ISSUES IN REPRESENTATION AND PROCESSING OF STRUCTURAL DESIGN CODES OF PRACTICE

A.Neilson[1], B. Kumar[2] and I A MacLeod[3]

ABSTRACT: This paper presents a hybrid approach to standards processing. Standards processing is a term used the denote the procedures of designing and checking a design in accordance with the relevant code(s) of practice. The use of the conventional printed version of codes of practice is inherently fraught with problems. They are renowned for being voluminous, ambiguous in nature, and frequently full of cross references. Thus the extraction of relevant information is difficult. The strategy adopted for representing and processing codes of practice, discussed in this paper is aimed at alleviating these problems. Hypertext is employed as a form of representation, allowing the drawbacks of ambiguity, and cross referencing to be overcome. In addition a method of organising standards information is proposed aimed at aiding the retrieval of relevant information.

1 INTRODUCTION

The term code of practice is used to denote a set of regulatory, mandatory or obligatory rules for design. All structural designs must be processed and checked in accordance with the applicable code of practice. There have been several attempts in the past to automate this process of checking a design against a standard. Before the advent of knowledge based systems technology, the approach commonly adopted was to hard-code the provisions of the relevant standard into an application program. There are some limitations of using this approach, this method does not allow a great deal of flexibility for modifications being made in the future, when the represented code is superseded. Most recent attempts have centred around applications of knowledge-based systems, hypertext systems and object-oriented programming. The approach described in this paper is a hybrid one encompassing knowledge based, hypertext, standard numerical processing, and case-based reasoning (CBR) systems. A hypertext system MOSAIC, forms the central core of the proposed system from which the other systems hang and interact. A schematic diagram of the proposed approach is shown in Figure 1. The paper will initially discuss the generic problems associated with conventional printed codes and then describe the proposed system aimed at alleviating these problems.

[1] Research Student, Department of Civil Engineering, University of Strathclyde, Glasgow, G4ONG, U.K. e-mail:cmar70@strath.ac.uk.

[2] Lecturer, Department of Civil Engineering, University of Strathclyde. e-mail: B.Kumar@strath.ac.uk.

[3] Professor, Department of Civil Engineering, University of Strathclyde. e-mail: I.A.MacLeod@strath.ac.uk.

2 DRAWBACKS ASSOCIATED WITH STANDARDS

Unfortunately standards processing is hindered by the generic nature of standards, which are notoriously ambiguous and frequently full of cross references. Using conventional paper codes it is common for the user to become disoriented. As a result of being constantly referred to different parts of the same code, or even entirely separate documents, they become confused as to where they originally commenced. Moreover standards are often voluminous. The volume of information involved compels users to invest considerable time and effort in becoming accustomed with a particular code. Furthermore it is unusual for an engineer to use only one code in his career, therefore the process of familiarisation has to be repeated for each code used. Having achieved a certain level of efficiency with a particular code an engineer may find himself working on another project in which different codes are applicable. On returning to the code he was once familiar with he may find he has not retained the knowledge.

Before a design can be processed or checked the engineer must find the correct code(s). It is not uncommon for the entirely wrong code to be used. This problem, termed Code Applicability is due to misleading or incomplete scope statements.

In order to process and check designs it is necessary to locate only the parts of the code pertaining to the design in question, eliminating irrelevant material. This is not always a straight forward procedure. Due to the ambiguous nature of standards information provisions are commonly applied to the wrong context, or even ignored completely. As a result it is often difficult to locate all the provisions relevant to a particular context.

After the provisions have been retrieved the user is still faced with the task of interpreting the information. It is common knowledge that provisions are often misinterpreted, due to the ambiguous and misleading manner in which they are written.

An additional drawback is that codes are generally aimed towards "standard designs". The term standard design is used to denote more regular designs, for example, in the case of beams a standard design would be such as rectangular or flanged beams as opposed to trapezoidal or irregularly shaped beams which have been termed 'non-standard designs'. It is the general opinion that codes of practice lack sufficient information to allow non-standard designs to be developed proficiently.

3 AN ENVIRONMENT FOR STANDARDS PROCESSING

Figure 1 shows a schematic diagram of an environment for standards processing. It comprises of several individual components each of which plays a different role:

a. The Information Base.
b.. Organisation Of Standard Information.

c. Provision Retrieval.
d. Processing/Conformance checking.

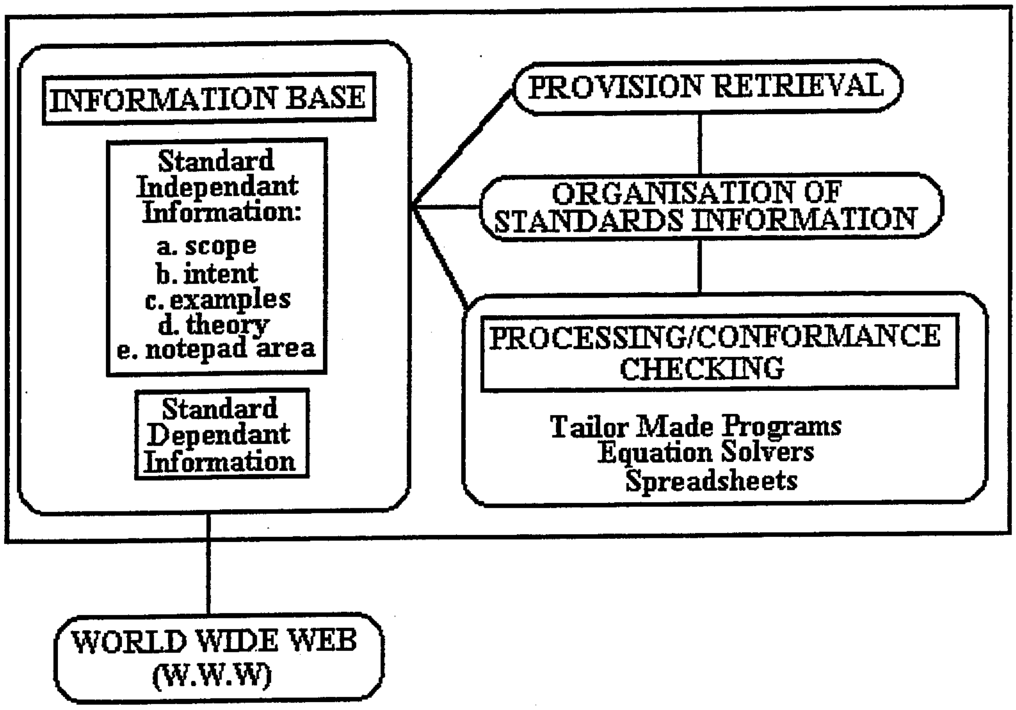

Figure 1. A Schematic Diagram of the Proposed Approach.

Each of these components are described in greater detail:

3.1 The Information Base

The Information Base contains all of the information contained in the conventional printed version of the standard (Standard Dependant Information) in addition to some supplementary information aimed at improving the understanding of the Standard Dependant Information (Standard Independent Information). Hypertext is employed for representing this information.

3.1.1 *Hypertext*

Hypertext can be defined as the creation and representation of interlinked discrete pieces of text. By implementing the standards information base in a hypertext environment many of the problems discussed previously can be relieved. Hypertext lends itself well to the representation of standards information. It possesses several attributes that allow the drawbacks traditionally associated with codes of practice to be overcome. Hypertext copes well with cross references, by creating "buttons" or "hotspots" to which the appropriate address is attached. The user can access the relevant information by simply clicking on the reference. This feature, in conjunction with the navigation facilities provided helps prevent the user from becoming disorientated.

Due to the linear nature of documents, differentiating relevant material from irrelevant material is often a time consuming process. The user is often required to review large amounts of information in order to extract the parts he wishes to read. Hypertext has an excellent user interface which allows the user to browse easily through the information, viewing only the parts that are of interest without encountering unnecessary information. In addition most hypertext environments incorporate a search facility which obviates this inconvenience further.

The problems of code familiarisation and ambiguity can only be overcome by providing supplementary information. This information which is described in detail in section 3.2.1, is aimed at improving the level of understanding in order to avoid misinterpretation, and aid the new/inexperienced user by reducing the time taken to reach an adequate level of perception.

The features remarked upon formerly merely describe the ability of hypertext to overcome the inherent problems associated with codes of practice. However further advantages can be attained as the hypertext environment can be used to provide the user with a rich set of tools. It is possible to access external processing software, such as spreadsheets. This aids the design and conformance checking processes.

3.1.2 *Components Of The Hypertext Information Base:*

The hypertext environment employed in the implementation of the prototype information base is MOSAIC, an interface to the World Wide Web (WWW). The information base consists of two main components: .

a. Standard Dependant Information - this consists of all the information from the code i.e. text, figures, tables, graphs etc. in electronic form.

b. Standard Independent Information - this section comprises of supplementary information aimed at improving the understanding of Standard Dependant Information. This information is discussed in detail in the following sections.

Standards Independent Information

The necessity for additional information:

The codes of practice are aimed at providing a 'design guide' to the engineer. Despite this the formal representation of a standard does not possess all of the knowledge required to actually perform a design task. Furthermore it is common knowledge that most codes of practice are not easy to interpret. Users find them ambiguous and there have been instances of incorrect usage of certain provisions from codes of practice leading to serious errors. On many occasions, it is found that some provisions of the design codes are misinterpreted or even ignored. The infamous Ronan Point collapse [reference] was attributed, in part, to poor interpretation of a code of practice. In order to overcome this it is necessary to address the reasons why this misinterpretation occurs. It is thought that the reasons for this could possibly be one or more of the following:

a. the provision in question is poorly described/explained or indeed not given in the detail which would be expected.

b. the engineer in question is not experienced or familiar enough with the code to interpret it properly.

Types of information that should be provided:

In order to overcome this misinterpretation and indeed the wasted time as a result of it, a commentary has been attached to each of the provisions which require it, by means of a hypertext link. The objective is to improve the clarity of the standards information. The commentary includes:

a. Scope - where it is unclear from the text of the provision what design context it applies to, a short explanation can be included. This is aimed at preventing provisions from being employed in the wrong design context. Scope has been provided at three levels, firstly for the whole code, to aid users decide if they are applying the correct code to the problem. It is provided again at each section i.e. 3.4 Beams, and at provision level.

b. Theoretical Description - a short explanation of the theory behind the provision is intended to serve multiple purposes as detailed below:

(i) Help clarify an ambiguous provision, the content of a provision cannot be fully understood unless the theory behind it is not. "Detailed guidance is available on the general principles of design. But engineers have to interpret this guidance properly. This requires a thorough understanding of the principles of design as well as their practical implications". (Kumar & Topping 1988) Supplying a theoretical explanation promotes a thorough understanding of the design principles.

(ii) In the case where the provision is not detailed perhaps the explanation will allow the engineer to realise what is intended.

c. Intent Of The Provision - sometimes the intention of the provision becomes lost in the words describing it. Therefore it would be useful if a description were provided in simple terms, of the actual intent of the provision. If the code user is aware of the aim of the provision then no problems of misinterpretation should occur.

d. Examples Of Use - by providing examples of the proper use of the provisions it is hoped this will provide an explanation of the practical implications of the provision.

e. Notepad Area - it may be the case that a user wishes to make a note of some observations he has made upon using certain provisions. An area should be provided to allow the user to do so.

3.2 Organisation of Standards Information

The ultimate aim of this organisation is to overcome the problem associated with retrieving relevant information succinctly from standards documentation. In addition the organisational system described allows the applicability of the code to be checked. Information cannot be retrieved effectively if it is not organised efficiently. An efficient organisational system allows the relevant provisions to be retrieved quickly and accurately. Considerable work has been carried out in this area (Fenves 1987, Harris 1980). In Fenves 1987, Garrett 1989, 1992, 1994, and Harris 1980 the information is classified using classifiers. This involves extracting a word or word(s) which concisely describe the scope of the provision. In order to retrieve the required parts the user is required to describe the design context in terms of the classifiers.

Extracting classifiers is generally a relatively straight forward process. However this is not always the case. In some instances it is impossible to extract classifiers which describe the scope of the provision accurately. In general it can be said that two types of provisions occur, those which apply to the design of standard elements, for example rectangular beams and columns, and those which apply to the design of non standard elements i.e. irregularly shaped beams and columns. Frequently the provisions which apply to non-standard designs are general in nature, and can often, also be used for designing standard elements. Classifiers can easily and accurately be extracted from provisions with a standard context. However due to the general nature of non-standard provisions it is difficult to describe the context using only a few words. The fact that the context they actually apply to has no distinct bounds makes it impossible to account for every design possibility. In conclusion it can be said that classifiers cannot describe the subject matter of such provisions efficiently, and cannot therefore, be relied upon as a method of retrieving all provisions.

Based on the previous statements a suitable proposal for organising standards information for retrieval is proposed here:

1. Distinguish provisions which apply to standard design only from those which are applicable to non-standard as well as those which are relevant to both non-standard and standard designs.

2. For the provisions applying purely to standard designs, the classifiers can be extracted and arranged in a suitable classification scheme.

3. Provisions which are applicable to non-standard designs will be implemented in a search type retrieval system (discussed later). For provisions which contain both a standard and non-standard context, these contexts will be separated. The standard context being classified, and the non-standard being implemented in the search type mechanism along with the other non-standard cases.

The reasoning behind this proposal is that provisions applying to non-standard designs cannot be described precisely by classifiers. On the other hand it would be inefficient to repeatedly search for a standard set of provisions. Whilst classifiers cannot easily describe every provision they can be easily extracted from those describing a standard design procedure.

3.2.1 Elements Of The Proposed Organisation Of Standard Information

In [Garrett '94] the drawbacks of existing classification systems are stated and a model aimed at overcoming these problems presented. In order to improve upon existing methods, proposed organisational systems should overcome these problems, which are:

a. An excessive number of subclauses are required to represent the information.

b. Redundancy within the network.

c. Lack of automatic instance classification.

Every organisational system from the references mentioned previously, classifies initially in terms of design objects. [Fenves '86(b)] states "The classification system is constructed by merging the facets in the following order : object-type, stress-state, limit-state, object-composition. Although this ordering was independently developed for the purpose of this study, it is the same as that found by [Nyman 73] for the most logical organisation of the text of the standard". From this statement it is clear that this is generally accepted as the correct order. Most probably because it restricts provisions relevant to specific contexts to one area of the network.

Standards are concerned with a number of general topics, for example definitions, loading, analysis, general design guidelines/procedures, design limitations. The proposed classification system separates these general design topics, which are referred to as issues, from the design objects and classifies each on a separate hierarchy. A hierarchy is constructed for Standard's Issues known as the Issues Hierarchy, inaddition separate hierarchies are constructed for each general design object found in the code(s) i.e. beam, column, slab etc. Each of these hierarchies is known as a Code Entity Hierarchy. An example of an Issues Hierarchy is shown in Figure 2a. Figure 2b shows and example of a Code Entity Hierarchy for beam. As a Code Entity Hierarchy describes the design contexts covered by the code this allows the code applicability to be checked. This is simply a case of comparing the object being designed termed the "Design Entity" with the Code Entity Hierarchy.

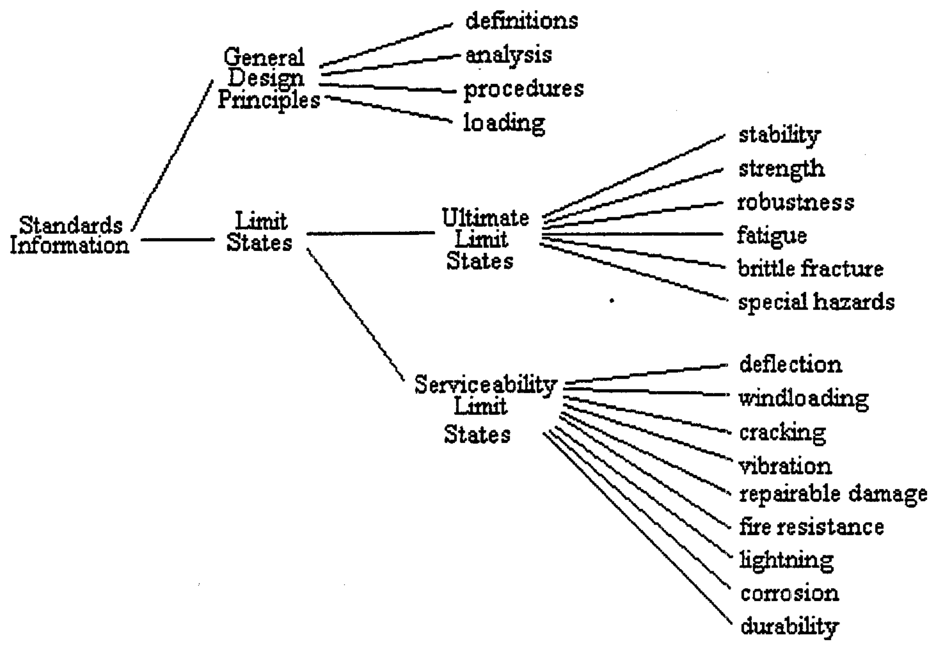

Figure 2a. An Example of an Issues Hierarchy.

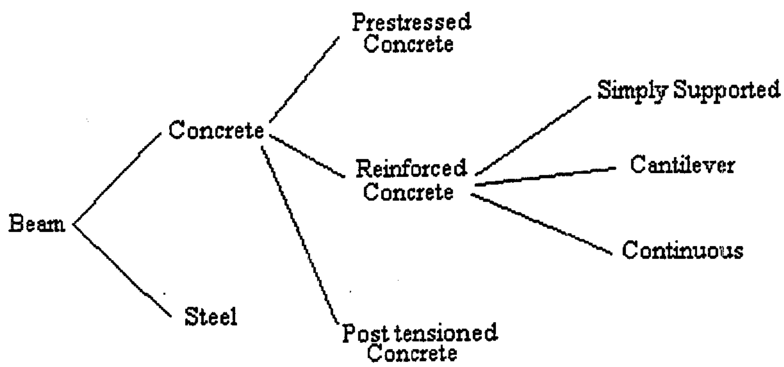

Figure 2b. An Example of a Code Entity for Beam.

Relevant provisions are stored under the respective nodes of each hierarchy. Inaddition nodes from the Issues Hierarchy can be linked to nodes from a Code Entity Hierarchy and provisions pertinent to this combination can be stored accordingly.

The advantages of employing this classification method to standards information are:

a. A large majority of the issues represented on the Issues Hierarchy are common to all structural design codes. Design Entity Hierarchies will also vary little from code to code. Subsequently this method of classification will permit the combination of several design standards with minimal amendments. Arguably, it is also possible to combine standards if design objects are found at the top level of the classification tree. However the Issues discussed are more commonly shared between standards than design objects, which are more likely to vary from standard to standard.

b. Structural design practice is to design for Ultimate Limit State (ULS) and check against the Serviceability Limit State (SLS). The design limitations found in any structural design code fall under one of these categories. The proposed method will result in the separation of clauses referring to ULS and SLS. In contrast if design objects are classified initially on a single hierarchy for design objects and design limitations the ULS and SLS clauses will become compound. It is preferable for design processing and conformance checking purposes that ULS and SLS clauses are separated.

c. Further standards can be easily added by the 'individual' user thus taking account of the fact that different design offices use different combinations of standards.

d. If the standard is updated it is unlikely to affect this organisational system, simply because the fundamental issues covered in structural standards are unlikely to alter in the future.

e. The classification system allows Code Applicability Checking to be carried out. This is possible by comparing the Design Entity with the Code Entity Hierarchy.

f. This classification method offers a great deal of flexibility. It can be employed for retrieving provisions of a general or specific nature. For example a standards user may wish to retrieve all of the provisions relevant to a specific design entity i.e. all provisions applicable to beams. If this is the case these provisions can be extracted easily from the Code Entity Hierarchy. This also applies to issues. All provisions which refer to a specific issue can be derived simply from the Issues Hierarchy. On the other hand it may be necessary to extract provisions relevant to a more exact design context, i.e. lateral torsional buckling of simply-supported rectangular beams. It is possible to establish provisions applicable to such a specific context due to the hierarchical linking system described formerly. This flexibility allows users of varying experience and demands to use it effectively.

g. Problems associated with existing classification systems are surmounted. The fact that design objects are represented by individual Code Entity Hierarchies, which can be linked to the Issues Hierarchy eliminates repetitiveness and also reduces the number of subclasses considerably. Redundancy is diminished to an agreeable level, however this problem is inherent to the character of standards provisions and cannot therefore be eliminated fully. In addition, the lack of automatic instance classification expressed with existing schemes is dealt with.

3.3 Provision Retrieval

The process of provision retrieval is highly dependant upon the classification system. An alternative to classification is to employ a search mechanism to retrieve the relevant provisions [Vanier 1991]. This involves describing the design context in terms of a few pertinent keywords. In response, a list of the provisions containing occurrences of the keywords is given. This process can be improved by incorporating a thesaurus. Therefore the search mechanism will not only retrieve the provisions containing the keywords but also those which contain words with the same or similar meanings to the keyword. The problem that exists with this technique is that it is difficult to perform the search to the required extent, the search may be too elementary, resulting in the elimination of some relevant provisions. On the other hand, it could also be carried out at too great a depth, the outcome being that the search mechanism retrieves a large number of provisions, many of them irrelevant. Consequently search mechanisms can only be used as an aid to guide the user to the correct area, leaving him to make his own decision as to which provisions are relevant and which are not.

As stated previously, a search mechanism cannot be relied upon as a stand alone provision retrieval system, even with the incorporation of a thesaurus. CBRExpress is a case-based reasoning tool, used for storage and retrieval of cases. Although a search mechanism forms the basis of the retrieval process it is not the sole retrieval mechanism. It accepts a natural language statement as its initial input and then presents a menu of questions to the user to elicit more specific information about the desired object, thus making CBRExpress particularly strong at finding cases that the operator cannot easily describe. Each provision can be represented as a case in CBRExpress, this allows each provision to be described by as many words as is necessary. The fact that the user can describe his design context in natural language, rather than being restricted to keywords is advantageous. CBRExpress employs a selection of sophisticated matching algorithms to find all cases that are similar but not necessarily identical to the search description thus the previously discussed reliability problem is eliminated.

3.4 Processing And Conformance Checking:

Once the relevant provisions or requirements have been established it is possible to carry out design processing and conformance checking. For this a variety of software packages will be employed including spreadsheets such as Microsoft Excel, Equation Solvers such as TKSolver, and Tailor Made Programs for specific procedures, which may be written in any suitable programming language. This processes is facilitated by the standard independent information contained in the information base. Moreover it is planned to store examples of existing non-standard designs in the case-based reasoning system aimed at aiding the processing of new non-standard designs.

3.5 World Wide Web (WWW)

Each of the components described in the preceding paragraphs collectively forms the Standards Processing Environment. From Figure 1 it can be seen that the environment can be accessed from the WWW furthermore any information on the WWW can be accessed by users of the system.

4 CONCLUSION

This paper presented an overview of the drawbacks commonly associated with the use of the conventional printed version of the codes of practice. It discussed a proposed standards processing environment comprising of four main components

a. The Information Base.
b. Organisation Of Standard Information.
c. Provision Retrieval.
d. Processing/Conformance checking.

Collectively these elements are aimed at aiding standards processing by alleviating the discussed problems. Following is a summary of the four components.

The Information Base provides a variety of additional information which will make the code of practice much more transparent to its users and aid the problems of ambiguity and misinterpretation. This will therefore aid the procedures of **processing and conformance checking.**

The Organisational System is intended to improve the accessibility of standards information. It is concluded that not all of the provisions of the code lend themselves well to classification. Therefore a classification system cannot be relied upon as the sole means of provision retrieval. A classification scheme is proposed for those provisions amenable to it.

Provision Retrieval is highly dependant upon the organisational system. An alternative method of provision retrieval is suggested. Efficient accurate provision retrieval is essential to both design **processing and conformance checking.**

5 REFERENCES

Chung, P.W.H., & D. Stone. 1993. Handling Regulatory Information Using Information Technology. *Development of Expert Systems for Structural Mechanics and Structural Engineering*.

Fenves, S.J. & J.H.Garrett 1986(a). Knowledge Based Standards Processing. *Artificial Intelligence In Engineering*. Computational Mechanics Publications, Southhampton. Vol. 1, No. 1.

Fenves,S.J. & J.H.Garrett 1986(b). A Knowledge Based Standards Processor for Structural Component Design. *Report #R-86-157, Dept. Of Civil Engineering, Carnegie Mellon University*, Pittsburgh, Pennsylvania.

Fenves, S.J., & R.N.Wright, F.I. Stahl, and K.A.Reed 1987. Introduction To S.A.S.E : Standards Analysis Synthesis and Expression. *Technical Report, NBSIR 873513, National Bureau of Standards*, Washington, D.C.

Garrett, J.H. 1989. Object-Oriented Representation Of Design Standards. *IABSE Colloquium Bergamo* - Expert Systems in Civil Engineering.

Garrett, J.H. & M.M.Hakim. 1992. Object-Oriented Model of Engineering Design Standards." *Journal Of Computing in Civil Engineering*. ASCE, N.Y. Vol.4. No.3, July, 323 - 347.

Garrett, J.H. 1994. The Context-Oriented Model : An Improved Approach For Representing And Processing Design Standards. *Computing in Civil Engineering*. ASCE, N.Y. Vol. 1. 145 - 152.

Hakim, M.M. & J.Garrett 1992. Issues in Modelling and Processing Design Standards. *Computers and Building Standards Workshop*, University of Montreal, Montreal, Canada.

Harris, & Wright. 1980. Organisation of Building Standards; Systematic Techniques for Scope and Arrangement. *Building Sc. Series, NBS BSS 136, Nat.Bureau of Standards*, Washington, D.C.

Kumar, B & B.H.V. Topping 1988. Issues in The Development of a Knowledge-Based System for the Detailed Design of Structures, *Artificial Intelligence in Engineering Design*, Ed. J.S.Gero, Elsevier and CMP.295-314.

Morris, L.J. 1981. A Commentary on Portal Frame Design, The Structural Engineer, Volume 59A, No.12, *Journal of the Institution of Structural Engineers*, London, December 1981, 394-404.

Nyman, D.J. and S.J.Fenves. 1973. An Organisational Model For Design Specifications, *Technical Report R73-4, Dept. of Civil Engineering, Carnegie Mellon University*, Pittsburgh, Pennsylvania.

Discussion - A Commentary on Portal Frame Design, The Structural Engineer, Volume 61A, No.6, *Journal of the Institution of Structural Engineers*, London, June 1983, 181-189.

Rosenman, M.A., & J.S. Gero. 1985. Design Codes as Expert Systems. *Computer-Aided Design*. Butterworth & Co. Vol. 17. No. 19. 399-409

Topping, B.H.V, & B. Kumar. 1989. Knowledge Representation and Processing for Structural Engineering Design Codes. *Engineering Applications of AI*. Pineridge Press Vol. 2,. 214 - 227.

Vanier, D.J. 1991. A Parsimonious Classification System to Extract Project-Specific Building Codes. *VTT Symposium 125. Computers and Building Regulations*. Espoo, Finland, 1991. 134 - 145.

Vanier, D.J. 1995. MiniCode Generator: A Practical Research Application for Standards Processing. *IABSE Colloquium Bergamo*.

Yabuki, N., & K. Law. Hyperdocument Model For Design Standards Documentation. *Journal of Computing In Civil Engineering*. ASCE, N.Y. Vol. 7, No. 2, April. 218-237.

British Standards Institute *(B.S.I)*, BS 8110 Structural Use of Concrete.

British Standards Institute *(B.S.I)*, BS 5950 Structural Use of Steel.

British Standards Institute *(B.S.I)*, BS 6399 Design Loading for Buildings.

British Standards Institute *(B.S.I)*, BS 5400 Steel, Concrete and Composite Bridges.