

Building and Environment, Vol. 40, No. 1, 2005, pp. 73-81

A KNOWLEDGE-BASED SYSTEM FOR LIQUID RETAINING STRUCTURE DESIGN WITH BLACKBOARD ARCHITECTURE

K.W. Chau

Department of Civil & Structural Engineering, Hong Kong Polytechnic University, Hunghom, Kowloon, Hong Kong

Email: cekwchau@polyu.edu.hk, tel.: (852) 2766 6014, fax.: (852) 2334 6389

F. Albermani

Department of Civil Engineering, University of Queensland, QLD 4072, Australia

Abstract

Owing to the high degree of vulnerability of liquid retaining structures to corrosion problem, there are stringent requirements in its design against cracking. In this paper, a prototype knowledge-based system is developed and implemented for the design of liquid retaining structures based on the blackboard architecture. A commercially available expert system shell VISUAL RULE STUDIO working as an ActiveX Designer under the VISUAL BASIC programming environment is employed. Hybrid knowledge representation approach with production rules and procedural methods under object-oriented programming are used to represent engineering heuristics and design knowledge of this domain. It is demonstrated that the blackboard architecture is capable of integrating different knowledge together in an effective manner. The system is tailored to give advice to users regarding preliminary design, loading specification and optimized configuration selection of this type of structure. An example application is given to illustrate the capabilities of the prototype system in transferring knowledge on liquid retaining structure to novice engineers.

Introduction

By its very nature, the design of a building structure is a complicated process which integrates both design knowledge and analytical skills. It is not easy to incorporate the available empirical and heuristic knowledge into a numerical model for decision making. It is even more difficult for liquid retaining structure design due to the high degree of vulnerability of these specialized structures to corrosion problems. There are stringent requirements against cracking and a considerable amount of heuristic inferences, design code requirements, and expert experience are involved, in accordance to the particular configuration, support conditions and loading requirements. However, up till the present, algorithmic models are often used to simulate structural design processes, resulting in large constraints in model uses as well as a large gap between model developers and practitioners. Most algorithmic models are not user-friendly enough and lack sufficient support to novice users and knowledge transfers in appropriate model application. Yet, there is a real necessity to transfer this expertise knowledge to novice engineers.

Moreover, design can be considered an open-ended problem to select the optimal solution, via minimization of either the total cost or weight of the entire structure. There are often several alternatives that can satisfy all the specified constraints. In this point of view, structural design can be represented by a cooperative constraint-satisfying process with several knowledge sources. In order to address such an ill-structured problem, skillful as well as iterative knowledge manipulation and learning process are entailed for selection of the most appropriate design parameters and successful implementation of computer-aided design.

Symbolic programming is desirable in this design process, which becomes one of the domain problems that are particularly suitable for the application of knowledge-based system technology. Other advantages of the application are the substantial reduction of the design duration and avoidance of any inadvertent human errors during the data transfer in different processes.

During the past decade, many knowledge-based systems have been applied in different domain problems (Chau, 1992; Chau, 2004a,b; Chau et al., 2002; Chau & Anson, 2002; Chau & Chen, 2001; Chau & Cheung, 2004; Chau & Ng, 1996; Chau & Yang, 1994; Chau & Zhang, 1995). There are also examples of knowledge-based system applications in structural design: BERTH (Ranga Rao & Sundaravadivelu, 1999); and, LADOME (Lin & Albermani, 2001). This paper delineates a prototype knowledge-based system for analysis and design of liquid retaining structures with blackboard architecture.

Architecture of prototype system

Figure 1 shows the flowchart that integrates the algorithmic programs and knowledge management for liquid retaining structures using the blackboard architecture. A characteristic of this type of architecture is the contribution from several knowledge sources at different levels for problem solving in a single system (Engelmore and Morgan, 1988). It suits the nature for design of engineering structure, which highly relies upon interaction between diversified knowledge sources. Under this type of architecture, expertise in the form of knowledge sources is often grouped into individual modules with various knowledge representation methods including rules, frames, object-oriented techniques, etc. The common data structure, termed the blackboard, undertakes the role to compile the data entries and to link among various knowledge sources to attain information sharing. It functions as the global system context and stores the existing state of the solution, comprising initial input data, intermediate variables and final solution.

Since several commercially available expert system shells are available nowadays, the determination of the knowledge representation scheme is largely based upon the nature of domain problem as well as upon the capability of the programming tool. It should be noted that each representation technique has both its own pros and cons. Whilst rule-based system has the edge of highly comprehensiveness and popularity, it is not on its own capable of addressing design task of formation or synthesis nature. Object-oriented technique is more versatile because of its modularity, data abstraction and inheritance characteristics, yet is too complicated to represent simple logic. Thus, a hybrid programming technique is employed here in order to take advantages of the characteristics of each scheme. The hybrid application development tool expert system shell, VISUAL RULE STUDIO (Rule Machine Corporation, 1998), integrates object-oriented techniques, relational database models, expert system technology, traditional procedural programming and graphics in a windowing environment. It works as an ActiveX Designer under the VISUAL BASIC programming environment.

Production rules and chaining inference

Rule type representation

The knowledge rules constitute the main core of the knowledge base in this prototype system. Hence, whenever the production rule format is appropriate, they are used to represent heuristics and deep design knowledge during both the preliminary synthesis and specification stages. They have the advantage of accommodating the grouping of knowledge together, thus facilitating program coding as well as updating. They are, however, not too appropriate to represent lengthy procedural methods because of their inherent nature on limited

mathematical functions as well as relatively simple syntax. For legibility and comprehensibility purposes, the rule syntax is usually represented in quite a natural English language. Production rules are basically represented in a IF-THEN-ELSE format so that knowledge can be easily added, deleted or revised by simply editing the rules in the knowledge base, thus facilitating future extension. Whenever rules and conditions match one another and certain conditions become true, the rules will be invoked and actions in conclusion statements will be triggered. The newly triggered actions may alter the existing state of contexts, thus invoking in turn new rule matching in a cycle-wise manner. It is through a backward inference mechanism that testing and firing rules, very often in the form of a linked chain, are accomplished. Currently, there are about 200 production rules in this prototype system for the determination of various context values of design entities during different design stages. The knowledge required for the determination of various geometrical ratios, interpolation of moment and shear coefficients, selection of design parameters such as shape factors in wind load determination, and different material properties are covered here. Figure 2 shows an example of knowledge rule and inference procedure.

Chaining inference

A mixed-chaining inference mechanism is adopted in this prototype system. Whilst the design steps are processed in a forward chaining way, the design decision and intermediate results are derived in a backward chaining fashion. All the design steps can be seen explicitly on the main screen display and there are no fixed agenda or monitor. The user is allowed to opt for the preferred sequence of design processes, whose validity is checked by Process Control knowledge modules. Upon being triggered by the user or situation, these modules act opportunistically during the design process. As an example, during the detailed design stage, the applicable design loads on the structure are dependent upon the type of liquid retaining structure. If an underground liquid retaining structure is adopted, the system will prompt the user to enter the soil properties whilst if the liquid retaining structure considered is above the ground, it will prompt the user to input the wind load parameters.

Knowledge elicitation and representation

During the development of the system, design knowledge is gleaned from literature, human expert as well as from the development process itself. In order to emulate the working processes of a human expert, the domain knowledge is first transformed into a representation form amenable to programming. The knowledge base is the core component of a knowledge-based system. It comprises contemporary design knowledge originated from many sources: case studies; commercial product catalogue; databases; derived knowledge in the system development process; design code; design standard; engineering handbook; empirical data; engineering reports; heuristic experience; and, journal literature. Its contents covers non-exclusively different types of liquid retaining structures, code provisions as well as heuristic practice about concrete cover, principal dimensions, effective span, minimum grade of concrete, minimum grade of reinforcement, minimum percentage of reinforcement, placing of additional reinforcement. In order to suit the individual nature of knowledge here, a multitude of knowledge representation methods including algorithmic programs, databases, objects, rules, procedural methods, are utilized.

Object-oriented representation

A major component in the knowledge base is the use of object to represent declarative knowledge for both data and executable procedural code. Under the blackboard architecture, the solution procedures in the design process are separated from the design parameters in the blackboard. As shown in Figure 3, these objects can be categorized under two main groups:

the blackboard; and, knowledge modules. The paradigm is tailored so that the attribute values of different objects can be saved and retrieved whenever they are entailed during the design process. Each object in the blackboard incorporates some attributes, which are used to express the physical structural components, the facts used in the design, or the pertinent design entities entailed during the design solution process whilst the blackboard is partitioned into a number of hierarchical levels, corresponding to different stages of the design process. The types of attribute may be: compound; instance reference; interval; multi-compound; numeric; simple; string; and, time. Facets, representing the fact about an attribute, are employed to design the inference strategy for processing an attribute. A typical example of facets is the search order, which facilitates the determination of the value of an attribute by defining the method used by the backward-chaining inference engine.

As shown in Figure 3, Design Entities and Design Stage are two groups of objects in the blackboard. Whilst Design Entities represent different types of entities, Design Stage is unique in that it is used mainly by Process Control knowledge modules to determine the next possible action during the design process, or to check whether or not it is possible to execute the user-requested functions. In deriving the next design process, forward chaining inference mechanism is triggered either by the system or the user. Each Design Stage indicator will be assigned one of the preset values, once a specific design stage has been properly implemented.

Also shown in Figure 3, Process Control and Design Process are the two categories of knowledge modules, which represent procedural expertise knowledge in solving design problems. The significance of Process Control modules is on their representation of meta-knowledge to monitor and control the nature and time of action for a particular design stage. In other words, they define the problem-solving strategy in the execution and processing of deep and heuristic design knowledge represented in the Design Process modules. The module is mainly event-driven and is executed in a forward chaining inference manner. After having validated whether or not to execute a specific user-interface task or a design step, they determine whether or not to proceed or to prompt certain warning message. Amongst various classes in Process Control knowledge modules, main Design Process monitors the design stage of all key tasks during the design process. Other subtasks, which are arranged under various design stages such as analysis and sizing, design summary, load determination, preliminary design, etc., are mostly related to the user interface. At any design stage, the user is allowed to change one's mind in modifying some design parameters or even adopting an entirely different configuration. Moreover, any updates to the parameters during any design stage will be propagated automatically and synchronously to all its pertinent design entities so that consistency of the system is maintained.

The objects in Design Process modules represent the major part of the knowledge, covering various steps to furnish feasible configurations which can satisfy the design specifications and to assess these alternative configurations for structural optimization under certain code stipulations. In this knowledge module level, a mixed problem-solving strategy, containing forward chaining and backward chaining inference mechanisms, is involved. When the attribute value of these objects changes, the procedural method attached to it will be processed. These methods have also the functions to call external algorithmic programs, to access database files or even to test and then trigger rules. The semantic network representing the relationships amongst different objects in Design Process modules is shown in Figure 4.

Procedural methods

A portion of the stipulated design procedures specified by British Standard BS8007 (British Standards Institution, 1987) are represented by procedural methods containing preset sequences of actions. Forward chaining is the main driving mechanism in this process when there is a change of the attribute value of the object that may be triggered by the user via the pertinent command button or through other rule chains. The functions of procedural methods include executing mathematical function, assigning new values, rule testing and triggering, database queries, triggering other methods, reading and writing external data files, and calling external programs. In order to represent the design processes, procedural methods are attached to attributes of objects in Design Process and Process Control knowledge modules, but not to the objects in the blackboard. Figure 5 shows examples of procedural methods attached to textbox TxtLiquidLevel in the screen of imposed load specification.

Algorithmic programs

In order to enhance its capability in numerical processing, VISUAL RULE STUDIO allows for switching over to and from other algorithmic programs. With this external extension, it can at the same time deal with both symbolic and algorithmic processing. Upon the completion of execution of the external program, the previous session in the prototype system is resumed and outputs resulted from numerical processing program can be retrieved for further processing. In this system, several custom-built algorithmic programs, written in VISUAL BASIC, are involved in the following design processes: code conformance checking; numerical model generation; optimized member sizing; and, preliminary design.

The finite element analysis program for the analysis and sizing is much more complicated. It is not cost-effective to custom-build an algorithmic program for this purpose. Hence an external conventional finite element model, ABAQUS, is adopted to perform nonlinear analysis (Hibbitt et al, 1998). A link to another package Structural Analysis Program (SAP) has been unsuccessful because the outputs of the current version are only in graphical forms but not retrievable in text files whilst its older version is DOS-based, which is incompatible with the windows-based environment (Computers and Structures, Inc., 1991). In the processes, unformatted ASCII text files are generated either by the knowledge base or the algorithmic program. These files constitute the common base for effective communication between the knowledge base and various algorithmic programs.

Database representation

A database format is an effective method to represent engineering knowledge for centralized and independent storage of huge amount information. Here, engineering knowledge covering structural properties of reinforced concrete sections, moment and shear coefficients for various configurations in preliminary design, structural properties of proposed alternatives and final member details in detailed design, are represented in a form of database table. The MICROSOFT ACCESS format, which is highly portable and is compatible with other popular database systems under the Window environment, is adopted here. Direct connection to ACCESS databases is accommodated in the system via an interface module linking the knowledge base and database files, without the installation the program itself. Various attribute groups, including bar area, bar area percentage, bar size, bar spacing, concrete cover, crack width, maximum span, service moment, slab thickness, total material cost, ultimate moment, ultimate shear, etc. represent the database on the structural properties of reinforced concrete sections.

Heuristics are employed to constrain the selection of some design parameters to practical

values. As typical examples, allowable reinforcement sizes are amongst 10, 12, 16, 20, 25, 32 and 40 mm, concrete cover can be 40, 50, 60 or 75 mm, the concrete slab thickness can only be one of 200, 225, 250, 275, 300, 350, 400, 450, 500, 600, 700, 800, 900, or 1000 mm, etc. Some knowledge is static in that they are created during the system development and cannot be changed by any design activities, unless when new information is obtained. Other category of knowledge is dynamic in that they are generated during the execution of the system and they are often stored outside the knowledge-base system shell. Typical examples of static and dynamic knowledge are structural properties of reinforced concrete sections and structural properties of proposed alternative, respectively.

Case Study

In order to demonstrate the application of the prototype system, a case study is made on a typical liquid retaining structure. The spatial requirements of the structure are as follows:

Shape = circular
Volume = 100 m³
Depth = 5 m
Location = underground
Exposure condition = very severe

The user is required to enter the values of the required parameters at the appropriate fields of visual edit screens and menus under a Windows-based user interface. Figure 6 shows the screen for structural specification in preliminary design by the user. The consistency and accuracy of all input data are checked to be within the allowable range. The system will prompt a warning message otherwise.

The system performs a preliminary synthesis and evaluates different alternatives using heuristics in accordance to these geometric constraints, crack width requirement, span depth ratio, and moment and shear coefficients of the configuration. 15 feasible proposed configurations are then proposed and ranked on the basis of the material costs. An option button is tailored for the user to select between recommendation made by the system or one's own choice. The default design parameters shown as follows are adopted here:

Concrete grade = 40
Reinforcement grade = high yield steel
Concrete cover = 40 mm
Aggregate type = granite or basalt
Temperature variation = 65 °C
Unit cost of concrete = \$3000 per m³ of concrete
Unit cost of reinforcement = \$500 per tonne of reinforcement

Prior to detailed design and analysis, the relevant detailed design specification, as shown in the following, is input for the selected alternative.

Support specification = fixed
Surcharge load = 10 kN/m²
Level of liquid = 5 m
Level of water table = 4 m
Ground level above tank bottom = 5 m
Specific weight of soil = 20 kN/m²

Active soil pressure coefficient = 0.3
Load combination = default load combinations
(1.4DL + 1.4WL/ 1.4DL + 1.6LL/ 1.2DL + 1.2LL + 1.2WL)
Number of elements per surface = 100 (10x10)

After several iterations of design process including numerical model generation, structural analysis, code conformance checking and member sizing, it is determined that a member thickness of 225 mm with reinforcement diameter 16 mm at spacing 100 mm results in a structure with minimum cost. Figure 7 shows the screen displaying the best member sizing record of the recommended section.

Conclusions

A prototype knowledge-based system for design of liquid retaining structures is developed and implemented with blackboard architecture employing hybrid knowledge representation technique under object-oriented environment. In order to tailor the nature of knowledge involved in the structural design process, various knowledge representation methods including objects, procedural methods, production rules, databases as well as external algorithmic programs, are employed. The blackboard architecture is demonstrated to be able to integrate various structural design processes. The system undertakes the role of storage of heuristic knowledge as well as medium of knowledge transfer from experts to novice designers, thus reducing the workload of experts. In addition to the recommended solution, the system furnishes a list of several feasible design alternatives in a priority order of total material cost, which is of great assistance even if the user desires to adopt other alternative. Other advantages are the consistency and efficiency of the design output solution.

References

- British Standards Institution, 1987, BS 8007: Design of Concrete Structures for Retaining Aqueous Liquids.
- Chau, K.W., 1992, An expert system for the design of gravity-type vertical seawalls, *Engineering Applications of Artificial Intelligence*, 5(4), 363-367.
- Chau, K.W., 2004a, A prototype knowledge-based system on unsteady open channel flow in water resources management, *Water International*, 29(1), 54-60.
- Chau, K.W., 2004b, Knowledge-based system on water-resources management in coastal waters, *Journal of the Chartered Institution of Water and Environmental Management*, 18(1), 25-28.
- Chau, K.W. and Anson, M., 2002, A knowledge-based system for construction site level facilities layout, *Lecture Notes in Artificial Intelligence*, 2358, 393-402.
- Chau, K.W. and Chen, W., 2001, An example of expert system on numerical modelling system in coastal processes, *Advances in Engineering Software*, 32(9), 695-703.
- Chau, K.W., Cheng, Chuntian and Li, C.W., 2002, Knowledge management system on flow and water quality modeling, *Expert Systems with Applications*, 22(4), 321-330.
- Chau, K.W. and Cheung, C.S., 2004, Knowledge representation on design of storm drainage system, *Lecture Notes in Artificial Intelligence*, 3029, 886-894.
- Chau, K.W. and Ng, Vitus, 1996, A knowledge-based expert system for design of thrust blocks for water pipelines in Hong Kong, *Water Supply Research and Technology - Aqua*, 45(2), 96-99.
- Chau, K.W. and Yang, Wen-wu, 1994, Structuring and evaluation of VP-expert based knowledge bases, *Engineering Applications of Artificial Intelligence*, 7(4), 447-454.
- Chau, K.W. and Zhang, X.N., 1995, An expert system for flow routing in a river network, *Advances in Engineering Software*, 22(3), 139-146.
- Computers and Structures, Inc., 1991, Structural Analysis Programs SAP90, Volume 1-3.

Engelmore, R. and Morgan, T., 1988, *Blackboard Systems*, Addison_Wesley, Wokingham.
Hibbitt, Karlsson & Sorensen, Inc., 1998, ABAQUS User's Manual, Version 5.8.
Lin, S. and Albermani, F., 2001, Lattice-dome design using A knowledge-based system approach, *Computer-aided Civil and Infrastructure Engineering*, 16(4), 268-286.
Rule Machine Corporation, 1998, Visual Rule Studio Developer's Guide.
Ranga Rao, A.V. and R. Sundaravadivelu. (1999). "A knowledge based expert system for design of berthing structures", *Ocean Engineering*, Vol. 26, No. 7, pp. 653-673.

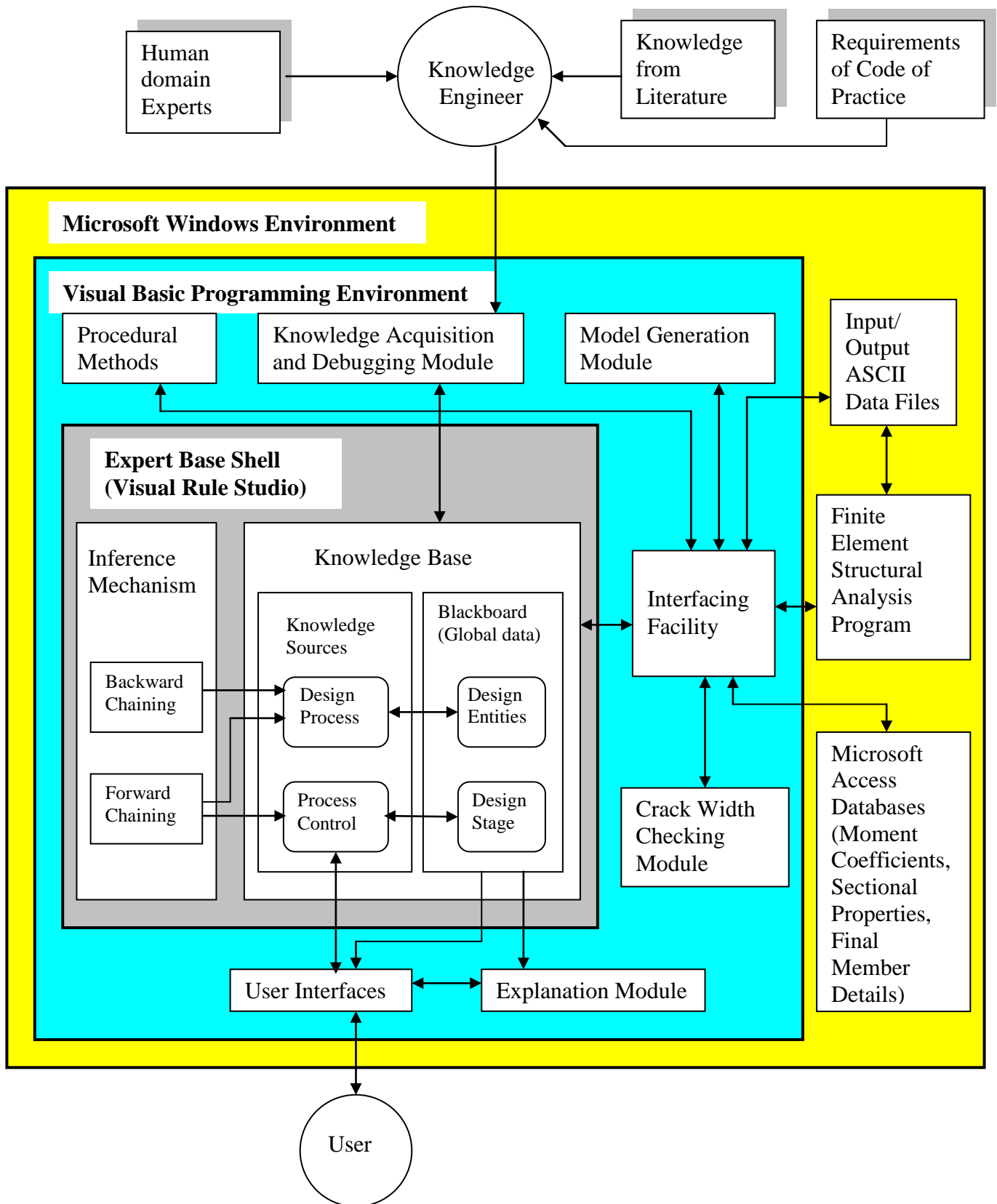


Figure 1. Architecture of the blackboard knowledge-based system

!RULE GROUP: design moment & shear OF BBPreliminaryParameters

RULE to find designServiceMoment : 1 of 7

IF shape OF BBConfigurationRequirement IS rectangular

AND momentMh OF BBPreliminaryParameters \geq momentMv OF BBPreliminaryParameters

THEN designServiceMoment OF BBPreliminaryParameters := momentMh OF BBPreliminaryParameters

RULE to find designServiceMoment : 2 of 7

IF shape OF BBConfigurationRequirement IS rectangular

AND momentMh OF BBPreliminaryParameters $<$ momentMv OF BBPreliminaryParameters

THEN designServiceMoment OF BBPreliminaryParameters := momentMv OF BBPreliminaryParameters

RULE to find designServiceMoment : 3 of 7

IF shape OF BBConfigurationRequirement IS circular

THEN designServiceMoment OF BBPreliminaryParameters := momentCoefficient OF

BBPreliminaryParameters*specificGravityOfLiquid OF BBConfigurationRequirement*height OF
BBConfigurationRequirement³

RULE to find designUltimateMoment : 4 of 7

IF designServiceMoment OF BBPreliminaryParameters $>$ 0

THEN designUltimateMoment OF BBPreliminaryParameters := 1.4*designServiceMoment OF
BBPreliminaryParameters

RULE to find designUltimateShear : 5 of 7

IF shape OF BBConfigurationRequirement IS rectangular

AND shearVh OF BBPreliminaryParameters \geq shearVv OF BBPreliminaryParameters

THEN designUltimateShear OF BBPreliminaryParameters := 1.4*shearVh OF BBPreliminaryParameters

RULE to find designUltimateShear : 6 of 7

IF shape OF BBConfigurationRequirement IS rectangular

AND shearVh OF BBPreliminaryParameters $<$ shearVv OF BBPreliminaryParameters

THEN designUltimateShear OF BBPreliminaryParameters := 1.4*shearVv OF BBPreliminaryParameters

RULE to find designServiceTension : 7 of 7

IF shape OF BBConfigurationRequirement IS circular

THEN designServiceTension OF BBPreliminaryParameters := tensionCoefficient OF

BBPreliminaryParameters*specificGravityOfLiquid OF BBConfigurationRequirement*height OF
BBConfigurationRequirement*diameter OF BBConfigurationRequirement/2

Figure 2. An example of knowledge rule and inference procedure

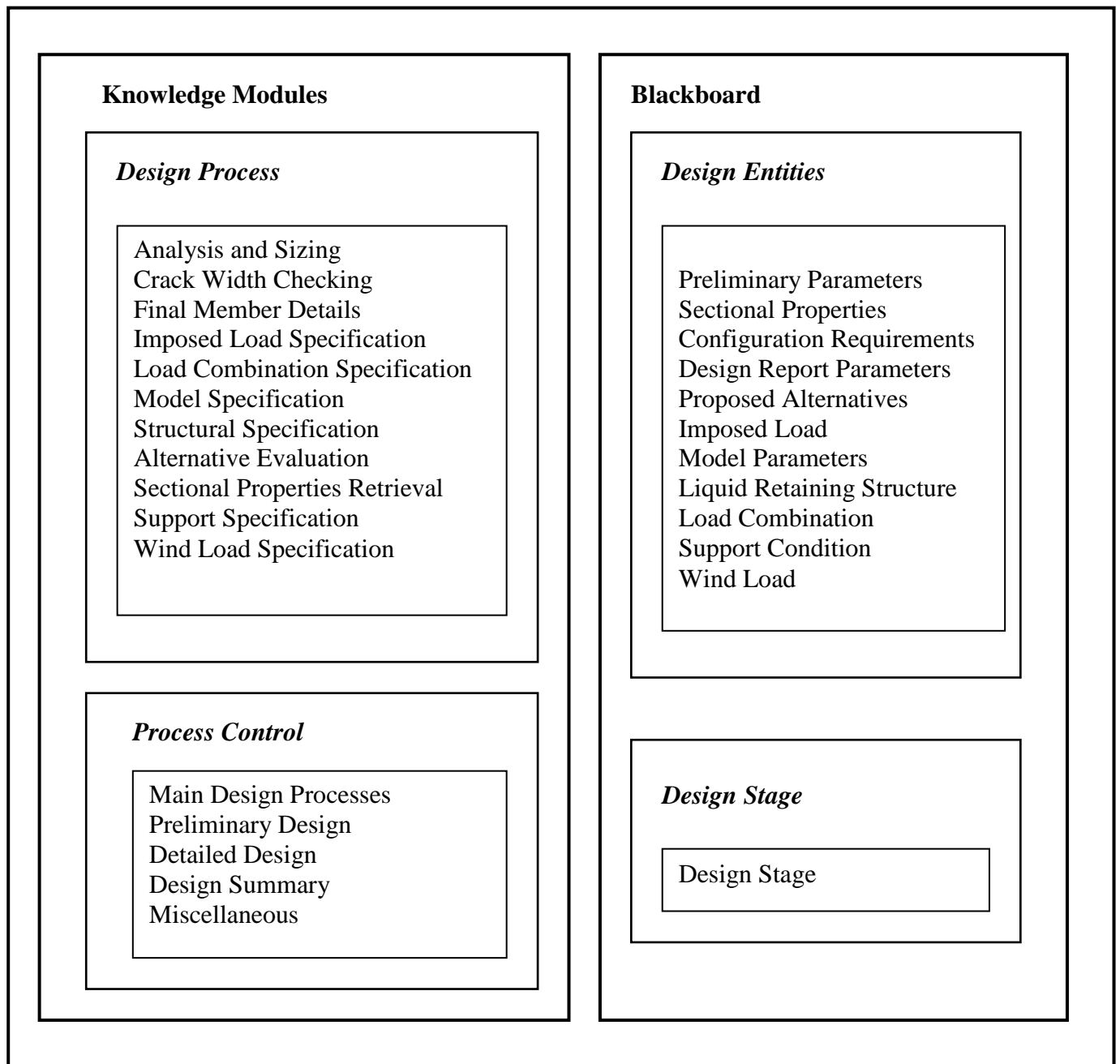


Figure 3. Various Classes in the knowledge base

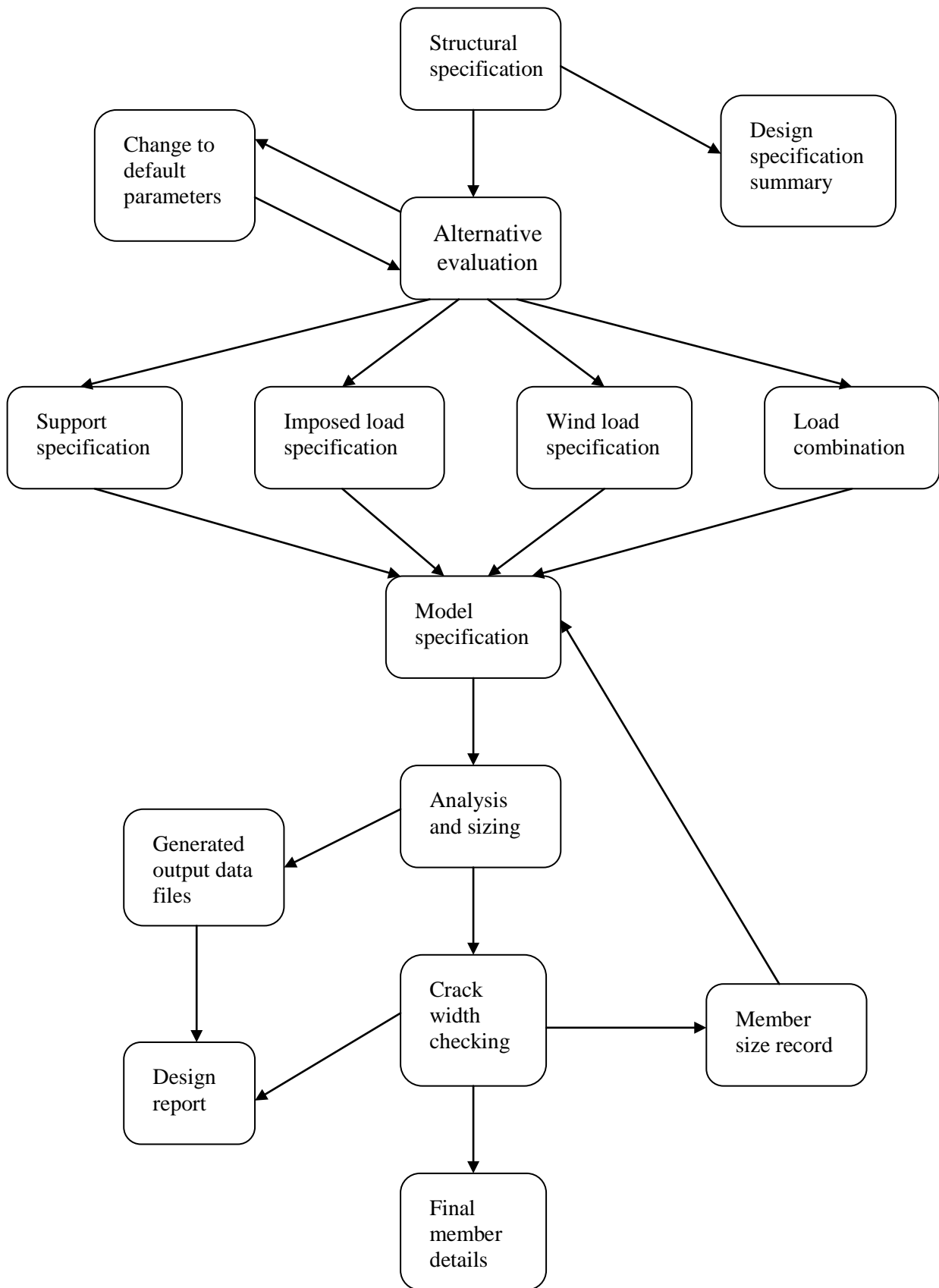


Figure 4. Semantic network representing key design process

```

Private Sub TxtLiquidLevel_Change()
    CheckText TxtLiquidLevel, "Liquid level", "LIQUID LEVEL"
    If TxtLiquidLevel.Text = " " Then BBImposedLoad.liquidLevel = Null _
        Else BBImposedLoad.liquidLevel = Val(TxtLiquidLevel.Text)
    If Val(TxtLiquidLevel.Text) > BBConfigurationRequirement.Height Then
        MsgBox "Liquid level should be lower than or equal to height of tank. " & _
            "Please enter again.", vbOKOnly + vbCritical, "ERROR: LIQUID LEVEL!"
        TxtLiquidLevel.Text = " "
        CmdContinueToMain.Enabled = False
    End If
    UnspecifyImposedLoad
End Sub

Private Sub CheckText(TextName As TextBox, String1 As String, String2 As String)
    If TextName.Text = " " Or TextName.Text = "" Then
        CmdContinueToMain.Enabled = False
    ElseIf Not IsNumeric(TextName.Text) Then
        MsgBox "Please enter positive numeric " & String1 & " value.", _
            vbOKOnly + vbCritical, "ERROR: " & String2 & "!"
        TextName.Text = " "
    ElseIf Val(TextName.Text) < 0 Then
        MsgBox String1 & " is less than or equal to zero. Please enter again.", _
            vbOKOnly + vbCritical, "ERROR: " & String2 & "!"
        TextName.Text = " "
    Else
        EnableSearchButton
    End If
End Sub

```

Figure 5. Typical examples of procedural methods attached to textbox TxtLiquidLevel in the screen of imposed load specification

PRELIMINARY DESIGN

Structural Specification

Spatial requirements

Volume required m³

Height m

Diameter 5.05 m

Location

☒ Underground

☐ Elevated

Shape of structure

☐ Rectangular (single compartment)

☐ Rectangular (2 compartments)

☒ Circular

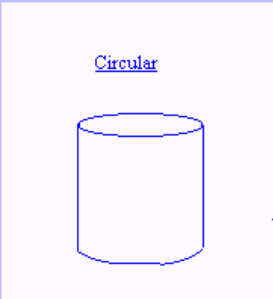
Exposure condition

☐ Severe exposure (0.2mm crack width)

☒ Very severe exposure (0.1 mm crack width)

Density of liquid kg/m³

Width/breadth ratio



Circular

Figure 6. Screen showing structural specification in preliminary design



Figure 7. Screen showing the best member sizing record