AN EXPERT SYSTEM FOR CHOOSING THE TYPE OF READY MIX CONCRETE

Karl-Johan Serén
Technical Research Centre of Finland
MScTech, Research Scientist

Synopsis: Expert systems offer some advantages compared to traditional algorithmic programs, especially in the construction industry where a lot of heuristic knowledge is used. An expert system prototype intended to be utilized as a decision support tool for the building site personnel in choosing ready mix concrete is described. The system, developed in the Technical Research Centre of Finland, runs on IBM PC/XT/AT and compatible microcomputers.

Keywords: Computer programs, artificial intelligence, expert systems, ready mix concrete.

1 INTRODUCTION

Expert system technology, a spin-off of the artificial intelligence (AI) research, offers a new way of solving various problems related to construction. It provides a new tool, symbolic reasoning, for handling information, which previously has not been accessible to traditional algorithmic computing. Expert systems can handle heuristics, facts based on judgment, experience, rules of thumb, intuition and other expertise to provide knowledgeable advice about a variety of tasks. In the construction industry there are exceptionally many problems which are solved on the basis of experience and judgment without strict mathematical models. The implication of this is, that there is a vast range of potential applications for expert systems in the construction industry /7, 12, 13/.

A collection of small expert system prototypes for various applications related to construction has been developed in the Technical Research Centre of Finland (VTT). The initial experiences gained from this work are accounted for in /11/. One of these systems is the system described in this paper, an expert system for choosing the type of ready mix concrete (**BETVAL**) /10/, which has been developed by the Concrete and Silicate Laboratory in cooperation with the Laboratory or Urban Planning and Building Design in VTT.

The characteristics of the concrete mix have a great effect on the success of concreting and the quality of the resulting hardened concrete structure. With modern technology, it is possible to manufacture ready mix concrete of many types for different uses and environments. This versatility sets high demands on the site foreman's skill in choosing the right type of fresh concrete for each purpose. This is where expert system technology can be of great help.

The expert system described is intended to be utilized as a decision support system for building site personnel in choosing the type of fresh concrete to be ordered from the ready mix concrete plant. At present, the system is a prototype and it cannot be used in production as such due to its somewhat limited knowledge base (the knowledge base does not contain any information about special cases or economical aspects).

A similar kind of expert system has been under development in the National Bureau of Standards (NBS) in USA /4/. The system, DURCON, is slightly different from the system developed at VTT. DURCON is intended to be used by the designers as an aid in choosing the constituents for durable concrete in some special cases.

2 GENERAL ABOUT EXPERT SYSTEMS

What is an expert system? Feigenbaum, a pioneer in expert systems, has stated /6/:

> "An expert system is an intelligent computer program that uses knowledge and inference procedures to solve problems that are difficult enough to require significant human expertise for their solution. The knowledge necessary to perform at such level, plus the inference used, can be thought of as a model of the best expertise of the best practitioners of the field."

The difference between most traditional computer programs and expert systems is described in fig. 1. The main feature of expert systems is that, in contrast to ordinary algorithmic computer programs, the knowledge is separated from the search and control system into a knowledge base. The main parts of an expert system are (see fig. 2):

1) A **knowledge base**, containing facts and heuristics (rule-of-thumb type of knowledge) of a specific problem domain. In most systems the knowledge is represented in the form of IF-THEN-ELSE rules (often called production rules). Other forms of knowledge representation are semantic networks and frames (see for instance /3/ for further information).

2) An inference system or **inference engine** for utilizing the knowledge base in the solution of the problem. The inference engine contains the control strategy for searching through the knowledge base. The two main control strategies for rule-based systems are forward-chaining and backward-chaining.

3) A **working memory** (global data base, workspace) for keeping track of the problem status, the input data and the relevant history of what has this far been done.

4) An **explanation subsystem**, which is one of the key features of expert systems. Expert systems are usually able, to some extent, to explain why a certain piece of data is asked from the user or how the system has come to the conclusions presented to the user.

5) A **user interface**, which provides the means for the user to communicate with the system. Usually the user communicates with the system through menus generated by the system. Some newer expert systems provide the user with a multi-window interface, which can be manipulated with a mouse.

6) A **knowledge acquisition system** for adding knowledge to the knowledge base. Usually this subsystem consist of a rule editor or screen editor. Some systems have the facility to induce the knowledge from examples given by the system developer.

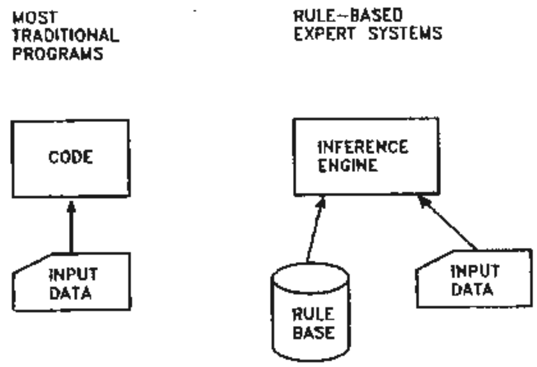


Fig. 1. The difference between traditional computer programs and rule-based expert systems /5/.

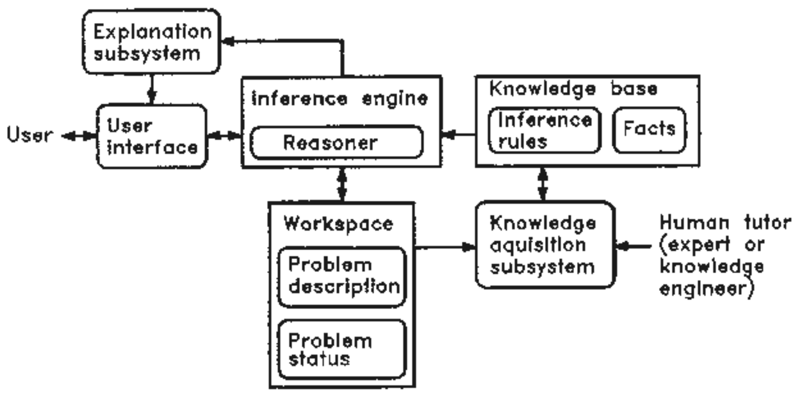


Fig. 2. The main parts of a rule-based expert system /8/.

The process for developing expert systems is different from ordinary computer systems development. Expert systems are built iteratively. This means that a small preliminary prototype is increased and updated gradually by extensive testing. This is usually done by a system developer (often called a knowledge engineer) in cooperation with the expert, who is providing the knowledge for a specific application. The prototyping system development approach is described in fig 3.

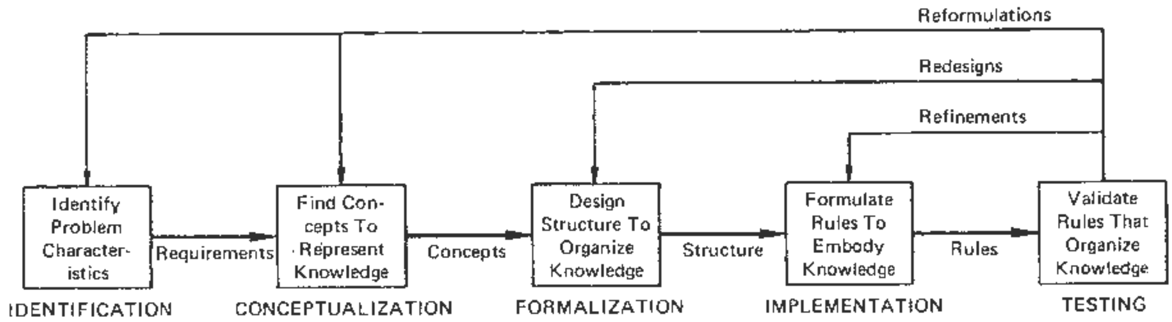


Fig. 3. Stages of expert system development /7/.

Many kinds of software development tools for expert systems are available: symbolic languages (LISP, PROLOG), knowledge representation languages (OPS5, POPLOG), hybrid environments (ART, KEE, Knowledge Craft) and empty expert system shells. Conventional languages (Pascal, Fortran, C etc.) can be used, although they are not as well suited for symbolic processing. Perhaps the most convenient way of developing small expert systems is to use empty expert system shells, which readily contain all other parts of the system except the knowledge base itself. A great variety of this kind of tools is available for microcomputers.

3 BETVAL, AN EXPERT SYSTEM FOR CHOOSING READY MIX CONCRETE

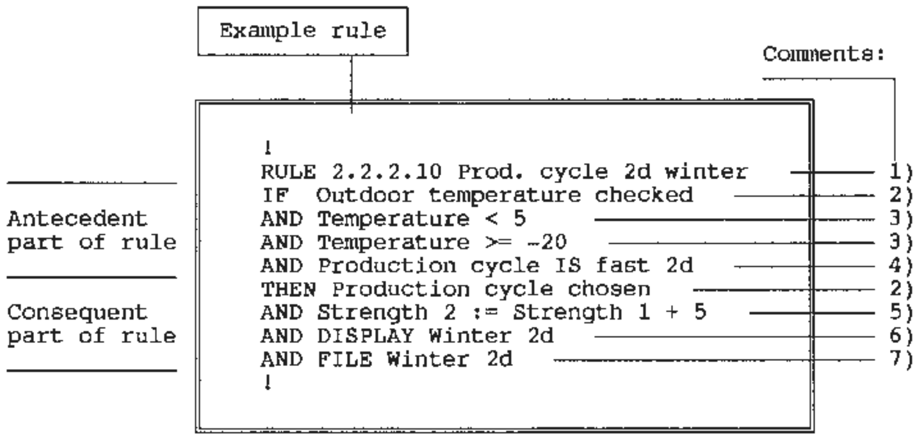
3.1 The expert system shell

The software tool used for developing the BETVAL system is a commercially available expert system shell called Insight 2+. The system runs on IBM PC/XT/AT and compatible microcomputers under PC-DOS 2.0 or MS-DOS 2.11 or later operating system versions. The recommended configuration is at least 512 kbytes RAM and a hard-disk drive /1, 2/.

The knowledge is represented in the form of productions (IF-THEN-ELSE -rules), which are formed of statements or facts bound together by logical operators (AND, OR and NOT). There are four fact types: simple facts, object-attribute facts, numeric facts and string facts. An example of a rule is shown in fig. 4.

The inference system is mainly goal-driven (backward-chaining), although some simple inference control statements can be used to achieve a forward-chaining like function. The Insight 2+ system includes a sub-set of Pascal, a full screen editor, a dBase II and III compatible database editor and functions to call procedures written in other languages.

The knowledge bases are developed by writing the source code in the knowledge representation language and by thereafter compiling the knowledge bases into an inner representation form, which the inference system can use during a consultation. The inference system automatically generates menu-type queries when it finds a statement with an unknown value during the search through the rules. Textual information can be attached to the facts to give a more finished appearance to the user interface.



Comments: 1) Rule title; does not affect the inference
2) Simple fact (or statement)
3) Numerical fact with relational comparison
4) Object-attribute fact
5) Numerical fact with assignment operation
6) Control element for information text display
7) Control element for saving text display in file

Fig. 4. Example of a rule written in the knowledge representation language of Insight 2+.

3.2 The system architecture

In Finland the building site personnel has to inform the ready mix plant about the following properties when ordering fresh concrete:

1) The compressive strength class (defined by the designer), taking into consideration appropriate concreting techniques.
2) The consistency value of the fresh concrete (usually defined as VeBe-time or sVB).
3) The maximum size of aggregate.

Accordingly, the domain is divided into three sub-problems or contexts (fig. 5), each of which is a typical classification problem, i.e. the system has to choose from a number of pre-defined solutions. A production system is a natural way of solving problems like this /14/.

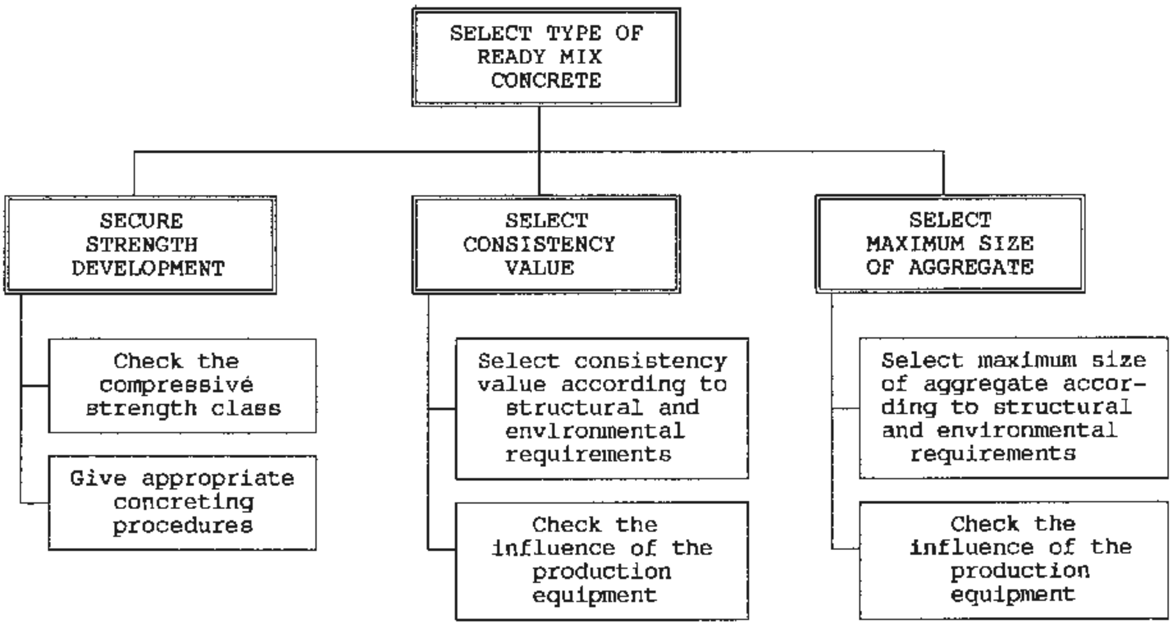


Fig. 5. The general structure of the BETVAL knowledge base. The problem domain is divided into three main contexts.

3.2.1 Securing the strength development

The first context contains knowledge about the compressive strength class and about appropriate concreting techniques (fig. 6). First the system checks the required minimum value for the strength class and compares it to the strength class defined by the designer. If the strength class given by the designer is too low the system gives a warning about this. The system queries following fact values from the user for this purpose:

- environment class,
- water impermeability requirements,
- frost proof requirements,
- corrosion proof requirements and type of corrosive environment and
- the compressive strength class given by the designer.

Secondly the system gives recommendations about appropriate concreting techniques, such as curing, heating and heat treatment; increasing the strength class for the ordered fresh concrete in some cases; some general information about the use of admixtures. The fact values queried from the user for this purpose are

- the outdoor temperature at the building site and
- the desired production cycle time (form stripping time).

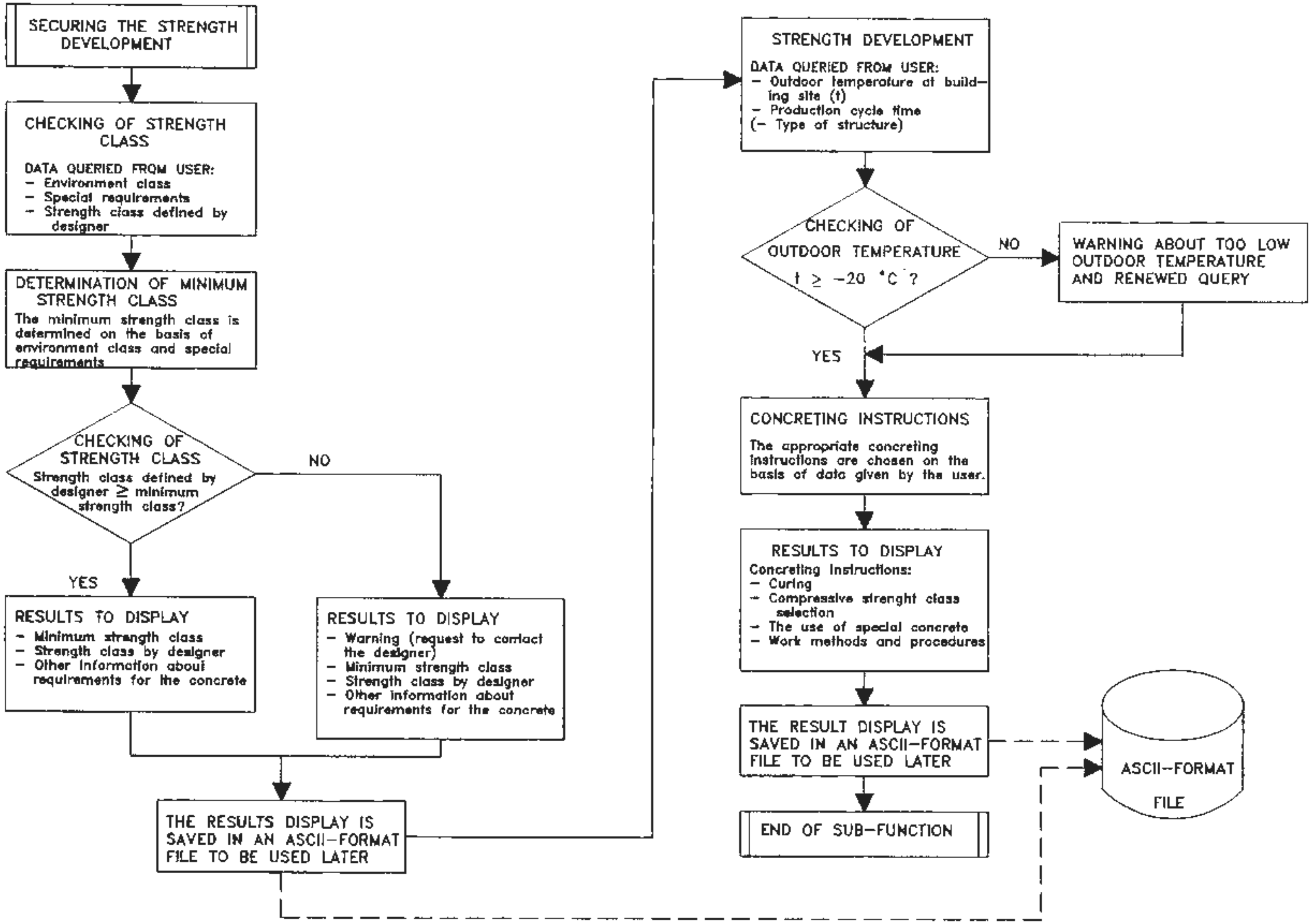


Fig. 6. The course of inference in the context for securing the strength development.

3.2.2 Consistency value

The second context contains knowledge about the consistency value of the fresh concrete (fig. 7). The system deduces the suitable consistency value using following fact values queried from the user:

- structure
  * the type of structure
  * the thickness of the structure
  * the spacing between the reinforcement bars

- production equipment
  * the compaction method
  * the transport method for fresh concrete used on the building site.

3.2.3 Maximum size of aggregate

The third context contains knowledge about the maximum size of aggregate (fig. 8). The facts influencing the choice of maximum aggregate size are

- structure
  * the type of structure
  * the thickness of the structure
  * the spacing between the reinforcement bars
  * the desired quality of the concrete surface

- environmental requirements
  * water impermeability requirements

- production equipment
  * the compaction method
  * the type of form, number of joints in the formwork and the sealing of the joints.

It may be noticed that some of the facts mentioned above have been queried in the previous contexts.

3.2.4 The report generator

The Insight 2+ system does not adequately support the generating of summary reports in knowledge bases, which are divided into several contexts. Therefore a procedure has been written in Turbo Pascal to perform this task. The result displays are stored in a cumulative ASCII-file during a consultation. When the session is over, the Pascal procedure is loaded from the knowledge base and it reads the ASCII-file to form a summary report, which is shown on the display, printed out as a hardcopy or both according to what the user wishes.

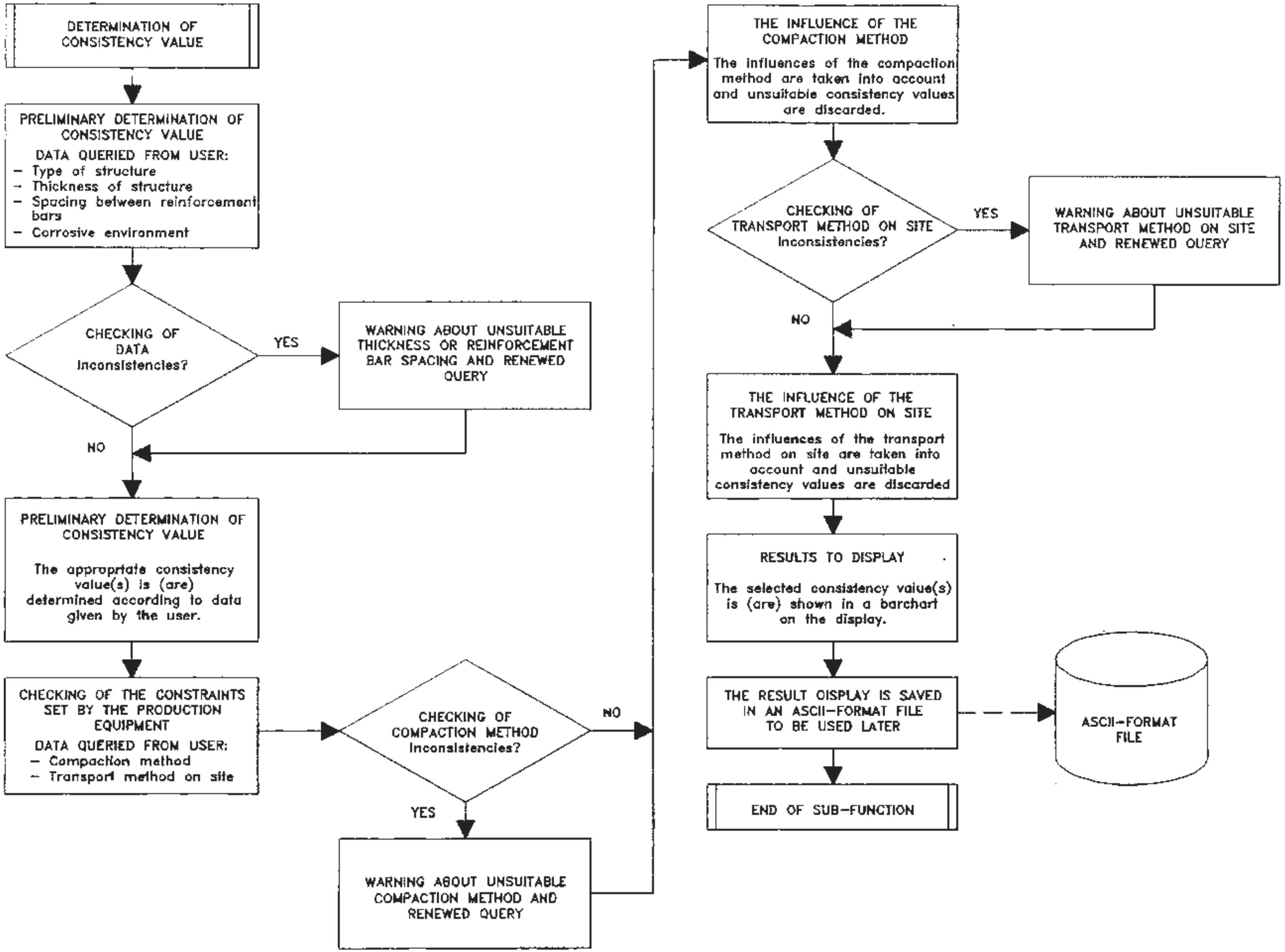


Fig. 7. The course of inference in the context for determining the consistency value.

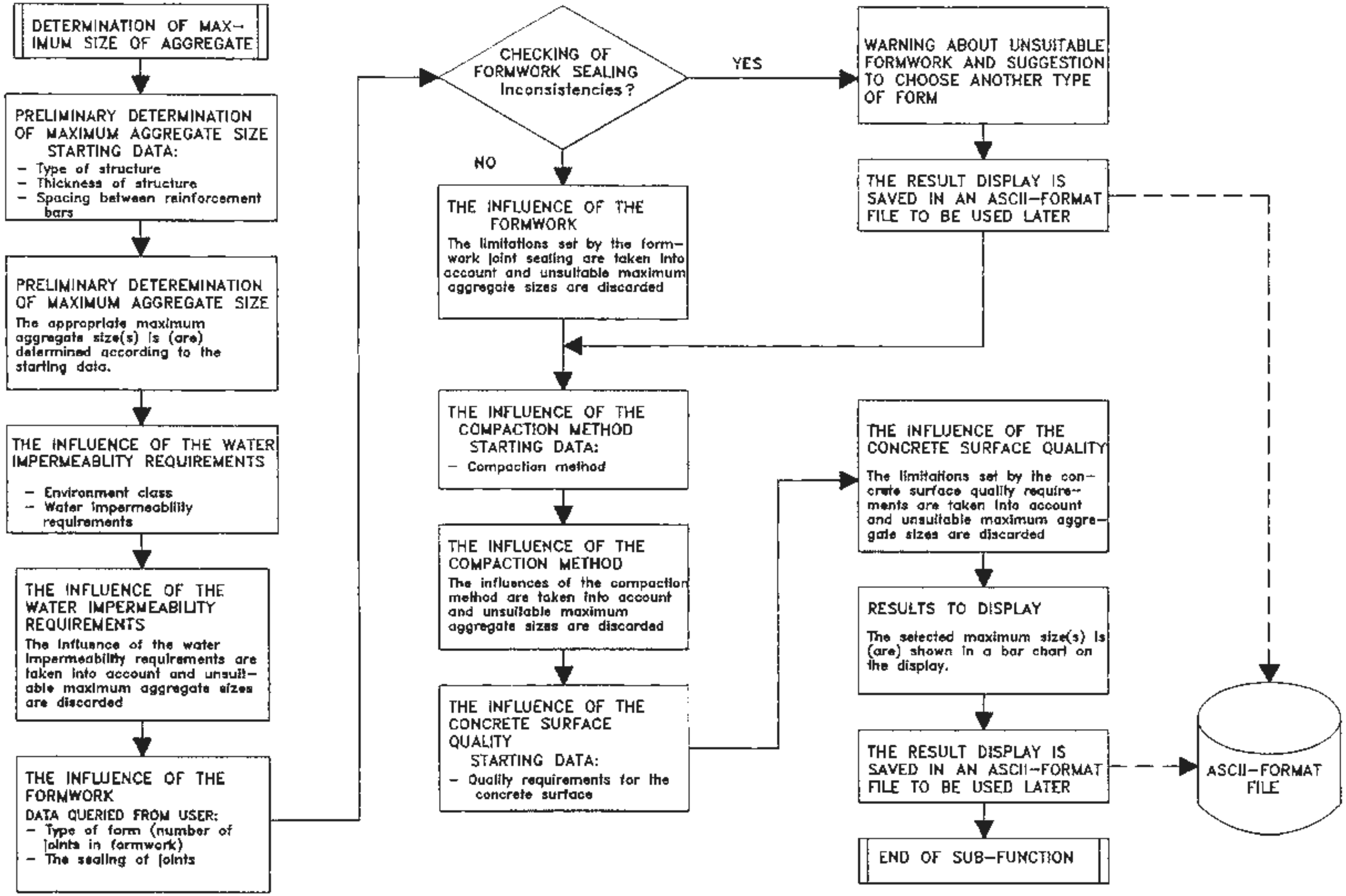


Fig. 8. The course of inference in the context for determining the maximum size of aggregate.

3.3 The user interface

The system automatically generates queries for data input on the basis of the fact statements in the rules. These queries mostly consist of a series of menu-type displays, from which the user can choose the appropriate alternative by moving the cursor with the arrow-keys on the keyboard (fig. 9). Some fact statements in the rules need numerical data, for which the system generates query displays with a field for numerical input (fig. 10).

This type of data input has some disadvantages: the system generates a great amount of queries and an experienced user may consider this timeconsuming. On the other hand, the system acts like a check-list, which may make it useful for educational purposes.

The results from each context are shown on the display as the inference proceeds. The results from the strength class checking and concreting recommendations are shown as textual displays (fig. 11) and the results of the determination of the consistency value and maximum size of aggregate are shown as bar-charts (fig. 12), where the suitability of each alternative is given by the length of the bar.

The user interface includes explanation facilities. The user can ask for additional information concerning a specific fact by pressing the EXPL-key, in which case a set of information defined by the system developer is displayed. Furthermore, the user can ask how the system has come to a certain conclusion or why a specific question is asked by pressing the WHY-key. This latter explanatory information is automatically generated by the inference engine.

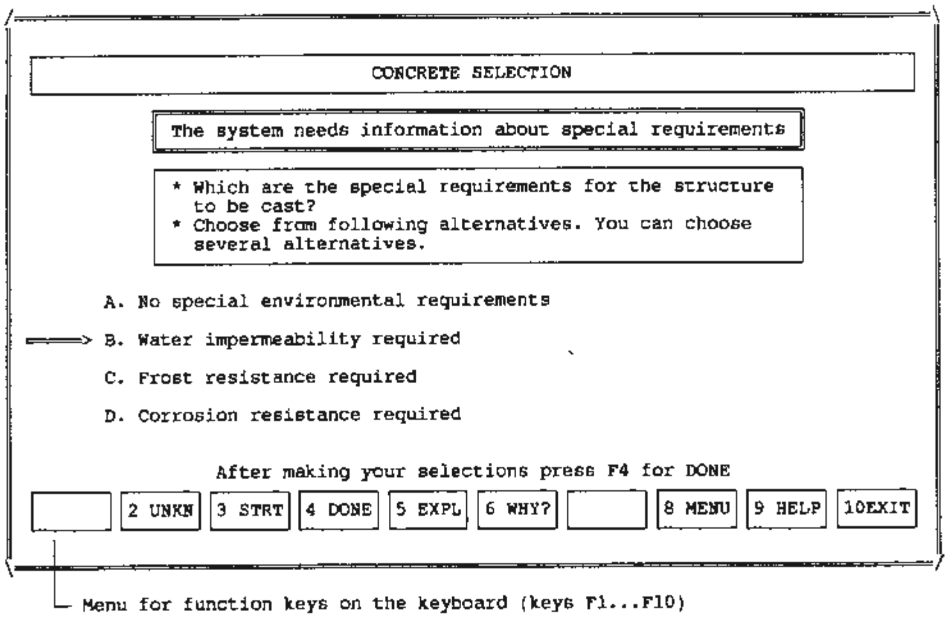


Fig. 9. Example display of menu-type query.

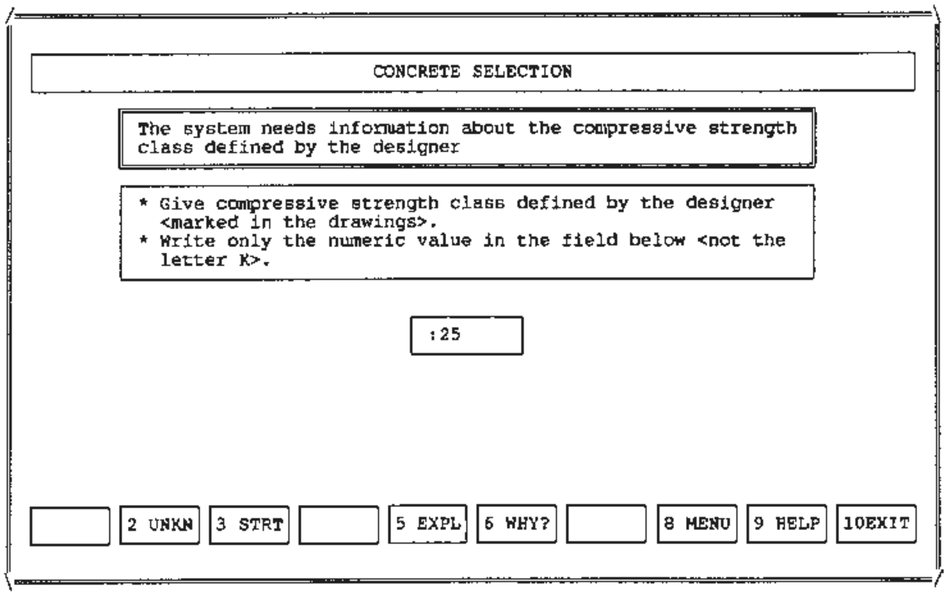


Fig. 10. Example display of numeric-type query.

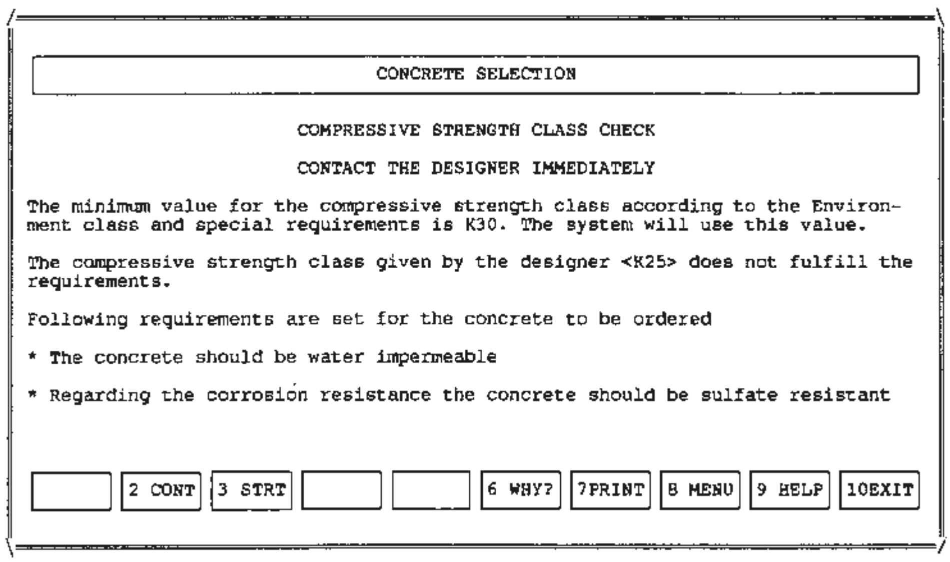


Fig. 11. Example of a textual result display.

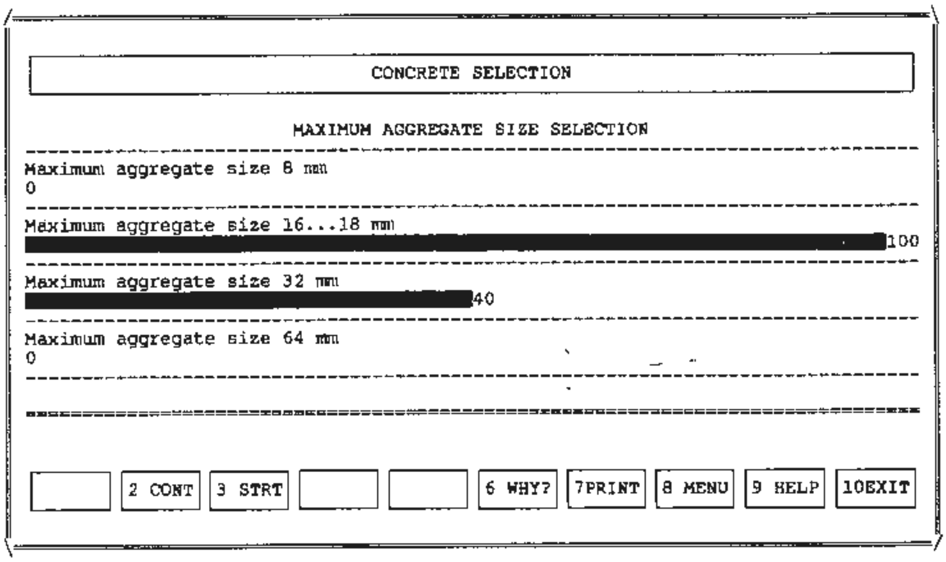


Fig. 12. Example of a bar-chart result display.

4 CONCLUSIONS

The experiences show that it is quite possible to build feasible small microcomputer based expert systems with the software tools available at present. The rapidly developing software and hardware technology shows a great potential for future applications of larger expert systems.

No full-scale system evaluations or tests have yet been performed for the system described. Some small tests with experienced site foremen indicate that the BETVAL-system cannot be practically used in production in its present form, mainly due to the fact that the knowledge contained in the knowledge base is too shallow. What is needed is a system with much deeper and more specific knowledge about special cases of concrete selection. The knowledge base should be divided into smaller knowledge bases according to type of structure. Knowledge about the economical aspects of concrete selection should also be added.

However, the BETVAL-system may be useful for educational purposes. The system works like an automated interactive checklist, which takes into account the basic aspects of ready mix concrete selection, and this, in conjunction with the explanatory facilities provided by the system, makes it an ideal tutorial tool for self-training. Presently the system is in use in one technical school in Finland.

REFERENCES

1. Addendum to the Insight 2 reference manual - Insight 2+. Indialantic, Flor. 1986, Level Five Research Inc. 125 pp.

2. Adeli, H. Software review: Insight 2+. Microcomputers in Civil Engineering 2(1987)2, pp. 173 - 174.

3. Allwod, R. J. et al. Report on expert system shell evaluation for construction industry applications. Loughborough 1985, Loughborough University of Technology, Department of Civil Engineering. 17 pp. + app. 148 pp.

4. Clifton, J. R. et al. Development of DURCON, an expert system for durable concrete: Part I. Gaithersburg, Maryl. 1985, National Bureau of Standards, NBSIR 85-3186. 18 pp.

5. Duda, R. O. & Gaschning, J. G. Knowledge-based expert systems come of age. Byte 6(1981)9, pp. 238 - 280.

6. Gevarter, W. B. An overview if expert systems. Gaithersburg, Maryl. 1982, National Bureau of Standards, NBSIR 82-2505. 29 pp.

7. Hamilton, G. & Harrison, A. Expert systems for the construction and building services industry. Bracknell 1986, Building Services Research and Information Association (BSRIA), Technical Note TN 7/86. 52 pp.

8. Hayes-Roth, F. et al. (eds.) Building expert systems. Reading, Mass. 1983, Addison Wesley. 444 pp.

9. Kinnucan, P. Computers that think like experts. High Technology 1984:1. Reprint. 11 pp.

10. Koskela, L. & Serén, K.-J. An expert system for choosing the type of concrete (in Finnish). Espoo 1987, Technical Research Centre of Finland, Research Notes 722. 62 pp. + app. 3 pp.

11. Koskela, L. et al. Expert systems in construction: Initial experiences. In: Pham, D. T. (ed.) Expert systems in engineering. Berlin 1988, IFS Publications & Springer-Verlag. Pp. 175 - 188.

12. Sriram, D. et al. Expert systems for civil engineering - A survey. Pittsburgh, Penn. 1982, Carnegie-Mellon University, Department of Civil Engineering, R-82-137. 97 pp.

13. Wager, D. M. Expert systems and the construction industry. Cambridge 1984, Construction Industry Computing Association (CICA). 128 pp.

14. Weiss, S. M. & Kulikowski, C. A. A practical guide to building expert systems. London 1984, Chapman and Hall. 174 pp.