Lecture Notes in Computer Science, Vol. 2436, 2002, pp. 95-105

Web-Based Knowledge-Based System on Liquid Retaining Structure Design as Instructional Tool

F. Albermani[1] and K.W. Chau[2]

[1]Department of Civil Engineering, University of Queensland, Australia
[2]Department of Civil & Structural Engineering, Hong Kong Polytechnic University, Hunghom, Kowloon, Hong Kong
cekwchau@polyu.edu.hk

Abstract. A novice engineer often faces many difficulties during the design of liquid retaining structures, which involve making many decisions on the basis of judgment, heuristics, code of practice, rules of thumb, and previous experience. There is a need to develop programming environments that can incorporate engineering judgment along with algorithmic tools. In this paper, the development of a web-based knowledge-based system in design of liquid retaining structures, using blackboard architecture with hybrid knowledge representation techniques including production rule system and object-oriented approach, is presented. It is based on British Standards Codes of Practice BS8007 and BS8110. Tailor-made explanations are furnished to direct and assist inexperienced designers or civil engineering students to learn and capture how to design liquid retaining structures effectively and sustainably in their design practices. The use of this intelligent tutoring system in disseminating heuristic knowledge as well as experience to practitioners and civil engineering students is demonstrated.

1 Introduction

A novice engineer may face many difficulties during the design process of liquid retaining structures involving making many decisions on the basis of judgment, heuristics, code of practice, rules of thumb, and previous experience. One has to select various design parameters including configuration, material, loading, etc. Although computer technology has been developing in a fast pace, their use in engineering field was mainly confined to the number crunching of large volumes of numerical data. In the realm of structural design, this use has been limited almost exclusively to algorithmic solutions [1]. Yet structural design problems are often ill structured and as such, needs arise for programming environments that can incorporate engineering judgment along with algorithmic tools [2]. The advances in artificial intelligence (AI) techniques have demonstrated the capability to furnish assistance in this domain during the past decades. A knowledge-based system (KBS), which is considered suitable for solving problems that demand considerable expertise, judgment or rules of thumb, has emerged promisingly covering a wide range of applications [3-11]. It has developed into practical problem solving tools that can reach a level of

performance comparable to that of a human expert in some specific problem domains. All these applications can be broadly classified into the following categories: diagnosis; design; data interpretation; planning; and education. Areas of early applications of KBS technology include medical diagnosis, mineral exploration and chemical spectroscopy. Many researchers are developing programs that borrow AI concepts to automate common engineering analyses.

In this paper, a web-based KBS in design of liquid retaining structures, using blackboard architecture with hybrid knowledge representation techniques including production rule system and object-oriented approach, is presented. An expert system shell, Visual Rule Studio, is employed to facilitate the development of this prototype system, which is a coupled system integrating symbolic processing and numerical processing. The KBS developed is based on British Standards Codes of Practice BS8007: 1987: Design of concrete structures for retaining aqueous liquids [12] and BS8110: 1985: Structural use of concrete [13]. Tailor-made explanations are furnished to direct and assist inexperienced designers or civil engineering students to learn and capture how to design liquid retaining structures effectively and sustainably in their design practices. The use of this intelligent tutoring system in disseminating heuristic knowledge as well as experience to practitioners and civil engineering students is demonstrated.

2 Liquid Retaining Structure Design

In Hong Kong, most liquid retaining structures are constructed by reinforced concrete with design life of 50 years. Crack width checking is performed to ensure impermeability of concrete and prevention from corrosion of reinforcement. Two kinds of classification are usually used, namely, on the basis of the shape or the location. According to the shape, it is classified as rectangular, circular or polygon. If its location is used as the criterion, it is classified as underground or above the ground. Compared with a circular tank structure with the same width, a rectangular liquid retaining structure has larger volume. However, because of stress concentration at corners, rectangular structures will be more vulnerable to failure. It also has a weaker deflection control. With a circular tank design, not all the spaces are utilized. Since a circular structure can be constructed monolithically without any construction joints, it has better strength quality. With precise structural analysis, a circular structure has a better control in deflection, crack width, bending moment resistance, axial compression resistance, and shear resistance than rectangular structure. A polygon liquid retaining structure is usually used for aesthetic purposes, such as a fountain in a garden and the retaining height is usually not very high.

In some cases, the tank is connected to underground pipe network system, in order to reduce maintenance cost, it will be constructed underground to suit the invert level of the pipe network system. The underground structure is mainly subjected to lateral earth pressure or lateral water pressure, which is due to the underground water table. Besides structural failure mode, bearing capacity of soil and settlement of structure also need to be checked. If the soil bearing capacity does not satisfy the requirement, pile foundation or raft foundation will be required. A liquid retaining structure above

the ground is only subject to liquid pressure due to its own retaining liquid. The structure can either rest on ground concrete slab immediately or rest on other types of supports.

In selecting design criteria and design method, considerations should be made to a myriad of factors including dimensions, location, ground condition, support condition, groundwater conditions, aesthetic properties, design life, exposure condition, usage, roofing, availability of construction materials. According to Code of Practice BS8007, two main classes of limit state are considered. Ultimate limit state is design against structural failure, including bending moment check and shear force check. Serviceability limit state is design against deflection and crack width. Normal crack width control is 0.2mm while, for severe cases, allowable crack width is 0.1mm. For underground liquid retaining structures, serviceability limit state design is also used in checking of bearing capacity of soil. In the design, factors of safety are involved to increase the structural reliability.

3 Features of KBS

KBS can be defined as an interactive computer system that incorporates expertise and provides advice on a wide range of tasks. It solves a specific complex problem mimicking the decision making and reasoning processes that resemble those of human experts. Intelligent tutoring system is a KBS with the purpose of instruction and teaching. The system has to engage the user in a dialogue systematically and actively. Solely explanations are not sufficient and the problem should be dealt with through interactive communication between the user and the system.

These systems typically consist of the following three basic components, namely, knowledge base, context, and inference mechanism. The heart and core of any KBS is the knowledge base, which is usually a collection of rules, typically in the form of IF….THEN….. The knowledge base is a collection of general facts, rules of thumb and knowledge specific to the problem domain. Other forms of representations commonly used are logic, frame-based schemes, nets, and the object-oriented approach. The context is a workspace for the problem constructed by the inference mechanism from the information provided by the user and the knowledge base. It contains facts that reflect the current state of the problem. The organization of the context depends on the nature of the problem domain. The context builds up dynamically as the problem domain is being considered. The context is used by the inference mechanism to guide the decision making process. The inference mechanism monitors the execution of the program by using the knowledge base to modify the context. Moreover, it manipulates the context using the knowledge base.

Apart from the three main modules described above, the system should also be provided with three other components that are not necessarily part of every KBS but are desirable in an integral final product, namely, a friendly user interface, an explanation facility, and a knowledge acquisition module. The function of the user interface module is to accept a problem description from the user and to interact with the rest of the system in order to analyze the problem or augment the capability of the system. It provides an interface between the user and the KBS, usually as a command

language for directing execution. The interface is responsible for translating the input as specified by the user to the form used by the KBS and for handling the interaction during the decision making process. The explanation module provides explanations of the inferences used by the KBS. This explanation can be *a priori* – why a certain fact is requested, or *a posteriori* – how a conclusion was reached. The knowledge acquisition module, which provides a means for entering domain specific knowledge into the knowledge base and revising this knowledge when necessary, serves as an interface between the experts and the KBS.

When compared to KBS, conventional programs are very inflexible since they consist of a set of statements with predetermined order of execution. Their updates need considerable effort, because the programmer has to locate the appropriate place to update in the predefined sequence. The programmer must ensure completeness, namely, that the program performs correctly for all possible combination of conditions, and uniqueness of the solution, namely, that the output is unique for every possible input. The user perceives the program as a blackbox, where the program generates results for the input provided; one does not have any idea as to why the program has produced certain results. KBS eliminates some impediments posed by conventional programs by making a clear distinction between the knowledge base and the control strategy. This partitioning allows for incremental addition of knowledge, without manipulating the overall program structure; the programmer needs not guarantee completeness. Further, by ranking several alternatives either by an evaluation scheme or by the use of inexact inference methods, several solutions can be provided for a particular set of input conditions, thus relaxing the uniqueness constraint. The user can also question the results produced by the program through the explanation module.

4 Architecture of Prototype KBS

In this prototype system, the blackboard architecture, which is capable of supporting the development of systems in domains characterized by the interaction between diverse sources of knowledge, is adopted. The blackboard architecture has been successfully used in solving a wide range of tasks, such as speech recognition, signal processing, and planning. It provides a framework for integrating knowledge from several sources. The blackboard serves as a global data structure, which facilitates this interaction. A common analogy may be made to problem-solving in domains where a number of experts in different areas of specialties co-operate over the solution which any one of them could never achieve alone. In order to facilitate this process, they agree to use a blackboard to post any partial result they can contribute separately. Each expert takes turns to write on the blackboard and, in case more experts wish to write simultaneously, the conflict is resolved by some pre-defined strategy. Because of the modularity of knowledge sources, the blackboard architecture enables easy incremental development of a software system and developers can integrate different methods of knowledge representation in a single system. In a typical blackboard system, a number of knowledge sources communicate through a blackboard and are controlled by an inference mechanism.

A number of knowledge sources containing the domain knowledge constitute the knowledge base. These knowledge sources are independent chunks of knowledge and do not directly communicate with each other. Instead, they participate in the problem solving process by creating entries in a global database – the blackboard. Knowledge modules look at the blackboard to see if suitable data is present to trigger their execution. If they are selected, the execution results in new or altered data on the blackboard, which will then trigger other knowledge modules. The key to the solution by using the blackboard architecture is the cooperation of the knowledge modules present. Each knowledge source consists of a condition-action pair. Whenever a condition is satisfied in the blackboard, the action in the corresponding pair is executed accordingly.

The context comprises the information or entries generated by the knowledge sources during the problem solving process. Entries are the immediate results produced by the system. In a typical system, each entry has a certainty factor as well as a specification. The principal units in the blackboard are hypotheses, which are either primary guesses about particular aspects of the problem or partial solutions. Hypotheses at various levels are related through structural relationships. The blackboard is organized into a number of levels each representing different aspects or stages of the solution process. These levels depict an *a priori* plan for the solution of a problem that can be naturally decomposed into a set of levels. Each level contains objects and attributes that are important to the representation of the problem. Normally, knowledge sources are specific to certain levels in the blackboard, namely, the activation of a certain knowledge sources depends on the entries generated at certain levels in the blackboard, while the actions of the knowledge source modify entries at some other level.

Two main components, namely, the agenda and the monitor constitute the inference mechanism. The agenda keeps track of all the events in the blackboard and calculates the priority of execution for knowledge sources that were generated as a result of the activation of other knowledge sources. It is a list of knowledge sources or rules to be executed in the next cycle. Based on the success or failure of a particular rule, new rules may get added on to it or some may be deleted from it. The basis of giving priorities to the rules on the agenda may vary from system to system. Several problem-solving strategies can be implemented using the monitor. The monitor takes the element with the highest priority and executes it.

5 Knowledge Representation for Structural Design

Structural analysis can be delineated as a three-stage process involving modeling, solution, and evaluation. Before knowledge can be represented in structural analysis, the type of knowledge involved must be identified and classified. Static knowledge comprises definitions, axioms, and laws. These may be *a priori* or the result of scientific investigation. Static knowledge is ‘deep’ in that it is deduced from fundamentals. Dynamic knowledge is not deducible from any axiom, rather it is generally gained from experience. It refers to heuristics, which is related to the process of search or to knowledge based on experience. Dynamic knowledge can also

be described as ‘shallow’, meaning that it cannot provide us with explanations of why certain decisions should be made.

The major approaches for declarative representation of knowledge in the AI literature are rule-based production system, frames and object-oriented programming. A production system is a collection of rules and is believed to be good at describing heuristic knowledge. For KBS developers, rule-based system tends to be more easily understood and thus accepted. A frame system, on the other hand, is suitable for a complex and rich representation of knowledge, such as static knowledge. Object-oriented programming concept is used, in which a computer program consists of a number of independent objects that process jobs by exchanging information they need via messages. Because of its modularity, data abstraction and inheritance characteristics, object-oriented programming will likely subsume other approaches in the very near future. To apply object-oriented program development, data representations for the model must be specified. There are three steps in the development of an object-oriented program, namely, selection of classes, specification of classes, and implementation of classes. Here, the word ‘class’ refers to a description of a set of similar objects whilst one member object in a class is called an ‘instance’. It is appropriate to utilize both representations together to solve structural design problems since it may take the advantages of both approaches.

6 Prototype Web-Based KBS

Expert system programming environments or shells are often employed in order to facilitate the development of KBS. These system shells contain specific representation methods and inference mechanisms. The knowledge base and explanation facility of this prototype system have been developed using a commercially available expert system shell called Visual Rule Studio which is a hybrid application development tool that integrates object-oriented techniques and expert system technology with traditional, procedural programming. Visual Rule Studio installs as an integral part of Microsoft Visual Basic 6.0 and appears within Visual Basic as an ActiveX Designer. As a part of the Visual Basic Integrated Development Environment, using a RuleSet in the application is similar to using a form or other Visual basic Designer. Besides, Visual Rule Studio is compatible with Microsoft Internet Information Server and Active Server Pages. As such, in order to allow it to reach any user with a web browser and Intranet or Internet access, the Ruleset components can be deployed as part of a web server based application.

The prototype system combines expert systems technologies, object-oriented programming, relational database models and hypertext/graphics in a windowing environment. It runs under and follows the conventions of Microsoft Windows. In a windowing system, any types of display windows can be represented as objects, each with its own private data or information. By defining various types of windows as different classes, such as checkbox group, hyperregion, promptbox, pushbutton, textbook, etc., they can inherit common characteristics and/or possess their own special properties. By isolating rules as component objects, separated from objects and application logic, Visual Rule Studio allows developers to leverage the proven

productivity of today's component oriented development tools, such as Visual Basic. With Visual Rule Studio, rule development becomes a natural part of the component architecture development process. The complex and time-consuming problems of integrating multiple development tools and managing incompatible object models no longer exist. Visual Rule Studio becomes an integrated part of the Visual Basic development and produces objects that can interact with virtually any modern development software.

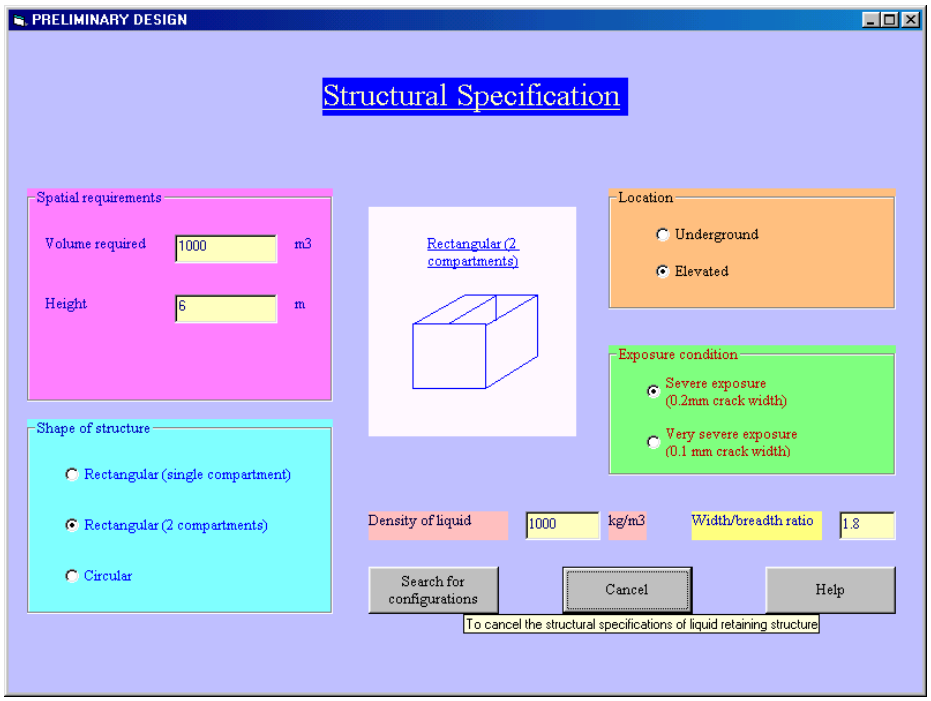

Fig. 1. Screen displaying structural specification of liquid retaining structure

In the system, Visual Rule Studio objects are employed to encapsulate knowledge structure, procedures, and values. An object's structure is defined by its class and attribute declarations within a RuleSet. Object behavior is tightly bound to attributes in the form of facets, methods, rules, and demons. Each attribute of a class has a specific attribute type. The Visual Rule Studio attribute types are compound, multicompound, instance reference, numeric, simple, string, interval, and time. Each attribute can have many facets and methods associated with it. Facets provide control over how the inference engines process and use attributes. Methods establish developer-defined procedures associated with each attribute.

Knowledge acquisition of the domain knowledge is basically from written documents such as codes of practice, textbooks and design manuals and complemented by experienced engineers involved with the design of liquid retaining

structures. The domain knowledge is translated into procedures and methods using object-oriented representation. The system is compiled and encrypted to create a run-only system on a web server. The user can always overrule any design options and recommendations furnished by the system, thus playing solely the role of a knowledgeable assistant. The design is still under full control of the user.

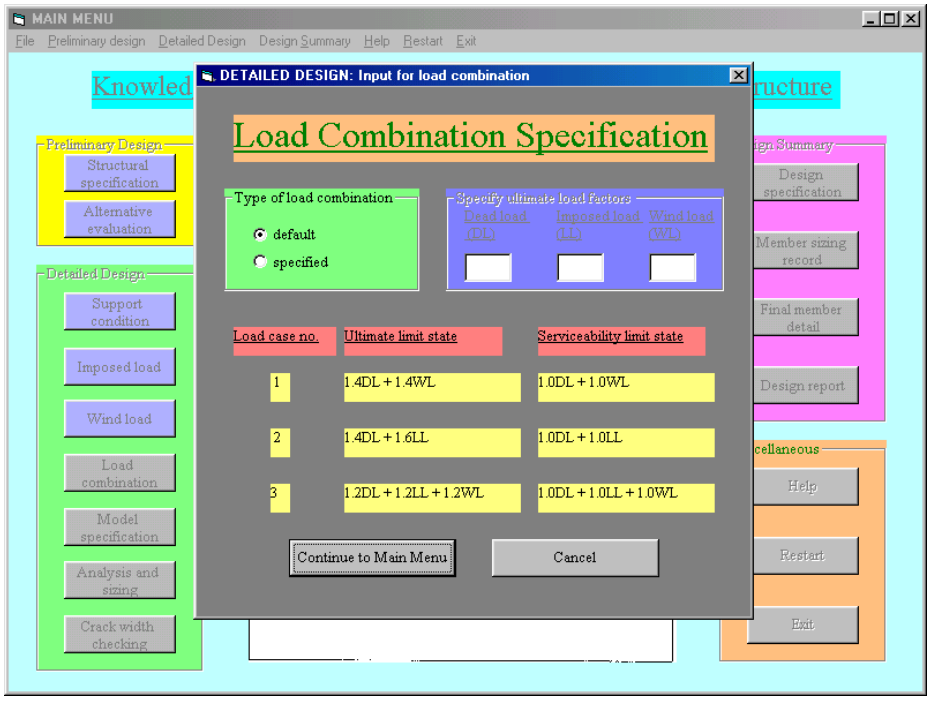

Fig. 2 Screen displaying specification on load combination

The inference strategies model the reasoning processes an expert uses when solving a problem. The Visual Rule Studio inference engines control the strategies that determine how, from where, and in what order a knowledge base draws its conclusions. It supports three types of inferencing strategies, namely, backward chaining, forward chaining, and hybrid chaining. Each of these inferencing strategies acts on specific knowledge base components. A mixed problem-solving strategy combining both forward chaining and backward chaining inference mechanism is employed in this knowledge module level. The user is required merely to supply the relevant data during each design stage and the system will determine the order in which different design knowledge modules are executed.

Besides, the system offers a state-of-the-art user interface. The use of a mouse or other pointing device makes the data entry a simple task even for novice computer users. As such, users simply point and click their way through the process to appreciate the dynamic behavior of the system. Input data entry is kept at minimum. Input data are provided by the user mostly through selection of appropriate values of

parameters from the menus and providing answers to the queries made by the system. It provides information about any individual member in multi-window graphics text display where graphic images are combined with valuable textual information. This kind of intelligent graphics is extremely valuable to structural designers because it enhances their confidence in the design provided by the KBS. Figure 1 shows the screen displaying structural specification of liquid retaining structure. The input data provided by the user is rejected if it is not within the range specified. It can explain its line of reasoning for obtaining an answer. Figure 2 shows the screen displaying specification on load combination. Figure 3 shows the screen displaying finite element analysis of liquid retaining structure.

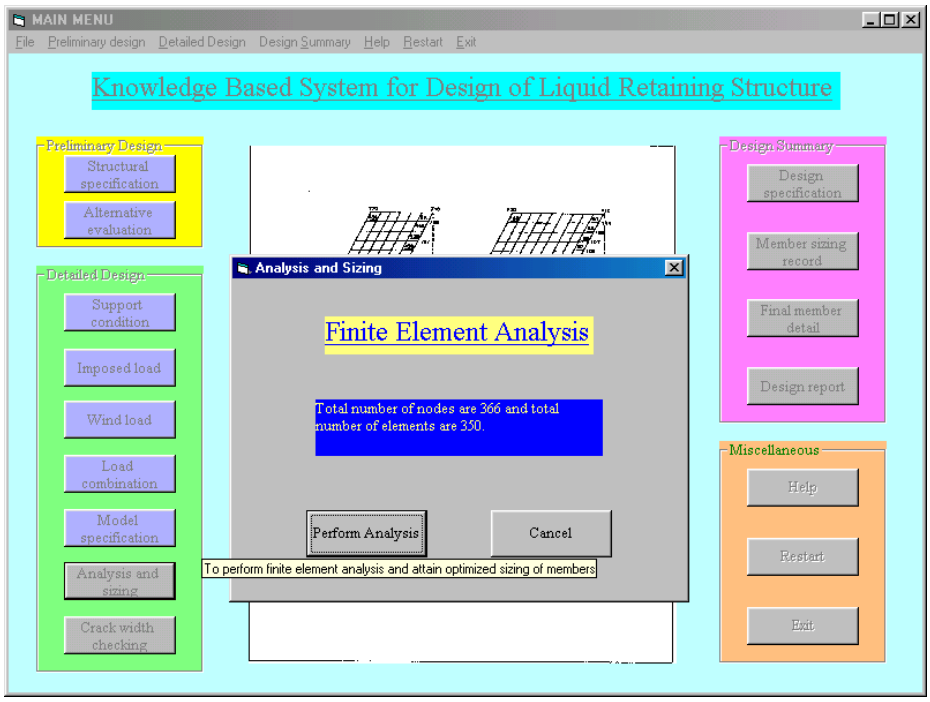

Fig. 3. Screen displaying finite element analysis of liquid retaining structure

7 KBS as Instructional Tool

It is generally acknowledged that one-to-one tutoring is the most effective teaching and training method, yet requires vast amount of resources such as instructors. Nowadays, personal computer is very popular in Hong Kong. Those who have a computer and a telephone line can have a tutor at home or in office by accessing the web-based intelligent tutoring system through the Internet. Intelligent tutoring system,

being one type of KBS with particular purpose of teaching, can assist novice engineers or civil engineering students to acquire deeper understanding of the topic through the use of this system. Explanations are made to assist them to learn and capture how to design liquid retaining structures effectively and sustainably in their design practices. One of the key advantages in the implementation of the prototype system is to reduce the dependence on experienced designers for routine design works. In this way, they will have more precious time left and can concentrate their effort on other innovative and creative civil engineering designs.

It has been shown in this application that the flexibility and open infrastructure of Internet are capable to perform its major role as a medium in teaching and learning processes. Students or novice engineers can easily get hold of domain knowledge on both theory and design of liquid retaining structure through interactive communication with this system. Furthermore, structural optimization and best practical design on liquid retaining structure may be accomplished or at least improved substantially by the exploration of the interactive "What-if" scenario analysis. Engineering professional institutions and their senior fellows should acknowledge that the world is changing in a fast pace and that both engineering companies and novice engineers have practical needs that cannot be fulfilled for all aspects under the current educational and training format. The extant prevalent situation with tight financial constraints, as well as both global and private competition amongst construction profession as a whole, may not just happen as a short-term effect. Under the current situation, even if all the members adhere to the principle of value-added resources, the adversity may not be easily overcome. Hence, in order to cope with these challenges, new methods with the aim of delivering quality training to novice engineers may be entailed. Amongst a variety of feasible solutions that can lead to solution of some of these problems, web-based learning is one that is worthy of extensive investment, application as well as implementation. In the near future, the integration of information technology together with other methods will alter significantly the nature of the teaching and learning process. It is foreseeable that web-based learning, which has the potential to effect fundamental changes in the design of learning processes and the pedagogical system, has been gaining momentum with an irreversible trend.

8 Conclusions

In this paper, it is shown that a web-based KBS employing the hybrid knowledge representation approach, which combines production rule system and object-oriented programming technique to the design of liquid retaining structures is feasible with the implementation of blackboard system architecture. It is appropriate to integrate algorithmic and symbolic programming on structural design into a single computer-aided instructional tool running under a Windows platform. With the widespread popularity of Internet nowadays, the use of web-based KBS in training novice engineers or in transferring knowledge can have great potential.

References

1. Chau, K.W., Lee, S.T.: Computer Aided Design Package `RCTANK' for Analysis and Design of Reinforced Concrete Tanks. Computers and Structures **41(4)** (1991) 789-799
2. Kitzmiller, C.T., Kowalik, J.S.: Coupling Symbolic and Numeric Computing in Knowledge-Based Systems. AI Magazine **Summer** (1987) 5-90
3. Chau, K.W.: An Expert System for the Design of Gravity-type Vertical Seawalls. Engineering Applications of Artificial Intelligence **5(4)** (1992) 363-367
4. Chau, K.W., Albermani, F.: Expert System Application on Preliminary Design of Liquid Retaining Structures. Expert Systems with Applications **22(2)** (2002) 169-178
5. Chau, K.W., Chen, W.: An Example of Expert System on Numerical Modelling System in Coastal Processes. Advances in Engineering Software **32(9)** (2001) 695-703
6. Chau, K.W., Cheng, Chuntian, Li, C.W.: Knowledge Management System on Flow and Water Quality Modeling. Expert Systems with Applications **22(4)** (2002) 321-330
7. Chau, K.W., Ng, V.: A Knowledge-based Expert System for Design of Thrust Blocks for Water Pipelines in Hong Kong. Water Supply Research and Industry - Aqua **45(2)** (1996) 96-99
8. Chau, K.W., Yang, W.W.: A Knowledge-Based Expert System for Unsteady Open Channel Flow. Engineering Applications of Artificial Intelligence **5(5)** (1992) 425-430
9. Chau, K.W., Yang, W.W.: Development of An Integrated Expert System for Fluvial Hydrodynamics. Advances in Engineering Software **17(3)** (1993) 165-172
10. Chau, K.W., Zhang, X.N.: An Expert System for Flow Routing in a River Network. Advances in Engineering Software **22(3)** (1995) 139-146
11. Lin, S., Albermani, F.: Lattice-Dome Design Using a Knowledge-based System Approach. Computer-Aided Civil and Infrastructure Engineering **16(4)** (2001) 268-286
12. British Standards Institution: BS 8007: Design of Concrete Structures for Retaining Aqueous Liquids. British Standards Institution, London (1987)
13. British Standards Institution: BS 8110: Structural Use of Concrete. British Standards Institution, London (1985)