A Design Tool with Integrated Knowledge and Code Checking for Concrete Beams

Nicolaas Stuurstraat
Assoc. Prof. of Delft University of Technology, Faculty of Civil Engineering,
P.O. Box 5048, 2600 GA Delft, The Netherlands

Abstract

Knowledge-based expert system technology has been applied most successfully to diagnosis problems. Expert systems have also been developed for fault detection, prediction, interpretation, monitoring and planning. Despite their useful applicability in the early design stage of the structural design process there is a poor number of expert systems developed for the structural engineer (fig. 1).

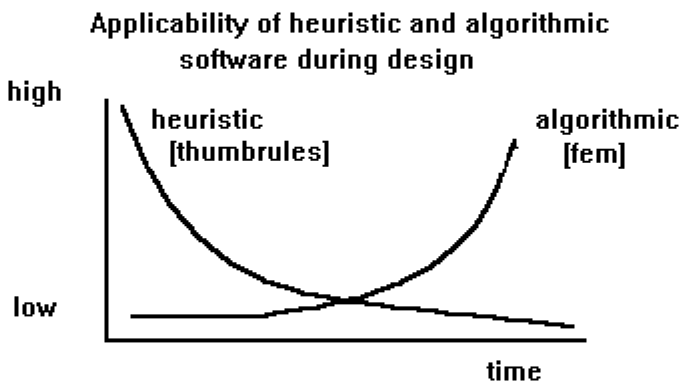

Figure 1 - Applicability of heuristic and algorithmic software during design

An expert system developed for structural design problems exhibits certain similarities to diagnostic expert systems. The results from the calculations have to be analysed and depending on the diagnosis certain actions can be taken or advises can be given. For solving a design problem the expert system will first of all have the task of finding an initial design configuration using heuristic knowledge based on rules of thumb and default values. Numeric algorithmic procedures are then applied to determine the performance of the design. The design process can therefore be thought of as being subdivided into a design part and a diagnostic/analytic part. The problem of structural design can in general be characterised as a search for a solution amongst a large number of alternatives that all satisfy the specified constraints.

Introduction

In 1995 the Dutch designers had to switch over from the old concrete code (VB74) to totally different and new regulations for concrete, the VBC 1995 (NEN 6720). Especially the complete design of partial prestressed concrete beams is difficult and has become a time consuming work. An integrated design expert system that consists of a combination of a heuristic rule-based system interacting with a code checking system based on algorithmic calculations can therefore help the structural engineer in designing partial prestressed concrete beams. The requirements in the codes form an important part in the search for a solution to a structural design problem. The codes set constraints in the lines of reasoning and thereby they limit the search space, reducing the number of design alternatives. When the design configuration is completed the codes are applied in order to demonstrate that the design complies with all applicable requirements. Conventional programming can solve the problem of calculating the stresses and strains of a structure with a given geometry and analyse the results. The design process, however, is an inverse analysis, i.e., the strains and stresses are given (supplied by the codes) and a structural configuration is devised exhibiting these properties. An important part of developing an expert system for solving design problems is therefore how to interpret and apply the requirements as constraints in trying to reach the design goal. Because the codes leave much of the decisions to the designer, it is not possible to arrive at a solution using the codes alone.

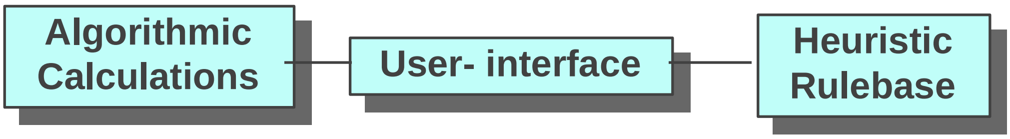

Figure 2 - The integrated design system

An adequate expert system will therefor apply heuristic rules combined with the constraints in the codes to guide the search towards the design goal. A prototype integrated design system that can help to design a complicated structure in a very short time has been developed.

Building the prototype system

One of the first steps in developing an expert system is the selection of a suitable programming environment or expert system shell. Prior to this a preliminary analysis of the problem to be solved is necessary to resolve which features and capabilities are desired in the programming environment. This step is not included in the development of the prototype design system. The programming environment was chosen beforehand and the reasons for this are availability but also for a little research into the possibilities of the environment. The available tool was KnowledgePro, a programming environment under the Microsoft Windows operating system. It is not a true expert system shell but an object oriented programming environment with a few features found in standard expert system shells. Applications developed in this environment will have the same graphical interface as other Windows programs. The appearance of and how to interact with KnowledgePro applications will therefor be familiar to Windows users. The structure of a KnowledgePro program can be visualised as a tree illustrated in figure 3. The main topic is the application program that contains all other topics. Topics can be compared with procedures in PASCAL. However, the term topic has a much broader meaning. In addition to structuring the program into separate components each fulfilling a specific task (like a procedure) a topic can represent a variable, a function, an object or can be used to hold a set of production rules.

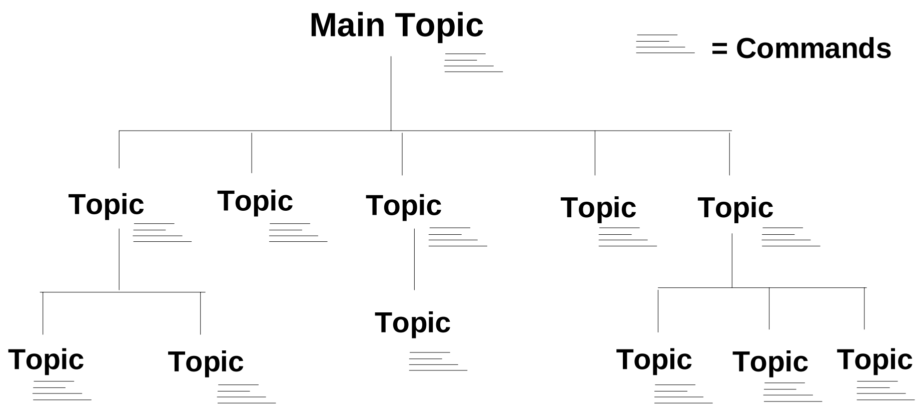

Figure 3 - The structure of a KnowledgePro program

Other features that deserve mentioning are those used for knowledge representation, the inference mechanism, user interface and the interface to external programs; namely:

- production rules and objects
- backward chaining
- hypertext, hyperregion and MS Windows user interface
- dynamic data exchange

Production rules and objects

The representation of the knowledge follows the rulebase approach and involves the creation of a collection IF-THEN statements in the same way as in other expert system shells (fig. 4).

```
topic profiel_keuze.   (* the rules to find the best type of girder cross section *)
set_number_of_values(profiel_keuze,1).

  if not(?vloerfunctie) and  ?M_q/?M_g > 2 and
      (?Balk:l >=10 or ?Balk:hoogte_schatting >=250) then profiel_keuze = 'I-profiel'.

  if not(?vloerfunctie) and (?Balk:l =<10 or ?Balk:hoogte_schatting<250) then
         profiel_keuze = 'Rechthoekig profiel'.

  if  ?Balk:hoogte_schatting > 400 and (?vloerfunctie
     or (?M_q/?M_g < 2 and ?Balk:l >= 10)) then profiel_keuze = 'T-profiel'.

  if ?vloerfunctie and ?Balk:hoogte_schatting =<400 then profiel_keuze is 'Plaat'.

end.
```

Figure 4 - A part of the rulebase

Another method for structuring knowledge is through the use of objects. This can be done in KnowledgePro applying its object oriented programming features. An example of an object class hierarchy is given in figure 5. In this example the simply supported beam object has three sub-classes which inherit the properties of simply supported beam. This could also have been achieved using rules but it would require more programming and it would not exhibit the same logical structure.

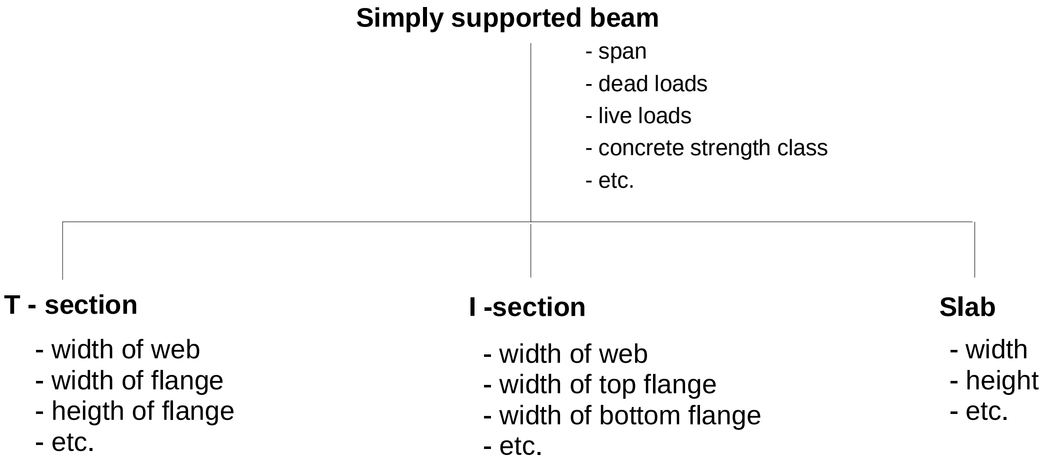

Figure 5 - An object class hierarchy

Dynamic data exchange

To build an integrated expert system there has to be a link of the symbolic processes with an external numeric program. Therefor we must have an interface controlling the manner by which data is passed between them. In MS Windows operating system applications can communicate with each other through a protocol called Dynamic Data Exchange (DDE). KnowledgePro also supports DDE making it possible to develop applications (the client) which can initiate conversations with other applications (the server) through the opening of a DDE channel between them. The client can then send commands and data to the server application using a syntax known by the server. A spreadsheet model (Excel) was chosen to contain the numeric procedures. One of the advantages of developing a spreadsheet model is a clear and comprehensible manner without the need for any complex programming routines.

The Design Expert System with Integrated Knowledge

The design of a simply supported concrete beam generally involves the following steps:

1. Choosing the section shape that most effectively can carry the prescribed loading and perform additional structural functions.
2. Choosing the section dimensions.
3. Calculating the necessary reinforcing and prestressing steel following the requirements in the codes.
4. If needed adjusting the section dimensions.
5. Checking for conformance with all requirements.
6. If necessary repeat steps 4 and 5.

The knowledge needed to assist in the decisions a designer has to make in each of these steps is structured in the knowledge base using a combination of rules and objects with backward chaining as the inference mechanism. The rules are primarily used in determining which section shape to use (fig. 6) and for giving advice to the user in the design process. The object class hierarchy facilitates the collection of knowledge at different levels. At the top level is an object class *beam* with attributes attached concerning the span, the loading, the material qualities used and the height of the section. For each section type a sub-class of *beam* exists with attributes specific to the type of section.

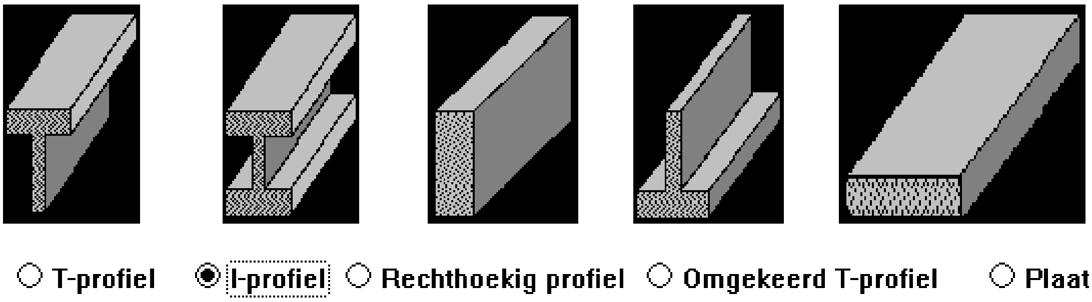

Figure 6 - Different types of section (sub-classes of the object beam)

The specifications of the initial design configuration are sent via the DDE channel to the spreadsheet model. In this model the numeric procedures are executed and the results are returned to the main program. The results include the minimum reinforcing and prestressing steel needed to comply with the requirements in the ultimate limit state for bending and longitudinal force (maximum downward loading). Other information retrieved are the results from the crack control requirement and the limit state of deformation (fig. 7). During the iterative design process the user can ask for advice in interpreting the results and suggestions to improve the design.

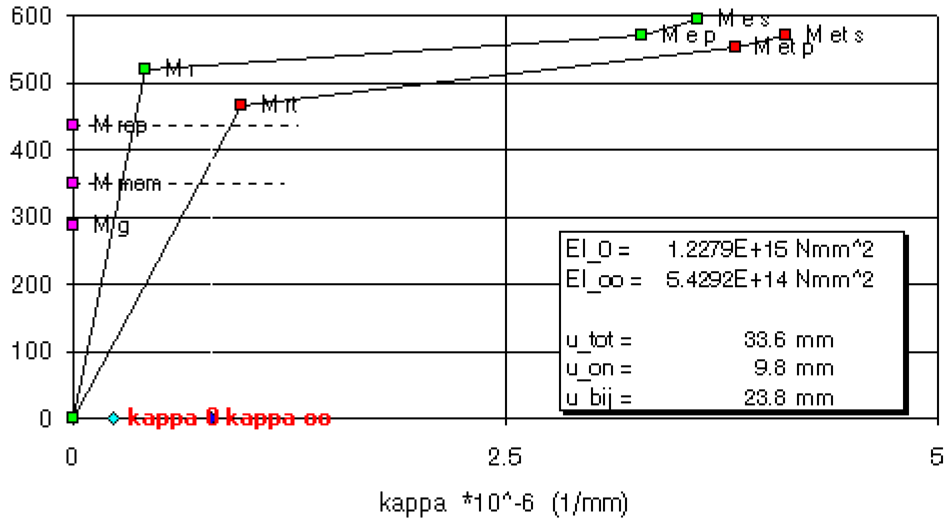

Figure 7 - Chart of the Moment-Curvature relation of a partial prestressed beam

Conclusions

- The importance of the selection of the best design on the one hand and the available knowledge on the other during the preliminary design process asks after a different kind of software than usual at present.

- Development of design tools in civil engineering is becoming more and more easier for the practising engineer. Useful tools working in the MS-Windows environment can help to build up some experience in this specific field.

- An integrated design system that consist of a combination of a rule-based system interacting with a system based on algorithmic calculations is more suitable in assisting the structural engineer in designing concrete beams.

References

1. Williamson M., 1988. Knowledge Pro Disperses Complex Data Efficiently, PC Week, the national Newspaper of corporate microcomputing.

2. H. Adeli, K.V. Balasubramanyam, 1988. Expert Systems for Structural Design, Prentice Hall, New Jersey.

3. T.O. Barnwell, J.H. Garrett, 1989. An Object Oriented Representation of Design Standards, Computing in Civil Engineering.

4. R.D. Coyne, 1989. Knowledge-Based Design Systems, Addison Wesley, Reading.