$

Summary

Classification		Slenderness Ratio
b/f ratio	5	$\lambda_{cx} = 110$
h/w ratio	30	$\lambda_{cy} = 110$
Class of Member	2	

Resistance		Status
Comp. Resistance	782.9 kN	Strength NG
		Stability OK
		Efficiency 1.31

Navigation: CALC, Know, Main Menu, Exit, Back

Figure 43, Results of Compressive Resistance of the Design Example Column, showing the Deformed Shape

From the given applied loads and constraints, the column fails in strength with a compressive resistance of 762.9 kN as compared to an applied load of 1000 kN. The slenderness ratios for both X and Y axis are below the constraints so there is no problem with stability. The width to thickness ratios indicate a Class 2 designation for W250x49 which indicates the possibility of the member attaining full plastic moment but with limited rotation ability.

One of the strengths of the template is in the ability of the user to choose pre-defined shapes that are available from the HSC shape tables, instead of manually inputting the member characteristics. The button *Member* is clicked to give the user access to the *Shape Table Library* database.

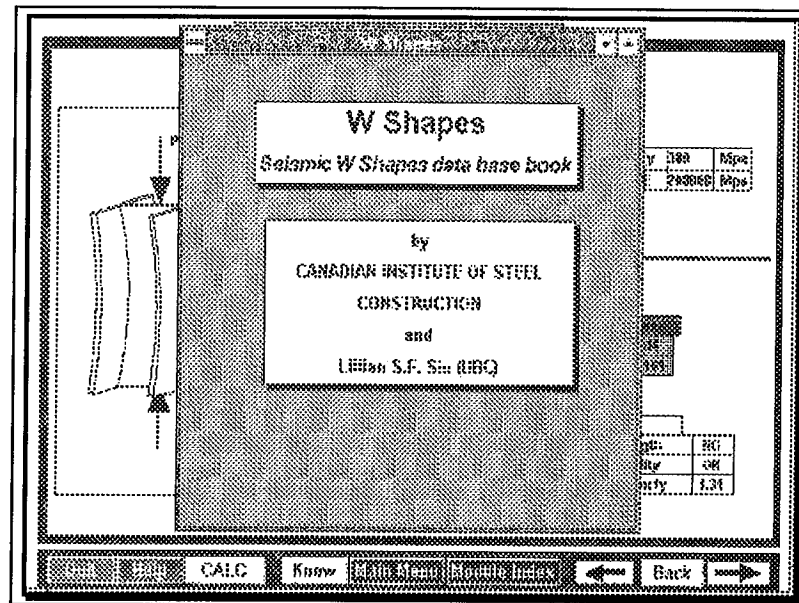


Figure 44, Activation of the W Shapes Library for the Compressive Resistance of Column, Class C

The *Shape Table Library* is presented as an overlay to the design template. Since the *Column Design, I-Section* template was accessed, the *Shape Table Library* automatically accesses I shapes (W shapes). More information regarding the *Shape Table Library* will be presented in Chapter 4, Supporting Infrastructure.

Once the *W Shapes* selection screen is accessed, the user chooses the nominal size of the member (Figure 45) and a more detailed section screen presents the member choices by weight breakdown (Figure 46).

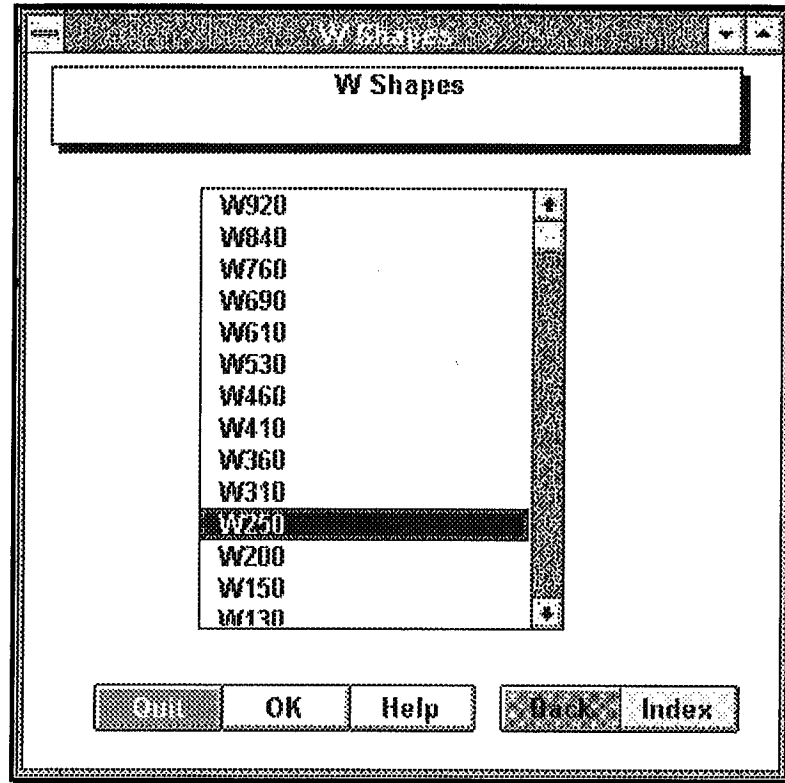


Figure 45, W Shapes Index

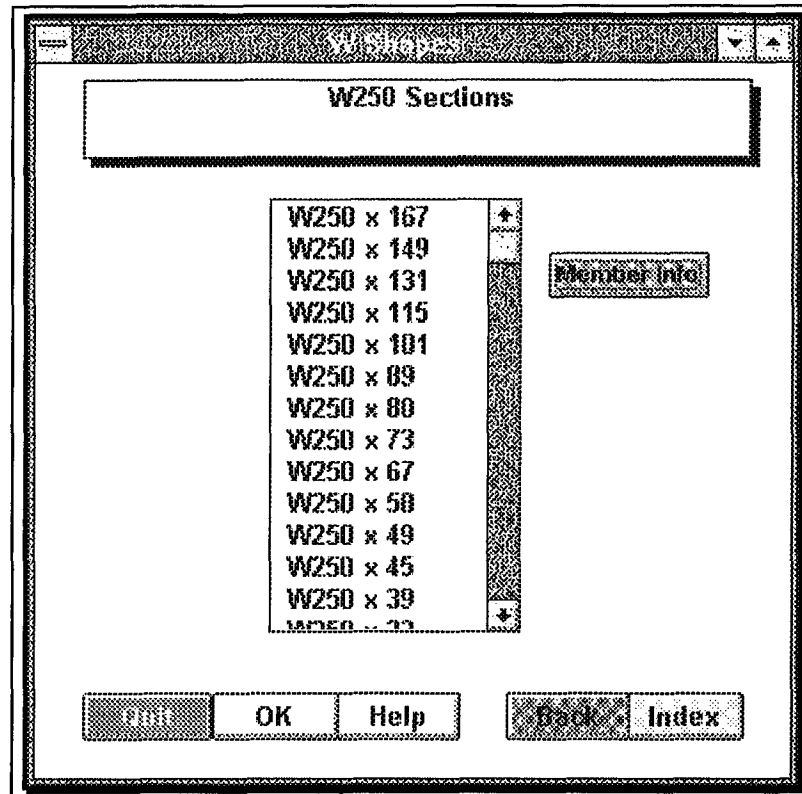


Figure 46, W250 Selection Index

The user selects the desired member size by clicking onto the member and then clicking onto the **OK** button. When a size of member is desired, the selection process is quick with the aid of the member selection windows. However, if there are further constraints to the member such as web or flange thickness considerations, it is beneficial to see these values. The button **Member Info** from the member selection by weight menu is clicked to see the geometrical values.

W250 x 49

Dead load	0.481 kN/m	Area	6250 mm ²
-----------	------------	------	----------------------

depth	247 mm	a	98 mm
flg. width	254 mm	b	194 mm
flg. thick	14 mm	c	28 mm
web thick	8 mm	d	19 mm

Torsional Constant	241,000 mm ⁴	Warping Constant	211 10 ⁴ mm ⁴
--------------------	-------------------------	------------------	-------------------------------------

	Inertia 10 ⁶ mm ⁴	S 10 ³ mm ³	r mm	Z 10 ³ mm ³
X-X Axis	78.6	572	49.2	633
Y-Y Axis	16.1	150	100	220

To accept the chosen element as the member for use in the design, click onto "OK". Click onto the "Back" button to chose another element.

Figure 47, W250x49 Member Geometric Properties

Member information for the W250x49 is shown in Figure 47. This menu is accessed by first clicking onto the W250x49 selection from the list on Figure 46 and then clicking onto the *Member Info* button. If the member characteristics are not as desired, the selection process may be repeated. That is, the user may choose the *back* button and go through the member selection menus again. If the geometrical constraints are adequate, the user confirms the selection by clicking onto the *OK* button. Control of the cursor goes back to the design template with the chosen material characteristics transferred to the template for use in the design.

Another feature to the SEITRON interface is the ability of the template to present more information than what is available on the screen. This ability is utilized in the *Definition of Variables* that was mentioned in the *Help* button.

Passing the cursor over the variables changes the cursor appearance from a white arrow to a solid black arrow with a question mark. This change in appearance indicates a variable definition that is obtained through the *Info Library* database.

Clicking onto the *kx* field reveals information about the field and also the possibility of more information in the form of information buttons within the definition screen.

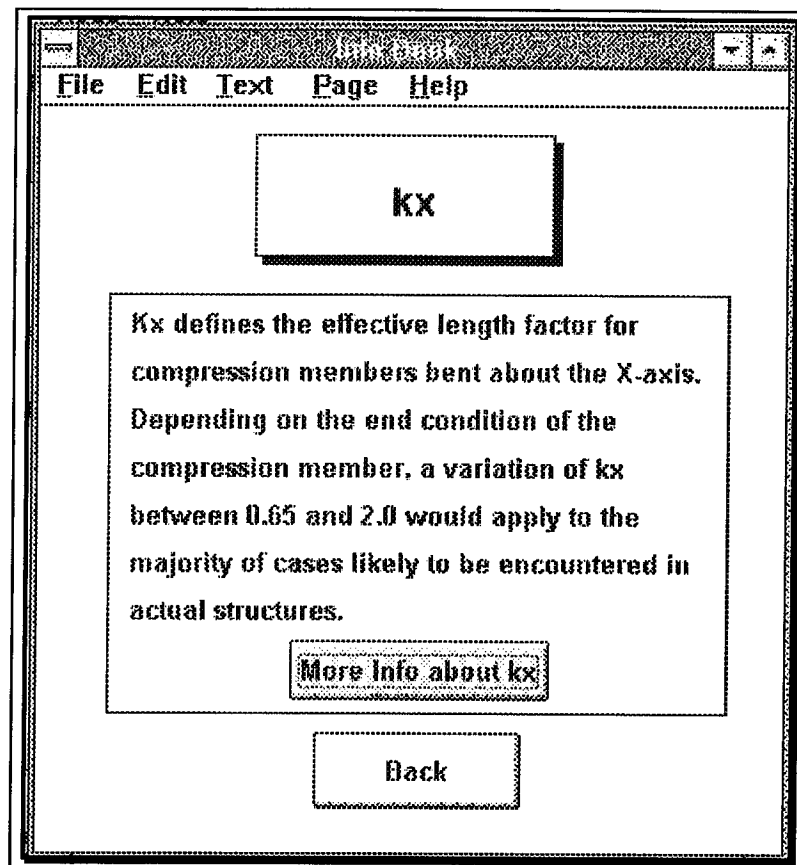


Figure 48, Definition of "kx" from Info Book

Although the information in the definition screen may be presented in the form of a scroll field to present more data, there may be some situations where there is not enough room to present all the material. In such situations, there may be additional buttons in the information bubble to access more information.

Clicking onto the *More info about Kx* button from the definition screen reveals common effective length factors as presented in the HSC for further clarification.

Effective Length Factors

Buckled shape of column is shown by dashed line

Theoretical K value	0.5	0.7	1.0	1.0	2.0	2.0
Recommended design value when ideal cond'n are approximated	0.65	0.8	1.0	1.2	2.0	2.0

End condition code		Rotation fixed	Translation fixed
		Rotation fixed	Translation fixed
		Rotation fixed	Translation fixed
		Rotation fixed	Translation fixed

Back

Figure 49, Effective Length Factor Data from Info Book

CHAPTER 4

SUPPORTING INFRASTRUCTURE

A. USE OF CURSOR APPEARANCE

Intelligent design of user interface environments combines the ease of use with the ease of information retrieval. The use of graphical linkages and the mouse pointing device provides the user-to-interface interaction.

Since all interactions are involved with the mouse pointing device, it is imperative that as much information is available on screen at any one time for potential information retrieval. However, placing too many concepts of a certain idea all on one screen would clutter the screen and cause more chaos than the original idea of having all information at the user's fingertips.

The solution of hypertext which effectively eliminates the clutter is by grouping pertinent data into nodes that may be retrieved through programmed links. The user is presented with buttons that are able to access different concepts of the main idea.

The next issue that arises is how does the user know when there is a link? That is, with so many graphic entities all within a screen, how does the user know if a button exists and that he is not clicking onto a non-existent button and by doing so, he is clicking onto a dead end.

In most cases, the presence of a button is obvious by its appearance. For example, buttons on the menu line are three dimensional in their appearance of a rectangle push button. Nevertheless, there may be instances where the shape of the button is not inherently obvious to indicate that a linkage exists.

Throughout **SEITRON**, provisions have been implemented in the interface to make the cursor appearance change whenever the cursor is positioned over an item that may reveal more information.

The normal appearance of the cursor is that of a small, pointer arrow. However, when the cursor passes over certain field or buttons, its appearance will change to reveal that a link is in place or that input is required. Below, in Table 7, is a list of cursor shapes, its meaning, and where they can be found within **SEITRON**. These cursor appearance changes are limited to linkages that do not fall under the appearance of conventional buttons. The cursor stays at its default value of a small, pointer arrow when conventional buttons are used. More information on conventional buttons are presented in the next section.







Cursor Shape	Cursor Description	Meaning	Location
	default	pointer	throughout SEITRON
	window	Link to <i>Clause 27 Knowledgebase</i> from either the clause, commentary, or reference	throughout SEITRON
	Arrow with black question mark	Information on elemental ductility requirements	<i>Moment Frame</i> investigation within the <i>Ductility and Redundancy Component</i>
	Arrow with black question mark	Definition of variables Link to the Info Library	<i>Design Template Component</i>
	I Shape	Elemental data inputs	<i>Design Template Component</i>
	Hourglass	Please wait	<i>Design Template Component</i> when the <i>Calc</i> button is activated

Table 7, Cursor Appearance

Within the *Clause 27 Knowledgebase* component, linkages are between clauses, commentary and references. The buttons are in the form of rectangles so when the cursor passes over these buttons, the shape of the cursor also changes to a box.

While the user is in the *Ductility and Redundancy* component of SEITRON, a discussion of the ductility requirements for each frame may be accessed. Within each graphical frame discussion, there are buttons relaying the specific design parameters and buttons in the menu line. Where it is obvious at locations where there are buttons, the user simply passes the regular pointer arrow cursor over the button and click onto the selection. Such is the case for the buttons within the menu line or for the radio buttons for the design parameter selection. However, there are instances where the use of standard buttons may not be suitable for the application.

In the investigation of elemental ductility requirements for moment frames, the user simply passes the cursor over the frame picture and over the individual elements of beams, columns, panel zone or connection to notice the change in cursor appearance. Again by pointing and clicking onto the element, an elemental information bubble is unveiled. Within the information bubble, there is another button providing access to *Clause 27 Knowledgebase*. This button is again not obvious so when the cursor is passed over the rectangle, the cursor changes into a window to indicate the potential to navigate to more information.






Within the *Design Template* component, a relationship has been set up between specific fields and the *Info Library*. This relationship defines variables used within the template design. When the cursor is passed over certain variables of the template, the cursor changes to a black arrow with the question mark to represent the possibility of variable definition. Clicking onto the variable gives the user access to the corresponding definition screen of the *Info Library*.

Also in the *Design Template* component, when data inputs are required, the cursor changes to the I shape to indicate to the user that data, usually numerical in format, may be entered.

B. TYPES OF BUTTONS

Buttons have two purposes in the interface. They may represent a linkage or they may be used in a selection process. Their specific appearance governs their usage.

The presentation of buttons is dependent on the type of information to be conveyed. For example, the shadowed buttons are used primarily in the menu line to direct the user to other components of SEITRON, whereas the checkboxes in the *Design Parameters* section of the *Design Template* component are used primarily for variable definition. Below, in Table 8, is a list of the types of buttons and their location within SEITRON.

Appearance of Button	Type of Button	Purpose	Location of Use
	Rectangle	Navigation	<i>Clause 27 Knowledgebase Ductility and Redundancy Design</i>
	Push	Navigation	throughout SEITRON <i>Menu Lines</i>
	Rounded	Navigation	throughout SEITRON
	Shadowed	Navigation	throughout SEITRON
	Radio Button	Navigation and Selection	<i>Ductility and Redundancy Design Design Template Component</i>


 Button	Checkbox	Selection	<i>Design Template</i> Component
---	----------	-----------	----------------------------------

Table 8, Button Appearance

Most of the buttons listed above act purely as navigation tools (i.e., clicking onto them activates the link in a pre-defined relationship). There are two styles of buttons, the radio button and the checkbox, that are often found grouped with other buttons of the similar style and whose function is not purely navigational.

The conveyance of a selection process in a graphical manner is usually presented in the form of buttons. Rather than providing an input box for the user to type in data, a selection of buttons with the possible choices is usually provided. This choice of input provides three main advantages

- ♦ minimize the number of user inputs and thus reduce the chance of human error,
- ♦ ease of use, and
- ♦ speed of input.

As listed in the table, checkboxes and radio buttons are used in the selection process. Checkboxes are grouped by two's and are used primarily in the *Design Template* component in individual elemental design templates. Their function is to assign one of two values to variables that are used in the elemental calculations. That is, with a design criterion that has one of two possible options, the checkbox buttons are employed such that the user may select one of the two choices.

For example, in the *Beam Design* templates, the user must assign the type of curvature (single or double curvature) that the beam undergoes for each of its axis. Under the design criterion of *Curvature of the Beam*, appears two checkboxes: one checkbox pertaining to single curvature and the other checkbox pertaining to double curvature. The user simply clicks onto the checkbox

that is relevant to his design and the checkbox activates the corresponding variable for use in the spreadsheet calculations. Whenever a check box is selected, an *X* automatically appears.

The use of radio buttons has a dual objective that is dependent on the location of its use. The radio buttons are seen to be utilized in both navigation and variable assignment duties. In either use, they are arranged together in groups of three or more.

When in use as navigation tools, they form a sub-menu of navigational choices. Such an example lies in the *Index* for the *Ductility and Redundancy component*. In this application, radio buttons display the types of frames that are presented. By selecting any of the radio push buttons, the user is directed to the design requirements of the frame desired.









Radio buttons are used for variable assignment in the *Design Template component*. The user of radio buttons in variable declaration is similar to that of the checkboxes. However, where a design parameter has more than two possible choices, radio buttons are used. Clicking onto the button colors the circle and assigns certain criteria in the design calculations.

C. MENU LINE

Information management is focal for **SEITRON**. While the format of an electronic handbook provides completeness, easy navigation, and structured organization, it is easy for the programmer to expand such a system such that the realms of navigation are lost.

As discussed before in the limitations of hypertext, the user may become lost when he is uncertain as to how he got there and how he is going to get out. In terms of the interface environment, the user may progress to a certain component of the system through a series of linkages but the linkage relationships may not be transparently clear or logical. The situation arises as to finding and utilizing suitable links such that the user may clearly navigate to the desired destination.

SEITRON has been created with these problems in mind for maximum hypertext *survival*. Part of this solution is achieved by employing a standard menu line that is seen throughout all three components of **SEITRON**. Within this menu line are buttons that are used in all levels and components:

Button Appearance	Button	Function
	Quit	end SEITRON
	Help	activates Help menu
	Main Menu	go to the <i>Main Menu of SEITRON</i>
 	"Sub-Menu" such as Frame Index or Module Index	go to the Sub-Menu of current component
	Know	access to <i>Clause 27 Knowledgebase</i>
	Backward arrow	go to the page before the current page
	Back	go to the previous page



	Forward arrow	go to the next page
	CALC	calculates template with desired inputs in the <i>Design Template</i> component

Table 9, Description of Menu Line Buttons

The main button line maintains its position, shape and appearance throughout **SEITRON** such that the user may have easy access, familiarity and continuity for their use. Below are the menu lines for each of the three components of **SEITRON**.



Figure 50, Menu Line for Clause 27 Knowledgebase

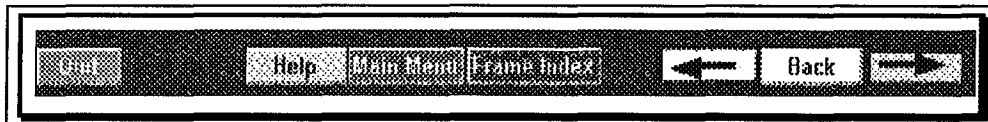


Figure 51, Menu Line for Ductility and Redundancy Design



Figure 52, Menu Line for Template Design

Within the three different menu lines, there are buttons that are common to each component: *Quit*, *Help*, *Main Menu*, backward, *Back* and forward. These buttons ensure that users do not get lost. Through the use of the *Main Menu*, *Sub-Menu*, backward, *Back* and forward buttons,

the user has the option of going back to his initial starting point or to his previous screen whenever he desires. It is in this way to ensure that the user does not move further into hyperspace to later discover that it is difficult to return to his starting point.

There are also other menu line buttons for button functions that are component specific. In the case of the *Design Template* component, the *CALC* button is included such that elemental design calculations may be implemented.

D. USE OF HELP SYSTEM

Throughout each interactive screen, a *Help* button exists in answering common user inquiries. Once the *Help* button is activated, a Help screen appears. Each *Help* button is screen specific. Since each screen is unique in its contents and in the information that it is conveying, there are points of interest that may be relevant to one screen and may not be relevant to another.

The *Help* button is more important in the *Design Template* than the other components. In the *Clause 27 Knowledgebase* and *Ductility Design* components, the *Help* button mainly acts as a general overview as to the operation of the screen in question. However, in the *Design Template*, *Help* informs the user of specific design criteria that is essential in the operation of the template. For example, *Help* may

- ♦ alert the user to special functions,
- ♦ discuss how data is to be input,
- ♦ discuss the sign convention of the inputs
- ♦ review limitations to the design,

- ♦ show how the supporting knowledgebase may be utilized, and
- ♦ show how the supporting infobase may be utilized.

As an example, the *Help* button from the beam template design screen will be investigated. *Help* gives a general description of how to execute the template and it also alerts the user to special things to watch out for.

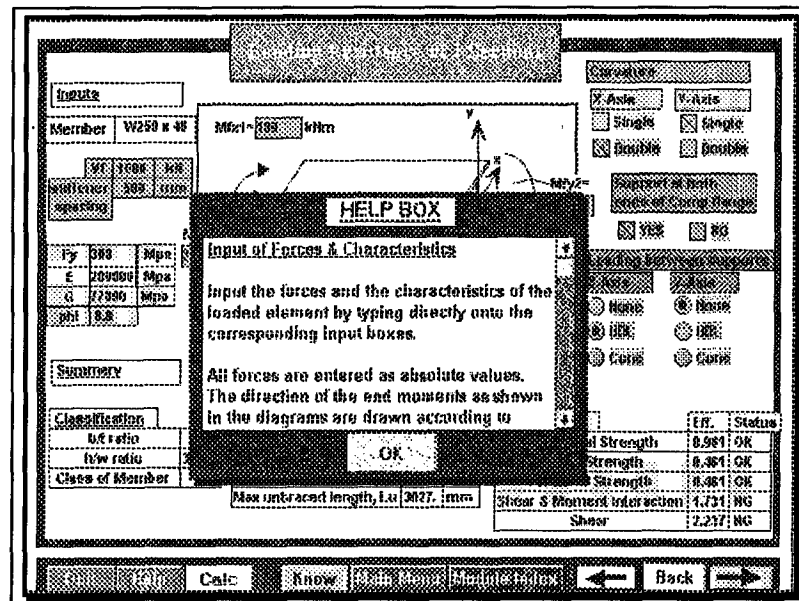


Figure 53, Bending Resistance of I-Section with Help Box

In this case *Help* presents four ideas as to the operation of the template:

- ♦ Input of Forces and Characteristics,
- ♦ Code Requirements,
- ♦ Definition of Variables, and
- ♦ Execution of Calculations.

The *Input of Forces* is the most important concept to the template and is template specific. This comment alerts the user to input all beam forces as positive values. Instead of inputting the forces

with their correct signage, the curvature of the beam that is expressed through the signage of the forces is instead explicitly entered as a button selection.

The other comments from the *Help* screen are common to other templates. *Code Requirements* informs the user of the clauses used within the template. The *Definition of Variables* alerts the user as to the use of *Clause 27 Knowledgebase* and *Infobase* to explain the variables used in the template. Finally, the *Execution of Calculations* informs the user to click onto the *CALC* button to perform the calculations.

While some of these comments are self-explanatory, other comments are useful in the successful execution of the template or screen. Thus the user is encouraged to read the *Help* button before attempting to execute the template or screen.

E. KNOWLEDGEBASE BUTTON

Since **SEITRON** can be used like an electronic handbook, access to clauses is one of the most vital components of the system. The *Ductility and Redundancy* and *Design Template* components require the support of the clauses to inform the user of the code application. It is therefore vital to be able to access *Clause 27 Knowledgebase* at any time.

The *Know* button that is found in the menu line is the linkage between the Design Template component to its seismic provisions. The *Know* linkage is to the *Clause 27 Index*, where clauses are listed in chronological ordering. The user may then scroll to the clause in question or access the *Table of Contents* button if the user is uncertain of the location of the clause.

F. INFORMATION LIBRARY

The *Info Library* is used primarily with the *Design Template* component. In these templates, the data displayed in the *Inputs* and in the *Summary* sections are often written as variables. The entire description of the input or output was not presented due to a limitation in the screen size and the system design requirement to show all inputs on one screen. Although care was taken to assign variables as representative as possible to the inputs and outputs, there may be uncertainty regarding their meaning.

A database of definitions that clarify the variables and gives some background in its use was implemented in the *Info Library*. While in the *Design Template Module*, the user may notice that when he passes the cursor over certain input or output fields that the cursor appearance changes to a large black arrow with a question mark. A link has been created with the field and the *Info Library*. By clicking onto the field, SEITRON refers to the *Info Library* regarding the variable at hand.

For example, users may be familiar with the variable "kx_lx/r_x" that is used frequently throughout the design templates. However, he may be unsure as to a suitable numerical value. Clicking onto the kx_lx/r_x field reveals information regarding the slenderness ratio and a suitable value for use in steel design. All theory that is given as background material within the *Info Library* is based on the code provisions of CAN/CSA-S16.1-M89.

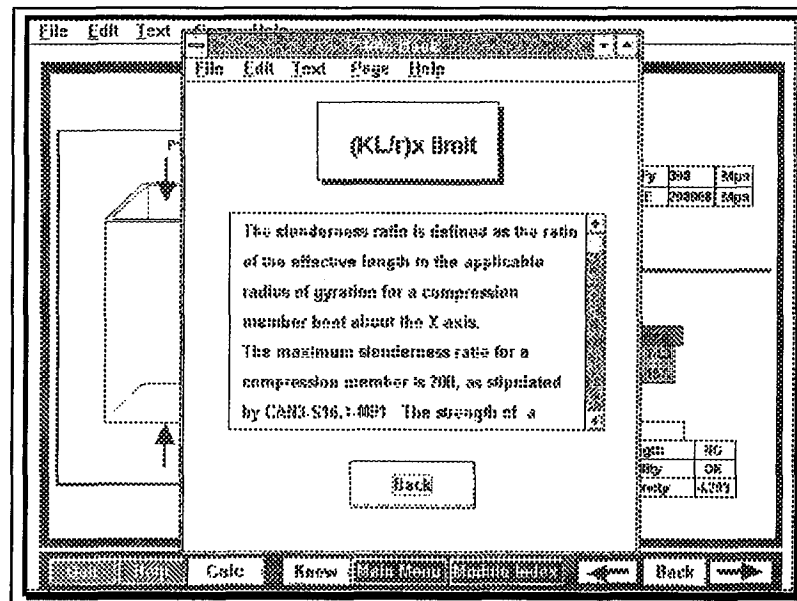


Figure 54, Help Info Library Activated to show Information on the "(KL/r) limit"

The *Info Library* may also be accessed on its own from the *Main Menu*. The title page and following pages from this database are shown. Clicking onto the *Next* button of the *Info Library* menu line gives the user access to the *Index of Variables*. Note that the menu line is similar to the menu lines of the **SEITRON** components in that control is in the form of *Quit*, *Previous*, *Back* *Next* buttons.

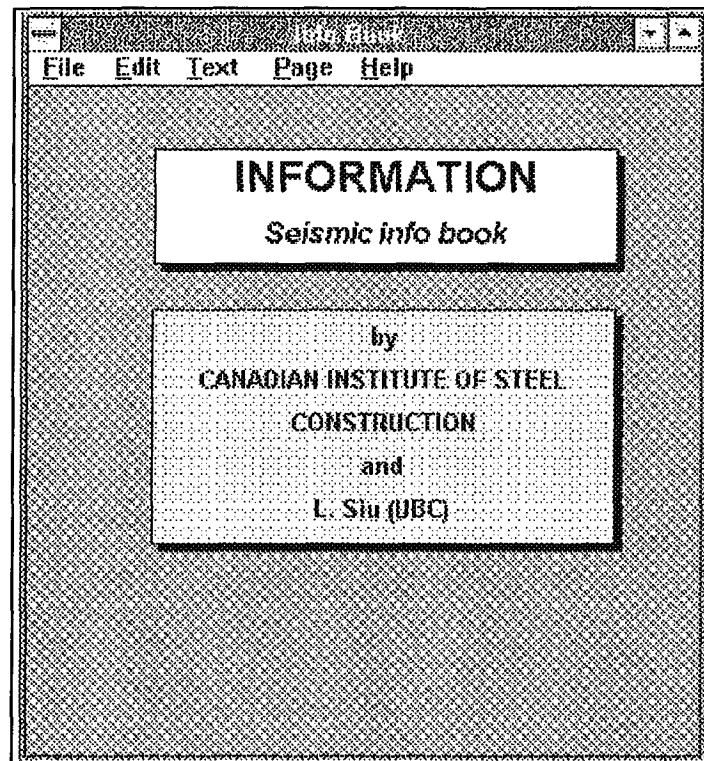


Figure 55, Cover Page of Seismic Info Book

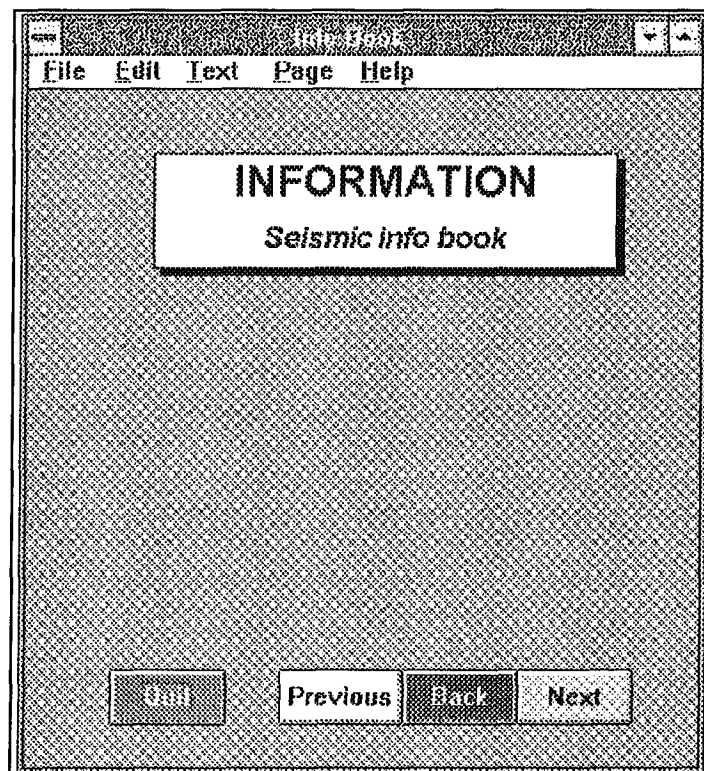


Figure 56, Title Page of Seismic Info Book

The *Index of Variables* is similar to the *Index of Clause 27 Knowledgebase*. An index of the variables in alphabetical listing is presented. The user simply scrolls through the list to find the desired variable and then clicks onto the variable in question to access the variable information screen. Similarly, instead of scrolling down the list, the user may click onto a letter of the alphabet that initializes the variable and the list advances to that variable.

The variable *Beta* from the *Bearing Connection* design template is chosen as an example to illustrate the definition information screen.

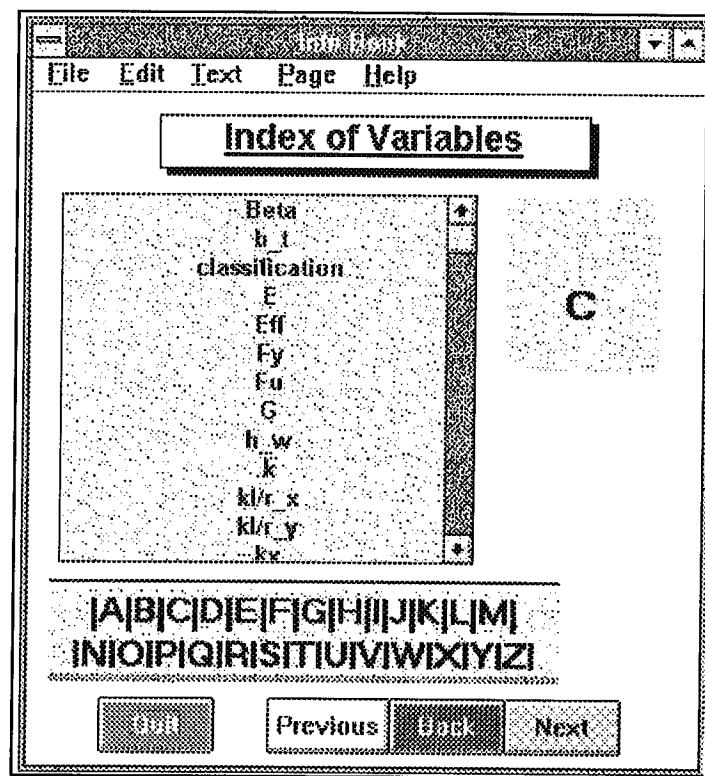


Figure 57, Index of Variables from the Info Book

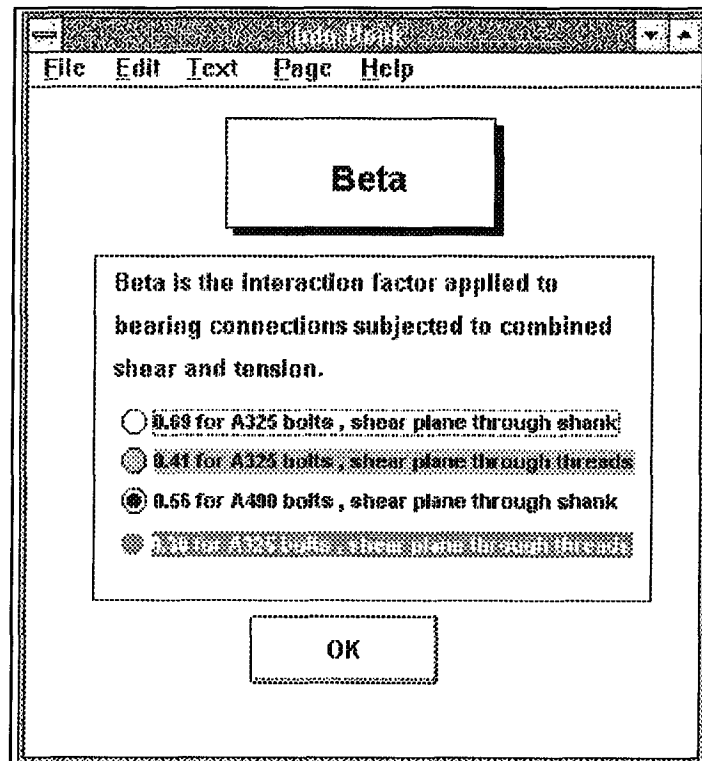


Figure 58, Beta Information from Info Book

G. SHAPE LIBRARY

The *Design Template* component of SEITRON requires the support of the *Shape Library* knowledgebase to provide member characteristics. The *Shape Library* comprises of the structural sections listed in the HSC with their dimensional and sectional characteristics.

Access to the *Shape Library* may be through the *Member Size* button from the *Design Template* component or through the *Main Menu*.

Operation of the *Shape Library* as it pertains to the *Member Size* button has been discussed in the *Column Design* example. Generally, when the *Member Size* button is clicked, the *Shape Library* database is activated and the user is presented with a selection of member sizes.

Operation of *Shape Library* from the *Main Menu* allows user access to any of the members that comprise the database.

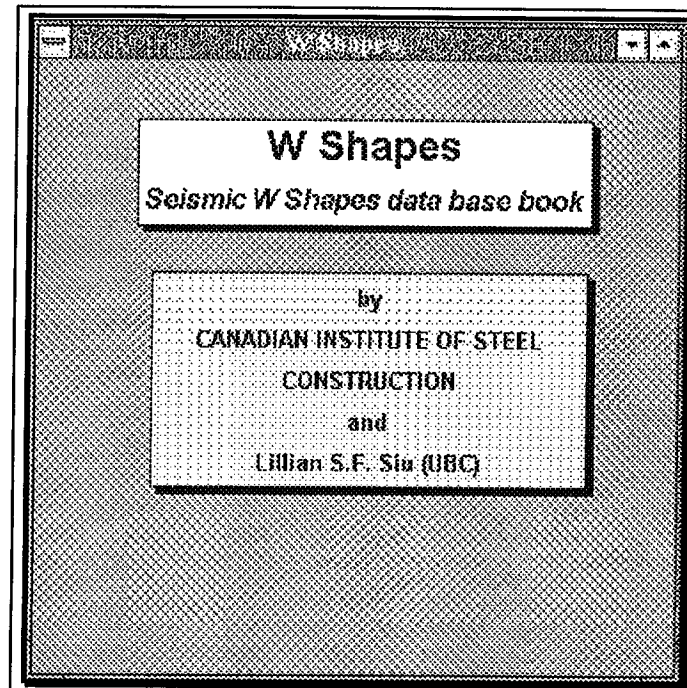


Figure 59, W Shapes Title Page

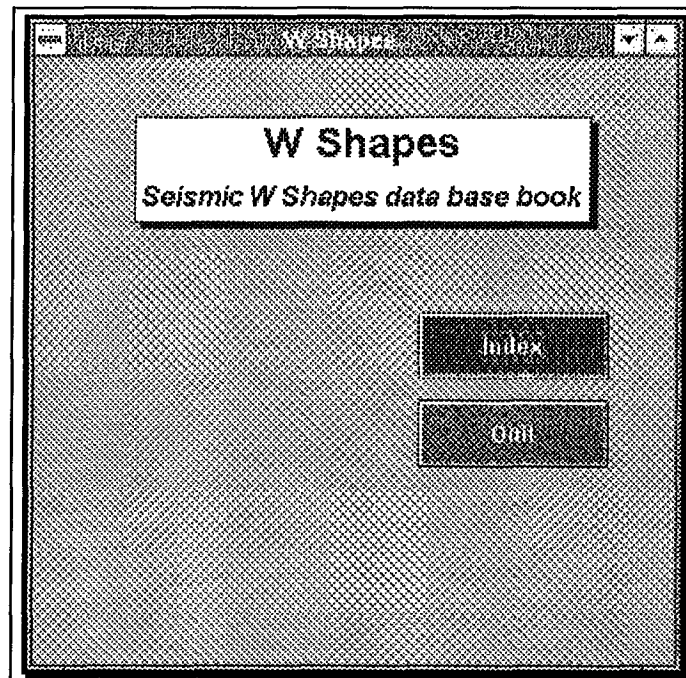


Figure 60, Access to W Shapes via the Index

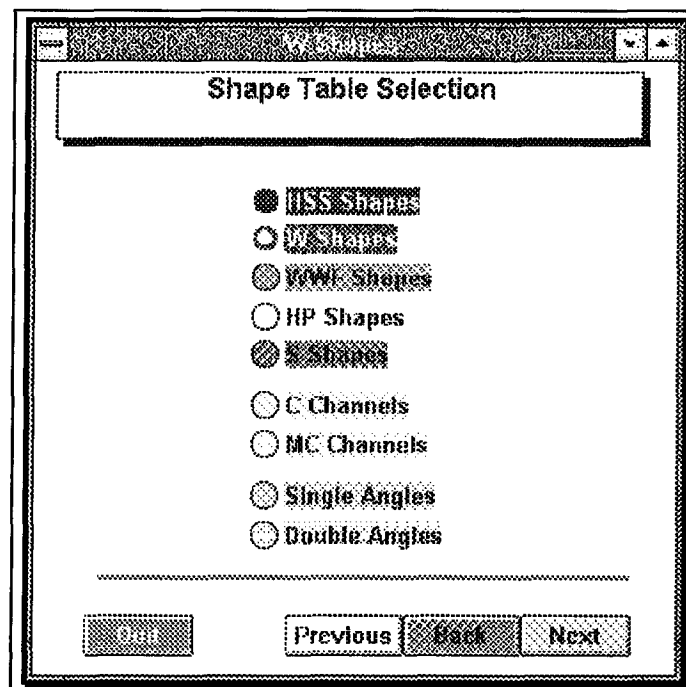


Figure 61, Shape Table Selection

The *Shape Table Selection* index of the *Shape Library* lists the shapes available for use. These shapes range from W shapes to Angles. Generally, clicking onto the desired shape accesses a sub-group of the shape by size. Clicking onto the appropriate size may reveal another selection grouped by weight. Finally, at the end of the selection process, the user may inspect the member section dimensions and geometric properties by clicking onto the *Member Info* button.

W sections are the predominate shape used in structural design. Due to the large range in sizes, W shapes are sub-grouped by their depth. Clicking onto W-Shape as the desired structural member reveals a choice of section sizes ranging from W100 to W1000. Clicking onto the desired section depth will access the final listing of members with common depth but varying weights. The user may find the desired size for use in the template design by clicking the *OK* button on he may review the sectional properties of that shape by clicking onto the *Member Info* button.

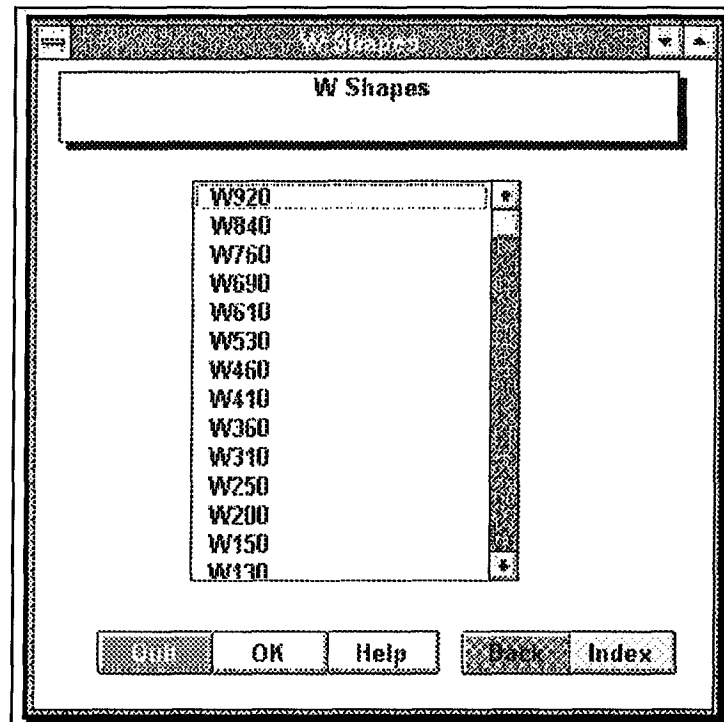


Figure 62, W Shapes Index

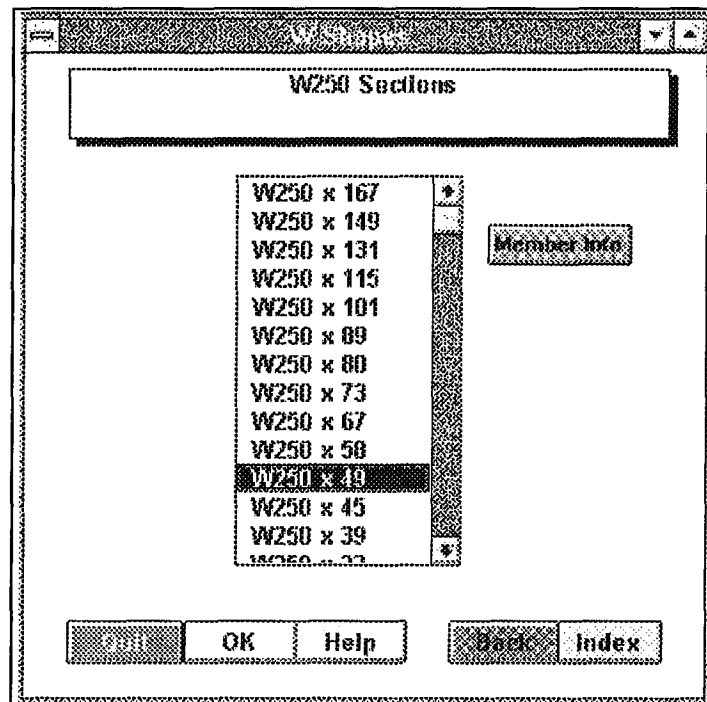


Figure 63, W250 Sections Index

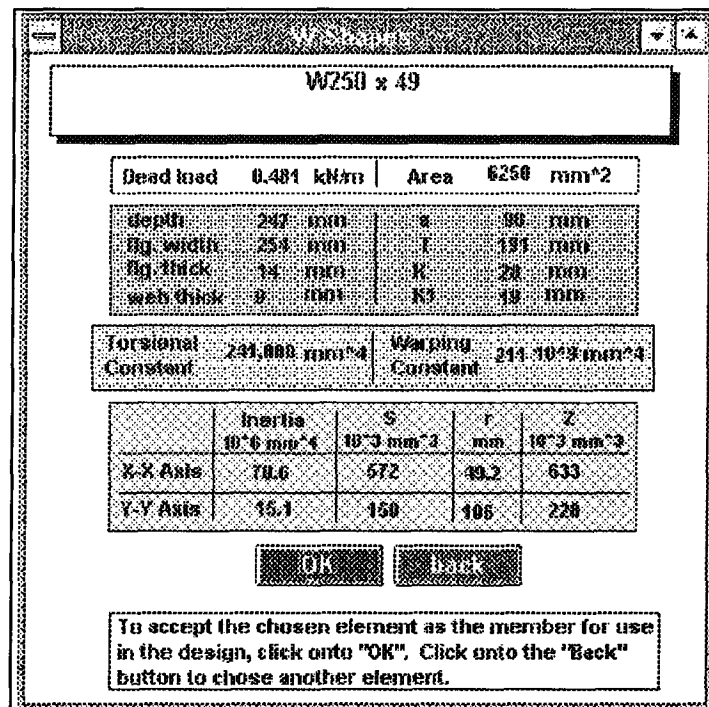


Figure 64, W250x49 Geometrical Properties

It is worth noting that at any time within the section process, the user may change his mind of the choice of section by going back to the previous screen or going back to the *Shape Table Selection* index. Thus if the user is reviewing the sectional characteristics of W shapes and finds that the properties are inappropriate, he may access another member with different depth by clicking onto the *Back* button or he may decide to use another shape altogether by clicking the Shape Index button.

Once the desired shape has been chosen by confirming with the *OK* button, all the sectional properties are transferred to the Design Template for use in the calculations.

CHAPTER 5

CONCLUSION

This thesis has attempted to explain the potential of a hypertext system as an integrated design tool. During the research and development of this system, the current steel design practices using traditional programming, along with the new seismic provisions and the design needs of the structural engineering community were evaluated. Complex code requirements and the potential of high level computing fueled the investigation of a new design tool. The outcome was a compilation of requirements which this thesis has endeavored to incorporate into the next generation of graphical design tools:

- ♦ provide a tool which may perform structural design and provide access to code requirements,
- ♦ clarify the design requirements within the steel code with the use of supporting material in the form of commentary, references, graphics,
- ♦ provide design tools for commonly used structural elements, and
- ♦ provide user-friendly atmosphere.

The advancement of seismic research has created new provisions to ensure ductile behaviour of structures. With these new provisions, **Clause 27** of **CAN/CSA-S16.1-M89**, the steel design code has recently undergone code changes to ensure ductile frame behaviour whereas in the past, ductility was deemed to be an inherent quality of steel design. Such changes have made the code difficult to interpret and may have brought about confusion to the steel designer and fabricator.

In an attempt to clarify the seismic provisions, innovative computer solutions were investigated. The introduction of computer technology into the office environment provided an opportunity to introduce new concepts and approaches into the design field.

The choice of a hypertext system approach was considered to meet the challenges of these criteria through an interactive system based on linkages, Dynamic Link Libraries (*DLL*) and Dynamic Data Exchange (*DDE*) communication and supporting features of graphics, colors, buttons. Through a linkage system, a large knowledgebase of various media may be presented to better explain the code concepts. In addition, hypertext may communicate with other applications and as such, it may integrate other applications in order to meet its member design criteria.

To answer the issues presented in this thesis, the interactive system **SEITRON**, **SEI**smic **ElecTRONic** Handbook, was created.

SEITRON serves three purposes:

1. Provide an electronic reference of Clause 27, along with electronic links to commentary and references.
2. Quantify the critical elements of each frame system and provide the corresponding code design requirements through the use of graphics.
3. Provide design templates as engineering design tools which utilize Clause 27 and other CAN/CSA-S16.1-M89 design requirements.

To effectively fulfill these three purposes, SEITRON is organized into three main components:

- ♦ Clause 27 Knowledgebase,
- ♦ Ductility and Redundancy Requirements, and
- ♦ Design Template.

Part of the potential confusion in Clause 27 is the inter-relatedness of the individual clauses. The *Clause 27 Knowledgebase* component acts as a hypertext which allows the user to access code information and theory of seismic steel frame design parameters. This inter-relationship is effectively managed in an environment of links and buttons. Further, the use of graphics and the interactive mode allows the user to access information that is pertinent to one's requirements and to omit nonessential material. The non-structured characteristic of **SEITRON** allows the user to browse through related material which in turn may clarify the seismic concepts.

Such a browsing ability is inherent in the *Ductility and Redundancy* component which clarifies the inter-relatedness and overlapping of code conditions for the four main frame configurations: Ductile Moment Frame, Nominally Ductile Moment Frame, Braced Frame, and Nominally Ductile Braced Frame. This component is a graphical representation of the ductility and redundancy requirements of frames and how the code provisions are applicable. The user chooses the critical design parameter and **SEITRON** defines the code conditions for each of the frame elements with respect to the chosen parameter.

This component acts similar to a teaching tool with its graphics and pop-up information capsules. However, its function is enhanced with the power of linkages between the graphic capabilities and the code provisions from the *Clause 27 Knowledgebase* component.

The hypertext ability of **SEITRON** is enhanced with the extension of spreadsheets as a useful design tool. Presently, there are many software packages for the static, dynamic and finite analysis of structures but there are few tools available that are able to interpret the output from the analysis tools into useful members sizes. **SEITRON** addresses this issue with the incorporation of commonly used design templates with the hypertext system.

In the *Template Design* component, **SEITRON** acts as an interface and controller of data flow. Inputs and outputs are presented in graphical fields for ease of design and presentation within the **SEITRON** interface. However, all calculations are performed on formatted templates behind the interface. The user inputs the desired data into the corresponding fields and **SEITRON** transfers the inputs and outputs between the template and interface. Communication between template and interface is through a series of complex DLL's and DDE's.

Convenience in the design tool is in the use of the steel member database for resistance calculations and in the ease and flexibility of spreadsheet templates for calculation. In terms of the member database, the user no longer has to manually input sectional and geometrical properties for design calculations but may choose the desired member through a menu system. The physical Steel Design Handbook may be replaced with an electronic one.

Strength in **SEITRON** comes with the recent **WINDOWS®** and Graphical User Interface (*GUI*) technology which allows communication between files and enhances graphics capabilities. This is important as the computing medium and the interface are kept as separate entities. The capabilities of **SEITRON** as a hypertext system are good, however, its computational powers are limited and hence the formatted spreadsheets were incorporated to complete the computational requirements.

Further to the power of the spreadsheet is the ease of formulation changes that can be done separately from the interface code. Since design code changes are regulated every four years, the spreadsheet may be easily modified without becoming obsolete as would be the case in traditional programmed interfaces.

Finally, it is important that the hypertext system be easy to learn and use. Object-oriented techniques were incorporated into the hypertext system shell to ensure that it is portable and user friendly. The use of spreadsheets as the computing medium is also easy to learn and to modify when further code changes are required. Also, since **SEITRON** is not composed just of text but contains a plethora of graphics, colors and buttons, the interactivity between user and system may seem fun and may enhance reader perception of the material.

It is the author's hope that **SEITRON** may serve not only as a design tool in a professional setting but that it may also act as a teaching aid for engineers to acquaint themselves with seismic design provisions.

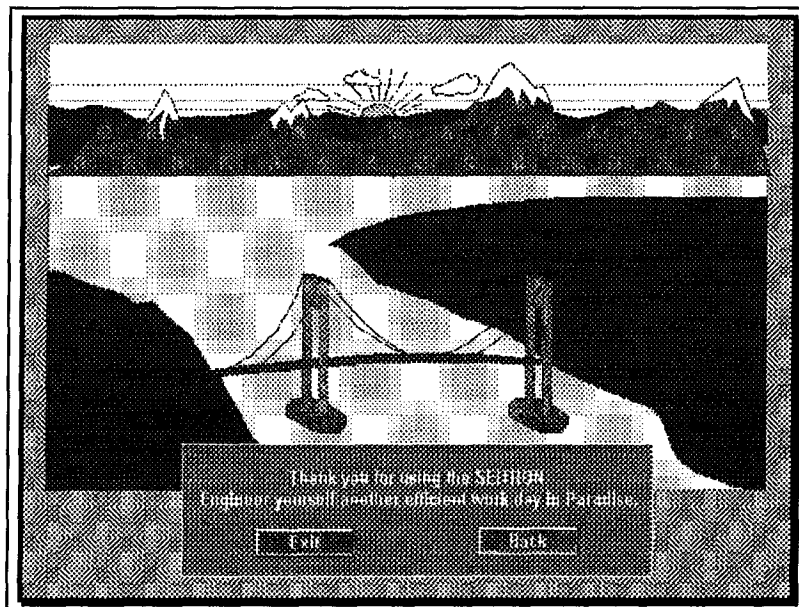


Figure 65, Good-bye!

BIBLIOGRAPHY

1. ASCE, *Expert Systems for Civil Engineers: Education*, 1989.
2. Asymetrix, *Using Toolbook, A guide to Building and Working with Books*, Asymetrix, 1989.
3. Beerel, Annabel C., *Expert Systems: Strategic Implications and Applications*, Ellis Horwood Limited, England, 1987.
4. Beaulieu, D., Perlynn, M., Dunbar, A., Adams, P.F., and Keller, D., *The Effects of column out-of-plumbs on the stability of core-braced buildings*, Canadian Journal of Civil Engineering, 3(3), 1976.
5. Birtwistle, Graham, *Artificial Intelligence Graphics and Simulation*, Society for Computer Simulation, La Jolla, California, 1985.
6. Blattner, M et al., *Multimedia Interface Design*, Addison,-Wesley, New York, 1992
7. Canadian Institute of Steel Construction, *Handbook of Steel Construction*, Fifth Edition, October 1991.
8. Canadian Society for Civil Engineering, *Earthquake Structural Design Seminar*, Vancouver/Victoria, April 1990.
9. Carpenter, James, *Response of Steel Frames to Earthquakes*, March 1990.
10. Chien, Eddie, *Connection Design & Ductility Requirements to NBC & S16.1*, Canadian Institute of Steel Construction, Willowdale, Ontario, 1992.
11. Chien, Eddie, *Low Rise Office Building Design Aid*, Canadian Institute of Steel Construction, Willowdale, Ontario, First Printing, May 1991.
12. Cheong-Siat-Moy, Francois, *Column Design in Gravity-Loaded Frames*, Journal of Structural Engineering, Vol. 117, No. 5, May, 1991.
13. Chorafas, Dimitris N., *Knowledge Engineering*, Van Nostrand Reinhold, New York, 1990
14. Fleming, C. and von Halle B., *Handbook of Relational Database Design*, Addison-Wesley Publishing, Massachusetts, 1989.
15. Hawkins, N.M. and Mitchell, D., *Seismic Response of Composite Shear Connections*. ASCE Journal of Structural Engineering, 110(9), 2120-2136, 1984.

16. IEA/AIE, *Proceedings The Third International Conference on Industrial & Engineering Applications of Artificial Intelligence & Expert Systems, Volume II*, Charleston, South Carolina, 1990.
17. Jackson, Peter, *Introduction to Expert Systems*, Second Edition, Addison-Wesley, Great Britain, 1990.
18. Jain, A. et al., *Hysteretic Cycles of Axially Loaded Steel Members*, Journal of the Structural Division, ASCE, Vol. 106, No. ST8, August, 1980.
19. Krawinkler, H., and Popov, E., *Seismic Behavior of Moment Connections and Joints*, Journal of the Structural Division, ASCE, Vol. 108, No. ST2, February, 1982.
20. Kennedy, D.J.L., Picard, A., and Beaulieu, D., *New Canadian Provisions for the Design of Steel Beam-Columns*, Canadian Journal of Civil Engineering, 17(6), 1990.
21. Lay, M., and Galambos, T., *Bracing Requirements for Inelastic Steel Beams*, Journal of the Structural Division, ASCE, Vol. 92, No. ST2, April, 1966.
22. Liebowitz, J., and De Salvo D., *Structuring Expert Systems, Domain, Design and Development*, Prentice-Hall, Inc, New Jersey, 1989.
23. Liu, Z and Goel, S., *Cyclic Load Behaviour of Concrete-Filled Tubular Braces*, Journal of Structural Engineering, Vol. 114, No. 7, July, 1988.
24. Massey, C., *Lateral Bracing Forces of Steel I-Beams*, ASCE Engineering Mechanics Division, 88(EM6), 1962.
25. McKnight, C., Dillon, A., Richardson, J., *Hypertext in Context*, Cambridge University Press, Cambridge, 1991.
26. Nixon, D., *The Use of Frame Action to Resist Lateral Loads in Simple Construction*, Canadian Journal of Civil Engineering, 8(4), 1981.
27. Pierce, Joseph R., *Toolbook Companion*, Microsoft Press, Washington, 1990.
28. Popov, E., *On California Structural Steel Seismic Design*, Earthquake Spectra, Vol. 2, No. 4, 1986.
29. Popov, E., and Black, R., *Steel Struts Under Severe Cyclic Loading*, Journal of the Structural Engineering Division, ASCE, Vol. 107, No. ST9, September, 1981.
30. Popov, E.P. and Pinkney, R.B., *Cyclic Yield Reversal in Steel Building Connections*, ASCE Journal of the Structural Division, 95(ST3), 1969.

31. Popov,, E.P. and Stephen, R.M., *Tensile Capacity of Partial Penetration Groove Welds*, ASCE Journal of the Structural Division, 103(ST9), 1977.
32. Nixon, D., *The Use of Frame Action to Resist Lateral Loads in Simple Construction*, Canadian Journal of Civil Engineering, 8(4), 1981.
33. Redwood, Richard, "Steel Detailing for Seismic Design Based on CAN/CSA-S16.1-M89", *1990 Canadian Society of Civil Engineering Annual Conference*, Hamilton, Ont., May 15-18, 1990.
34. Robinson, Guner S., *Proceedings WESTEX-86 IEEE Western Conference on Knowledge-Based Engineering and Expert Systems*, Anaheim, Ca, 1986.
35. Schmitke, C. and Kennedy D., *Effective Lengths of laterally Continuous, Laterally Unsupported Steel Beams*, Canadian Journal of Civil Engineering, 12(3), 1985.
36. Smith, Connie, *Performance Engineering of Software Systems*, Addison-Wesley, 1990.
37. Vancouver Structural Engineers Group, *Seismic Evaluation of Existing Buildings Seminar*, Vancouver, June, 1992.
38. Waterworth, John, *Multimedia Technology and Applications*, Ellis Horwood Limited, England, 1991.
39. Wigan, M.R., "Engineering Tools for Building Knowledge-Based Systems on Microsystems", *Microcomputers in Civil Engineering*, 1, July, 1986.
40. Wilson, Peter, "More Than Meets the Eye", *Saturday Review*, Vancouver Sun, Vancouver, B.C., 30 January 1993.
41. Winter, G., *Lateral Bracing of Columns and Beams*, ASCE Journal of the Structural division, 84(ST2), 1958.
42. Wood, B, Beaulieu, D., Adams, P., *Column Design of P Delta Method*, Journal of the Structural Division, ASCE, 102(ST2), 1976.
43. Wood, B, Beaulieu, D., Adams, P., *Further Aspects of Design by P Delta Method*, Journal of the Structural Division, ASCE, 102(ST3), 1976.