Zeitschrift: IABSE reports = Rapports AIPC = IVBH Berichte

Band: 58 (1989)

Artikel: Knowledge-based finite element analysis

Autor: Stiemer, Siegfried F. / Forde, Bruce W.R.

DOI: https://doi.org/10.5169/seals-44922

Knowledge-Based Finite Element Analysis

Analyse par éléments finis par base de connaissance

Wissensbasierte Finite-Element-Berechnung

Siegfried F. STIEMER
Associate Professor
University of British Columbia
Vancouver, BC, Canada

Bruce W.R. FORDE
Research Assistant
University of British Columbia
Vancouver, BC, Canada

Siegfried F. Stiemer completed his studies leading to a Dipl. Ing. Degree at the University of Stuttgart in 1972. He spent two post-doctoral years at the University of California in Berkeley before joining the Faculty of Engineering of the University of British Columbia in 1981. His various research interests range from experimental studies to computational applications in the area of steel design.

Bruce W.R. Forde recently completed a Ph.D. degree in a cooperative programme at the University of Stuttgart, West Germany and the University of British Columbia, Canada. His present research interests focus on artificial intelligence applications to computer-aided engineering.

SUMMARY
This paper describes some innovative uses of knowledge-based techniques for problems such as process control and solution optimization. A simple displacement computation scenario is used to demonstrate the natural and flexible methods offered by knowledge-based systems for controlling interactive analysis and for dynamically formulating problem-solving approaches. Emphasis is placed on finite element analysis applications.

RESUME
Cette publication décrit quelques utilisations innovatrices des techniques de bases de connaissance pour les problèmes liés au contrôle de procédés et à l'optimisation de solutions. L'exemple simple d'un calcul de déplacement est utilisé pour démontrer la méthode naturelle et flexible qu'offre le système de base de connaissance pour contrôler les analyses interactives et pour formuler des approaches de résolution de problèmes de façon dynamique. L'accent est mis sur des applications d'analyse par élément finis.

ZUSAMMENFASSUNG
Dieser Beitrag beschreibt einige innovative Anwendungen von wissensbasierten Methoden auf Probleme wie Prozesskontrolle und Lösungsoptimierung. Ein einfaches Verschiebungsberechnungs-Szenario wird benutzt, um darzustellen wie natürlich und flexibel wissensbasierte Systeme interaktive Berechnungen und dynamische Formulierung von Problemlösungswegen ermöglichen. Der Schwerpunkt wird auf eine Anwendung der Berechnung mit finiten Elementen gelegt.

1. INTRODUCTION

Almost all existing applications of artificial intelligence in engineering belong to a group of elementary products called knowledge-based expert systems (KBES) — programs that mimic the decision-making capabilities of an expert in a particular domain. The popularity of the KBES approach can be attributed to the development of expert system shells that provide simple consultation environments with rule-based knowledge representation and manipulation. This paper demonstrates some of the more recent innovative uses of knowledge-based techniques for problems such as process control and solution optimization applied to finite element analysis.

Traditional algorithm-oriented methods constrain the logic, data structures, and procedures found in engineering analysis. Subsequent sections explain the nature of this problem and show how knowledge-based methods can simplify and improve the analysis process.

1.1. Traditional Algorithm-Oriented Methods

Finite element analysis programs usually employ algorithms. The purpose of these algorithms is to provide structured patterns that may be repeatedly applied in the solution of a specific kind of problem. This approach can provide very efficient results for rigidly structured problems that are defined in terms of simple logic and data structures such as found in many numerical processes. Early batch-job finite element software used the algorithm-oriented approach and was successful due to the rigid nature of the problem — exactly how and when to compute data is decided in advance, so the logic associated with these decisions may be rigidly embedded into the application software during the program implementation. Conversely, interactive finite element analysis programs allow the end-user to arbitrarily enter data and to experiment with alternatives, so the analysis process must be dynamically altered at run-time. Many possible combinations of data and procedures are required, so the associated logic results in an extremely large and complex algorithm.

Some basic problems with the algorithm-oriented approach are illustrated by the following simple scenãrio. Linear structural analysis programs typically employ a displacement computation algorithm like that shown in Fig. 1. Components of this algorithm include data ([K], {R},{D}) and procedures (*Read* Input, *Assemble* [K], *Assemble* {R}, *Compute* {D}, *Write* Output). The data may be viewed as a set of facts that determine the current problem context, and the procedures are tools that organize these facts into more useful forms.

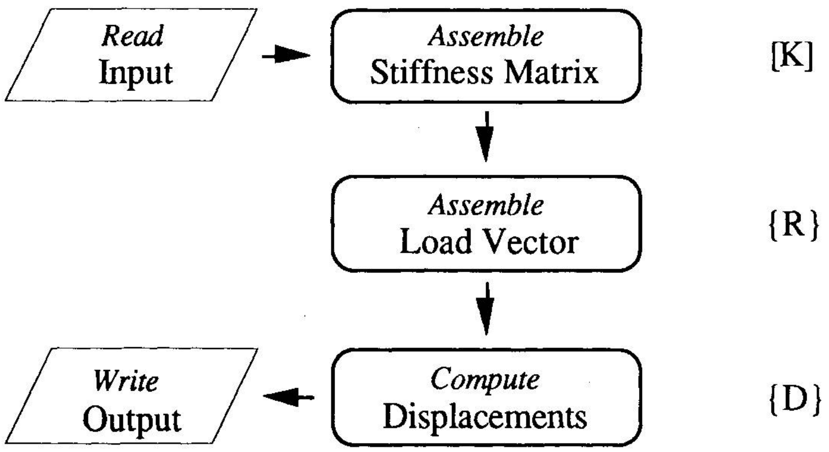

Fig. 1 Displacement Computation Algorithm.

This algorithm provides additional logic that shows how and when to invoke procedures to uncover desired facts. For example, this algorithm indicates a sequence of: *Assemble* [K], *Assemble* {R}, and *Compute* {D}. The order of activities is important and non-unique. Displacements cannot be computed before stiffness and load assembly; however, loads can be assembled before the stiffness giving another valid activity sequence of: *Assemble* [R], *Assemble* {K}, and *Compute* {D}.

Algorithm designers can arbitrarily select either of these alternatives, but in doing so they impose an unnecessary constraint on the analysis process (stiffness and load assembly are independent processes and one does not necessarily take place before the other).

In addition to the implied activity sequence, algorithms place bounds on the overall problem-solving approach. By adopting a specific algorithm, the program developer accepts the associated logic. For example, the displacement computation problem could be approached using several alternative formulations in place of the given displacement-based matrix format. Since this approach requires a stiffness matrix and a load vector, these components become part of the problem. The original logic employed by the program developer in deciding how and when computation should take place is embedded in the algorithm. If a new analysis procedure is found in the future, the developer must reexamine the logic behind the algorithm in light of the new set of alternatives.

1.2. Knowledge-Based Methods

Knowledge-based expert systems have been used in many engineering applications, but primarily as consultants or intelligent-interfaces [1, 2]. There are many other potential KBES applications within engineering software such as control and optimization. Knowledge-based approaches provide more natural and flexible methods of applying control logic for interactive analysis, as well as the potential for dynamic formulation of problem-solving approaches, that would not be possible with existing algorithms [3]. Problems associated with this new approach involve the integration of logic processing and analysis computation, as well as the representation of appropriate knowledge for the analysis domain. For example, control logic associated with the displacement computation problem can be summarized by a single rule (**IF** have [K] & have {R} **THEN** can compute {D}). A KBES that uses this rule would recognize that stiffness and load assembly are processes that take place before displacement computation (avoids unnecessary restrictions imposed by an algorithm).

Advantages of the knowledge-based approach over the traditional algorithm-oriented approach are amplified in problem areas where many options are available due to complex logic or changing technology. If the purpose of algorithms is to provide structured patterns that may be repeatedly applied in the solution of a specific kind of problem, this approach is bound to be inefficient in cases where the problem is not well defined and the desired solution pattern is constantly changing to maintain the state-of-the-art. Some aspects of the finite element analysis process remain relatively unchanged (eg. numerical solvers), yet others are constantly being improved to accommodate new problem types and solution procedures. A combination of algorithm-oriented and knowledge-based approaches is required to achieve efficient solutions in modern software.

1.3. Objectives and Organization

This paper shows that knowledge-based techniques can be used for a variety of tasks in the finite element analysis process aside from the popular roles as consultant and interpreter. Some of the problems examined include:

(1) Potential KBES applications to finite element analysis.
(2) Knowledge-based expert systems for process control.
(3) Dynamic formulation of problem-solving approaches.

The second chapter of this paper identifies a variety of potential KBES applications to finite element analysis including: consultation, interface, control, and optimization. These topics involve logic and decision-making that is easily handled using heuristics in place of traditional algorithms. Simple descriptions are given for each application in terms of the problems handled and the potential computer software configurations.

The third chapter of this paper examines a simple KBES that controls an interactive finite element analysis program. The kinds of data and procedures found in the analysis process, the relationships between these components, and the factors that make the knowledge-based approach attractive are briefly introduced. Some basic methods are discussed for rule-based representation, followed by a summary of alternative techniques for reasoning and a comparison of conventional expert systems to a KBES used to manage an interactive computation process.

The fourth chapter of this paper discusses the potential implementation of a KBES for dynamic formulation of problem-solving approaches. Rather than relying on any single algorithm with its predetermined set of procedures for computation, the proposed system employs logic from applicable technologies to construct its own algorithms, data structures, and procedures as required to achieve the best solution for any given problem.

2. KBES APPLICATIONS TO FINITE ELEMENT ANALYSIS

This chapter identifies a variety of potential KBES applications to finite element analysis including: consultation, interface, control, and optimization. These topics involve logic and decision-making that is easily handled using heuristics in place of traditional algorithms. Simple descriptions are given for each application and the potential computer software configurations.

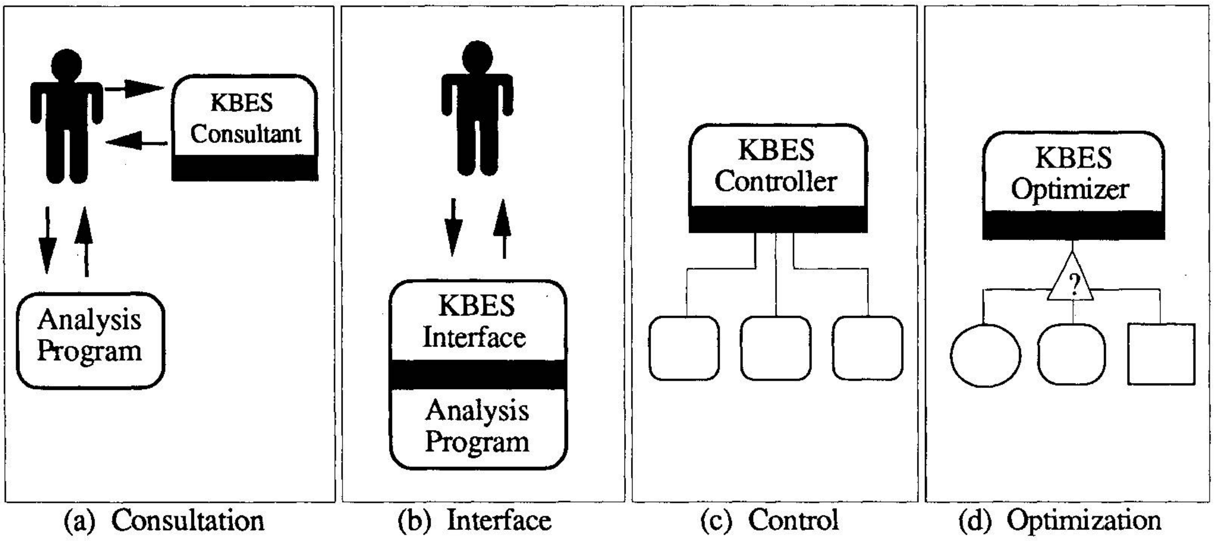

Fig. 2 KBES Applications to Finite Element Analysis.

2.1. Consultation

One of the first applications of knowledge-based technology in engineering was the introduction of automated consultants that help engineers in the use of complex computer programs. Engineering software training often involves on-the-job expert consultation to supplement the explanations given in computer manuals. Experienced engineers often do not have time to answer questions for junior personnel, so an automated consultant as shown in Fig. 2(a) is desired to work in parallel with the user of an analysis program. Early systems such as SACON [4], functioned in this way to help explain input and output files for traditional batch-oriented finite element analysis. Expert system shell consultation environments, with facilities for representation and manipulation of rule-based knowledge, are the most popular KBES technology used in engineering.

2.2. Interface

A second generation of knowledge-based technology emerged in response to problems with the consultation approach. These systems recognized that automated consultation is not always effective due to limited understanding of complex conventional application software. Even with excellent advice from a consultant, engineers inevitably find difficulties with program operation. A KBES can provide a simplified interface to engineering software [5]. Systems such as shown in Fig. 2(b) translate user-requests into facts understood by the intelligent interface. These facts are used in conjunction with a knowledge base and an inference mechanism to logically evaluate and implement operations that will achieve user-requests.

2.3. Control

Knowledge-based technology has recently been applied to problems that are traditionally handled by algorithms within conventional software. One example application is as the controller of an interactive finite element analysis program [6]. The system shown in Fig. 2(c) uses a KBES to select and apply a variety of analysis computations. Interactive software was made popular by the spreadsheet concept — changes made in one cell automatically propagate throughout the spreadsheet. Similar capabilities in an analysis program require integration of modelling, solution, and interpretation activities. Computation of all results for each change is not necessary and is impractical in terms of real-time operation, so some decisions must be made at run-time. The KBES approach is attractive in this case because only a few simple heuristics are required to describe the analysis process even for an arbitrary number and order of actions. Conversely, equivalent algorithm-oriented approaches would be more complex to implement and even more difficult to improve or upgrade in light of new facts or strategies.

2.4. Optimization

Engineering technology relies heavily on experience to assure that the best solutions are used to obtain the best end-product. Heuristics or rules-of-thumb are traditionally incorporated within engineering software to achieve practical results, yet the solution procedure itself is defined by a specific fixed algorithm. Optimization of the end-product is often equated with the selection of a few possible alternatives followed by evaluation and comparison using objective functions. This approach can be built into algorithms, but doing so constrains the potential solution domain. KBES technology makes it possible to perform optimization without limiting the number or type of alternatives, as the data defining facts and strategies may be part of a changing knowledge-base. Although product design optimization can be improved using KBES approaches, this paper focuses on solution optimization that was not possible using algorithm-oriented methods. Unlike traditional programs that simply select and apply a variety of analysis procedures, the system shown in Fig. 2(d) uses a KBES to dynamically formulate the problem-solving strategy. A finite element analysis program operating with this paradigm can decide when and how to store and compute data. Whereas traditional software employs a "brute-force" approach for many numerical problems, new systems can use knowledge to take advantage of common computations and to share results as humans would do when solving problems by hand calculation.

3. KNOWLEDGE-BASED PROCESS CONTROL

Control logic can be separated from analysis software, and a knowledge-based expert system can use this logic to perform interactive computation. Traditional algorithms for finite element analysis can be replaced by a few simple heuristics that define how and when computation should take place. Rule-based techniques are used to represent and to reason about this analysis domain knowledge. Subsequent sections explain why the KBES approach is more effective and flexible than traditional algorithm-oriented approaches.

3.1. Domain Knowledge

Finite element analysis is a three-stage process of: modelling, solution, and interpretation [3]. The components of a simple analysis problem vary depending on the particular domain, so this paper will restrict discussion to the example shown in Fig. 3.

Knowledge related to modelling and interpretation deals with questions such as how to select elements that will properly represent the physical problem, and how to recognize when numerical results correspond to real behaviour. These questions are subjective and difficult for junior engineers to answer, as they rely heavily on experience, so early KBES consultants focused on heuristics of this nature. On the other hand, control is a relatively simple topic. In order to achieve a solution, analysis data must be used in structured processes. As in mathematics, explicit logical answers are often available for questions about control. For example, stiffness and loads must be assembled before displacements can be computed. Such knowledge can be easily represented using rules and then used by an inference mechanism to reason about specific problems.

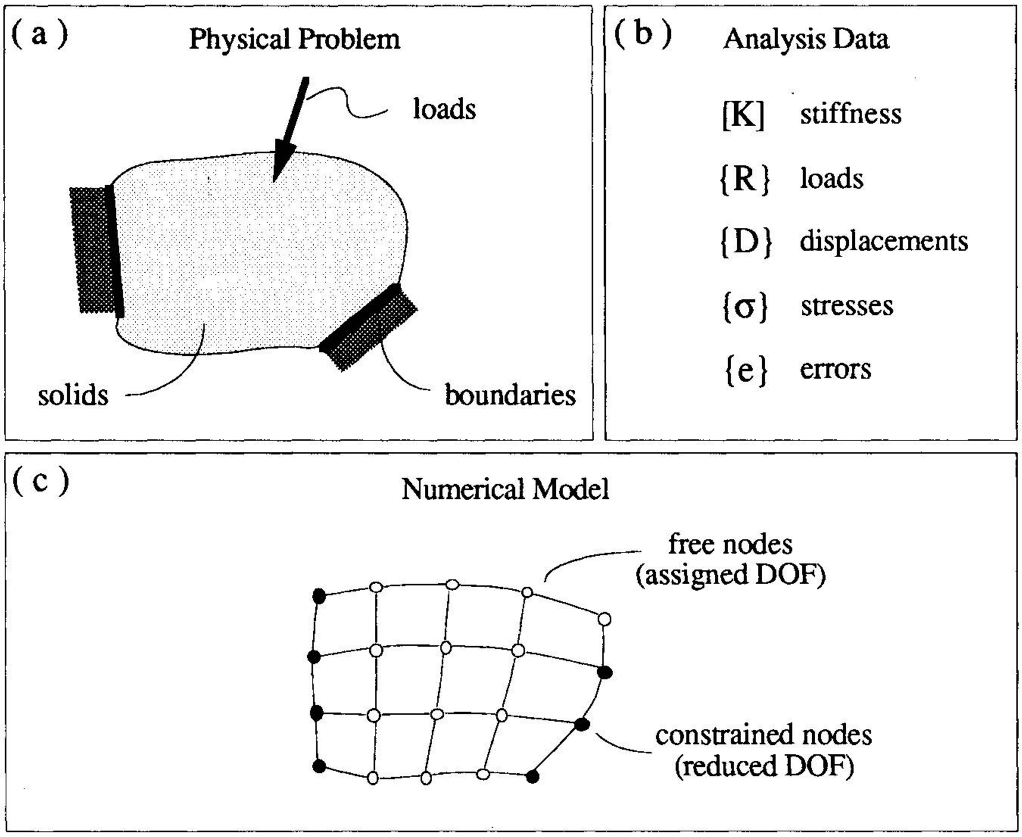

Fig. 3 Components of a Simple Analysis Problem.

3.2. Representation

Engineers that are familiar with conventional approaches recognize that they have encoded their knowledge into the algorithms that control their programs. There are two major problems with this approach. First, expert knowledge (facts or rules) that is built into an algorithm cannot be questioned or updated by the program. Second, traditional data structures and procedures used to describe the problem or its solution place unnecessary constraints on the problem-solving approach.

Systems are called "knowledge-based" if they separate domain knowledge from the program control structure. This allows KBES programs to question and to update their own knowledge base of facts and rules. Most engineering expert systems do not actually employ machine learning with automatic updating of knowledge bases, but explicit definition of domain knowledge is a step in this direction in comparison to previous approaches.

Knowledge about the problem-solving approach is often subtly implied by the data structures and procedures linked to algorithms. For example, algorithms found in finite element analysis often make references to matrices that hold information about the problem. Although these data structures may be arbitrarily sized, the fact that there is any mention of matrices implies the kind of solution. Object-oriented programming is becoming popular because it encapsulates data and procedures so that problem-specific information may be placed in packages. A similar trend is appearing in the design of knowledge-based systems leading to islands of knowledge with local KBES management.

3.3. Reasoning

Most expert systems use a form of deductive inference to reveal facts. Some theorem-proving systems start by examining rule premises. If all conditions in a rule premise have been satisfied, the rule hypothesis may be deduced. This new hypothesis may be used with other known facts to prove additional rules. Since this approach works directly through the rules from conditions to hypotheses, it is known as "forward-chaining" inference. An alternative strategy involves searching rule hypotheses for a desired goal. Once a rule has been located with the appropriate hypothesis, each of its conditions are examined. Conditions that are not proven are added to the list of goals. If all conditions have been proven true, the rule hypothesis is also proven true. Eventually, the original goal is achieved or additional information is requested. Since this approach works from hypotheses back to conditions, it is known as "backward-chaining" inference. Real expert systems employ a mixture of the two approaches.

Reasoning is performed by an inference engine as shown in the knowledge-based control portion of Fig. 4. Replacing the logic that was embedded into a traditional algorithm-based approach with explicit facts and rules in a knowledge-based system allows program developers to construct applications around a generic event-driven core (a popular paradigm for interactive software). Instead of invoking computation in a rigid order, as mentioned earlier, the KBES approach can make logical decisions based on the current problem context and knowledge base.

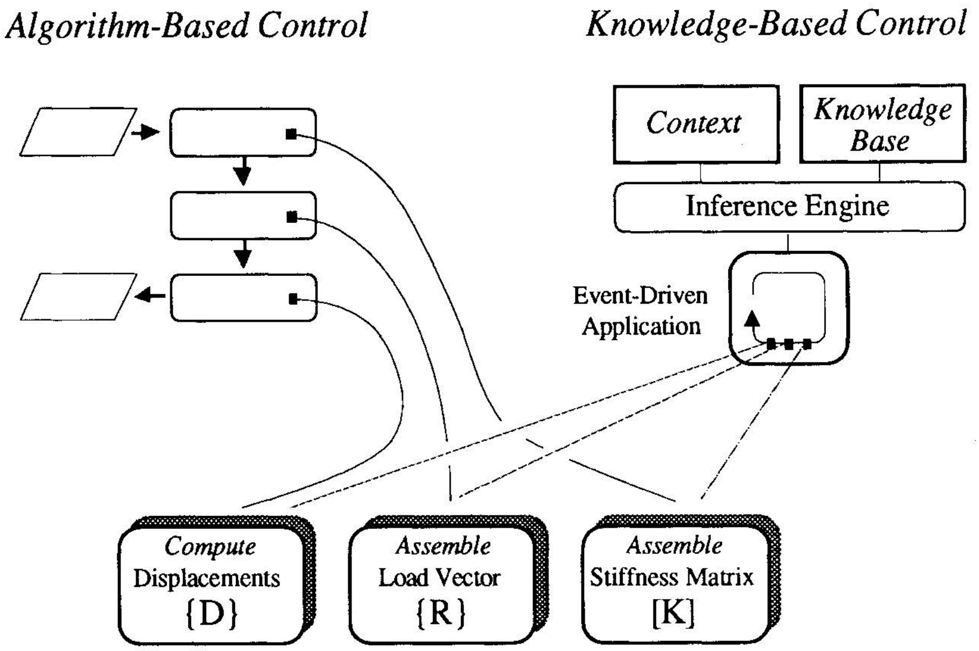

Fig. 4 Alternative Control Paradigms.

4. KNOWLEDGE-BASED SOLUTION OPTIMIZATION

Analysis programs designed and implemented by experts may provide efficient solutions to current problems; however, increasing specialization and advancing technology will eventually make it impossible to achieve optimal analysis using traditional methods. The problem is that algorithm-oriented techniques require software developers to formalize the complete analysis process in terms of a specific problem-solving strategy, a limited number of procedures, and a predetermined set of data structures. The proposed solution is to automate the selection and application of low-level analysis components using a knowledge-based system that can organize and solve equations in the same manner as an expert.

For a simple finite element analysis program, this involves the representation of equations for stiffness, displacement, stress, and error computation. Using its knowledge base, the equation-solving expert-system should be able to identify: the order in which the equations are applied, where the data is stored, and what calculations can be used in multiple contexts (eg. stiffness, stress, and error computations share certain intermediate matrices). This new approach must take advantage of the significant number of efficient general-purpose procedures currently available for conventional software development. Object-oriented programming can be used to create a hybrid environment that integrates the best features of KBES technology for reasoning and representation with the existing base of numerically efficient procedure-oriented software.

4.1. Object-Oriented Design

Computer software development involves the implementation of an application or concept in a programming environment. Finite element analysis programs are usually built around concepts of nodes, elements, stiffness matrices, etc. Although software of this kind is often called modular and reusable by its developers, this is usually not the case. Conventional finite element procedures (subroutines) assume many facts about their data structures. A typical assumption may be that nodes have a certain number of degrees of freedom. In the event that future software has a different kind of node, many procedures may not be used without modification. The intention of object-oriented programming is to encapsulate data and procedures into useful packages that provide a higher degree of modularity than traditional software. These packages are the building blocks used to dynamically construct solutions based on the problem context [3].

4.2. Optimization Strategy

Rather than starting program construction with a fixed problem-solving strategy in mind, a knowledge-based system can be used to evaluate the suitability of available object-oriented components and to assemble them into a specialized form for any given problem. Conventional programs place emphasis on designing a robust algorithm that will efficiently solve a specific problem type. Conversely, this system places emphasis on general representation and reasoning capabilities. Domain knowledge, including the problem-solving strategies, is maintained separately from the expert system shell so it can be modified during the solution process.

5. CONCLUSIONS

Knowledge-based techniques can be used to improve a variety of tasks in the finite element analysis process. Although traditional algorithm-oriented approaches are useful for rigidly structured problems, modern interactive computer software requires more flexible and efficient methods for handling data, procedures, and associated logic.

Simple rule-based systems can be used in innovative applications such as process control and solution optimization. KBES programs are well-suited for these problems because they explicitly separate facts and rules from the program control structure, allowing both the analysis results and the procedures used to achieve them to be scrutinized.

REFERENCES

1. Adeli, H., Ed. *Expert Systems in Construction and Structural Engineering*. New York, NY, USA: Chapman and Hall, 1988.
2. Allen, R.H. *Expert Systems in Structural Engineering: Works in Progress*. Computing in Civil Engineering **1**, 312-319, 1987.
3. Forde, B.W.R. *An Application of Selected Artificial Intelligence Techniques to Engineering Analysis*. Ph.D. Thesis. Department of Civil Engineering, University of British Columbia, Vancouver, B.C., Canada, 1989.
4. Bennett, J.; Creary, L.; Engelmore, R.; Melosh, R. *SACON: A Knowledge-Based Consultant for Structural Analysis*. Technical Report STAN-CS-78-699, California, USA: Stanford University Press, 1978.
5. Weiss, S.; Kulikowski, C.; Apté C.; Uschold, M.; Patchett, J.; Brigham, R.; Spitzer, B. *Building Expert Systems for Controlling Complex Programs*. Proceedings of the Second National Conference on Artificial Intelligence (AAAI-82), Pittsburgh, Pennsylvania, USA, 322-326, August 1982.
6. Forde, B.W.R.; Stiemer, S.F. *Knowledge-Based Control for Finite Element Analysis. Submitted to:* Engineering with Computers, 1989.