Earthquake Engineering, Tenth World Conference © 1994 Balkema, Rotterdam. ISBN 90 5410 060 5

Knowledge engineering in earthquake engineering

Hojjat Adeli
Department of Civil Engineering, The Ohio State University, Columbus, Ohio, USA

ABSTRACT: Several knowledge engineering approaches that have been employed recently or have potential applications in earthquake engineering are reviewed in this paper. They include symbolic processing, relational databases, object-oriented programming, parallel processing, genetic algorithms, and neural network computing.

1. SYMBOLIC PROCESSING

Lessons learned from destructive earthquakes are recorded in numerous printed documents, many of which are not readily available to practicing engineers and designers. In order to maximize the utility of the valuable knowledge learned from destructive earthquakes, we have been investigating symbolic processing approaches for presenting the performance evaluation and behavior of buildings stressed or damaged during earthquakes.

Using a production system model, Adeli and Shwe (1990) present EXQUAKE1, a prototype knowledge-based system for evaluation and selection of building configurations in seismic regions. The knowledge base of EXQUAKE1 contains lessons learned from earthquakes and published in books and papers. EXQUAKE1 has been developed on an IBM personal computer using the expert system shell INSIGHT 2+ (Level Five Research, 1986). The Representation of textual knowledge is through IF-AND-OR-THEN-ELSE rules in a language called Production Rule Language (PRL). For displaying various building configurations, EXQUAKE has a graphics interface developed in Turbo Pascal. EXQUAKE1 can point out the problems in building plans and elevations and recommend suitable shear wall arrangements for rectangular, triangular, and circular building plans.

Design codes represent the experience of designers and researchers accumulated over a long period of time. They include a wealth of information expressed mostly in textual and linguistic form. Shwe and Adeli (1991) present a prototype knowledge-based system for presenting the earthquake resistance building design requirements of the Uniform Building Code (UBC, 1991). A combined network-hierarchical architecture is proposed for code processing. Such an architecture make it possible to access various portions of the code quickly and conveniently. The knowledge base of EXQUAKE3 consists of a network of eight main knowledge bases. It has been developed on an IBM personal computer using the expert system shell INSIGHT 2+. EXQUAKE3 is intended to be a user-friendly electronic version of the code.

2. RELATIONAL DATABASES

Adeli and Hung (1990a) present a novel approach for processing the lessons learned from earthquakes and the knowledge of earthquake-resistant design by integrating the production system model, relational databases, and interactive computer graphics. The feasibility of this approach has been demonstrated by developing a prototype knowledge-based system, called WQUAKE, for processing the earthquake engineering knowledge learned in recent California earthquakes. WQUAKE has been developed on an IBM personal computer using the expert system development environment GURU (MDBS, 1987) and Turbo C language. Two methods of knowledge representation are used in WQUAKE for representing the knowledge of structural damage and behavior. They are production rules and relational databases. GURU does not provide a frame knowledge representation. However, the relational database capability of GURU is used to represent the knowledge in a structure similar to frames.

The relational databases are set up to store explicit relationships between records as well as implicit relationship among them created during the information processing. The problem domain is decomposed into independent sub-domains created separately. In order to create, modify, and query the knowledge base, relational databases are used to store information in defined formats. For example, for the problem of structural damage during a strong ground motion, the damage information must be related to the earthquake event and the type of the structure. Thus, three independent relational databases, called *EARTHQUAKE*, *STRUCTURE*, and *DAMAGE* are created to store the relevant information individually. In addition, secondary relational databases, related to the main databases, are created in order to describe the details of the problem domain.

For effective interaction with the user, WQUAKE provides four types of user interface: menu-driven interface, query interface, graphics interface, and natural language processor. Developed in Turbo C, the graphics interface can display the elements and damage mechanisms of structures. The integration of production rules, relational databases, and interactive computer graphics provides an effective means of processing the knowledge of earthquake-resistant design which is documented in numerous reconnaissance reports, papers, and books, mostly in the form of descriptive texts and figures.

3. OBJECT-ORIENTED PROGRAMMING

Object-oriented programming characteristics of abstraction, inheritance, and encapsulation of data and operations support three desirable features in the development of complicated engineering software systems: modularity, reusability, and ease of maintenance.

Adeli and Hung (1990b) present an object-oriented model for processing the knowledge of earthquake damage electronically. The model has been implemented in C++ in a prototype system, called OQUAKE, on a SUN workstation. Knowledge representation in OQUAKE is through a combination of frames and scripts. Two general purpose object classes provide support for developing and managing frame and script knowledge bases. Frames and scripts integrated into an object-oriented model provide an effective way of representing stereotypical sequences of knowledge in a text-base knowledge domain such as building damage and structural behavior during earthquakes.

Potential applications of object-oriented programming for analysis and design of structures subjected to seismic loading have yet to be fully explored. We briefly describe recent developments in related areas with potential applications in earthquake engineering. Yu and Adeli (1991) present computer-aided design (CAD) of structures using the object-oriented programming paradigm and a blackboard architecture for management of input/output data as well as intermediate data created during a consultation with the CAD system. A large amount of data is created in computer-aided engineering analysis and design. They use a blackboard architecture with a series of slots for storing data as the computation continues. As the number of data is increased, an effective way of managing these data is needed. Techniques from database management systems (DBMS) are useful for this purpose (Adeli and Yu, 1993). Not only the input and output data but also the intermediate data created during the execution must be managed. Therefore, in addition to ordinary slots, special slots, called controlling slots, are added to the blackboard in order to supply information on whether the data need modification, have been modified, or have yet to be created.

During the execution of the program, first the controlling slots on the blackboard are checked. If all input data for an object to execute are not available, it means that the object should wait until other objects are executed. In this case, the controller does not create this object until the next iteration. If a controlling slot indicates that the required input data for an object are available, and the output data from this object have not been created, then the controller creates this object which performs its task and creates the required data. After the computation is completed, the data are stored and the controlling slot is marked with a flag to indicate the newly modified data. Then, the object is deallocated from the computer memory. For an implementation of this model in C++ refer to Yu and Adeli (1991).

Yu and Adeli (1992) present an object-oriented finite element analysis (OOFEA) approach for solution of complex engineering problems. Traditional database models, called record-oriented data model, such as relational, hierarchical, and network models were developed primarily with business applications in mind. These models are not entirely satisfactory for engineering applications, because engineering problems often require complex data relationships. For efficient processing of a multitude of data types generated in finite element analysis of complex systems they have developed an object-oriented database management system using an enhanced entity relationship model with superclass/subclass and generalization/specialization features.

A class library is created for performing object-oriented finite element analysis in C++. These classes model the basic concepts and tools needed for finite element analysis of engineering problems. OOP languages do not provide facilities for manipulation of vectors and matrices often required in analysis of engineering systems. Classes VECTOR and MATRIX have been created for performing vector and matrix manipulations. These classes also provide methods for converting one-dimensional and two-dimensional arrays defined in FORTRAN to C++ and vice versa. With these methods, subroutines already implemented in FORTRAN can be linked to an OOFEA application program without a need to recode them in C++.

4. PARALLEL PROCESSING

The decade of 1990s will see a proliferation of multiprocessor computers. At this time most of the multiprocessing capabilities is limited to high-performance parallel machines and supercomputers (Adeli, 1992a&b). We should see increasing availability of multiprocessing capabilities on desktop machines. In order to solve problems on these machines we need to develop parallel or concurrent algorithms (Adeli et al., 1993). For an introductory review of parallel processing and parallel machines, see Adeli and Vishnubhotla (1992) and parallel programming languages and techniques, see Vishnubhotla and Adeli (1992).

Adeli and Kamal (1990) present an efficient 3-stage algorithm for automatic partitioning of frame structures for concurrent processing. Emphasis is directed towards attaining a balance of workload among processors, minimizing the bandwidth within each subdomain, and minimizing the number of linear equations to be solved for the interface degrees of freedom. Using this substructuring algorithm, Adeli and Kamal (1992a&b) present parallel algorithms for optimization of structures subjected to combined static and dynamic (such as seismic) loadings on shared-memory multiprocessor computers. Parallel stratagems are developed to assure the best concurrent performance and **speed-up within each step such as** assemblying the stiffness and mass matrices, solving the eigenproblem, finding the temporal solution, and computing the dynamic response. The algorithms have been implented in C and applied to structures of various size. It is shown that the efficiency of the parallel processing algorithms increases with the size of the structure, thus making them particularly suitable for optimization of large structures.

The supercomputer Cray YMP 8/864 has eight processors. Therefore, one can combine vector processing with parallel processing to achieve maximum processing speed. The simultaneous use of more than one processor is referred to as multitasking. Concurrent processing or multitasking on Cray YMP 8/864 is performed by macrotasking, microtasking, and autotasking (Saleh and Adeli, 1992). Macrotasking is performed at function level. Normally, major tasks that can be processed concurrently are macrotasked. Macrotasking is implemented by function calls and is suitable for tasks requiring large processing time because its overhead is large compared with that of microtasking. Macrotasked tasks should be identified when the general concurrent algorithm is developed. Microtasking is parallel processing at the loop level. Autotasking is the automatic distribution of tasks to multiple processors by compiler. It attempts to detect parallelism in the code automatically. It basically combines vectorization and microtasking automatically. Generally, autotasking works best on programs where most of the code consists of nested loops.

Hsu and Adeli (1991) present a microtasking algorithm for optimization (minimum weight design) of structures using the optimality criteria approach on the CRAY YMP 8/864. When using microtasking techniques, the programmer should first study the features of the problem and investigate the potential parallelism of microtasking. The most important step in developing a microtasked program is to understand the characteristics of the data. Data of a microtasked program can be divided into two groups: global data and local data. Global data are those which can be accessed and changed by any of the tasks during the execution. The local data of a particular task are known to and used by that task only.

Saleh and Adeli (1993) investigate judicious combination of various multitasking approaches in order to develop a vectorized and multitasked algorithm with maximum speed-up performance. They present strategies for various structural engineering problems.

5. GENETIC ALGORITHMS

Inspired by characteristics of biological evolution, genetic algorithms provide new approaches to optimization, search, and machine learning (Davis, 1991). For solution of optimization problems, genetic algorithms have been investigated recently and shown to be effective at exploring a large and complex space for finding values close to the global optimum guided by the equivalent biological evolution mechanisms of reproduction, crossover, and mutation (Goldberg, 1989).

Most optimization algorithms seek a solution in the neighborhood of the starting point, similar to local hill climbing. If there are more than one local optima in the problem the result will depend on the choice of the starting point and a global optimum cannot be guaranteed. Further, when the objective function and constraints have multiple or sharp peaks, the gradient search in the gradient-based algorithms becomes difficult and unstable. Adeli and Cheng (1993a) present an algorithm for optimization of structures by integrating genetic algorithm with the penalty function method. Genetic algorithms can be used directly only for solving unconstrained optimization problems. Engineering optimization problems are usually constrained problems. Thus, the constrained optimization problem is transformed to an unconstrained problem via a quadratic penalty function. To improve the robustness of the integrated genetic algorithm, they employ a variant penalty function coefficient strategy. That is, they increase the value of this coefficient after every so many iterations. What value to use for the coefficient and at which iteration to boost the value requires experience with the optimization problem domain.

Numerical experimentation can provide insight for judicious selection of the variables. The integrated genetic algorithm has been applied to optimization of space truss structures.

We find three characteristics of genetic algorithms attractive for optimization of large structures (Adeli and Cheng, 1993a). First, they yield a near global optimum solution with relatively fast convergence. Second, Only simple function evaluation is involved in genetic algorithms resulting in computationally efficient algorithms. Third, it can be easily implemented in a parallel processing environment because each string (design) in the population in independent of others.

An extension of the work of Adeli and Cheng (1993a&b) is development of genetic algorithms for optimization of structures under seismic loadings.

Genetic algorithms also provide an approach to machine learning. A hybrid learning algorithm using genetic algorithm will be described in the next section.

6. NEURAL NETWORKS

Neural networks computing can be used in earthquake engineering specially for development of smart or intelligent structures, systems with machine learning requirement. A few relevant and potentially applicable papers are described in this paper. Hung and Adeli (1991a) present a model of machine learning in engineering design, called PERHID, based on the concept of perceptron learning algorithm (Rosenblatt, 1962, Adeli and Yeh, 1989) with a two-layer neural network. PERHID has been constructed by combining perceptron with a single layer AND neural net. The problem of structural design is cast in a form that can be described by a two-layer neural network.

Hung and Adeli [1991b] present a hybrid learning algorithm by integrating genetic algorithm with error backpropagation multilayer neural networks. The algorithm consists of two learning stages. The first learning stage is to accelerate the learning process by using a genetic algorithm with the feedforward step of the backpropagation algorithm. In this stage, the weights of the neural network are encoded on chromosomes as decision variables. The objective function for the genetic algorithm is defined as the average squared system error. Therefore, it becomes an unconstrained optimization problem: find a set of decision variables minimizing the objective function. After performing several iterations and meeting the stopping criterion, the first learning stage is terminated and the chromosome returning the minimum objective function is considered as the initial weights of the neural network in the second stage. Next, the backpropagation algorithm performs the second learning process until the terminal condition is satisfied.

The convergence speed of neural network learning models is slow. Several hours or even days of computer time may be required using the conventional serial workstations. The number of iterations for learning an example in neural networks is often in the order of thousands (Adeli and Yeh, 1990, Hung and Adeli, 1991a). Thus, how to improve the learning performance of neural networks is currently an important research issue. One approach is the development of special purpose hardware. An alternative line of research is the development of novel software engineering techniques. One approach is development of learning algorithms on general-purpose parallel computers (Adeli, 1992a&b) with the objective of reducing the overall computing time. A second approach is development of hybrid algorithms by integrating neural network learning algorithms with other learning techniques.

Hung and Adeli (1991b) present a concurrent algorithm for the hybrid neural network/genetic algorithm described earlier for **distributed multicomputer systems.** The hybrid concurrent algorithm has been implemented in C on a transputer multicomputer system using Trollius programming environment (Burns, 1989, Trollius, 1991).

7. FINAL COMMENTS

Application of knowledge engineering approaches and techniques in earthquake engineering is a recent development. Out of 1492 papers presented at the 10th World Conference on Earthquake Engineering and published in the 10-volume proceedings of the conference very few papers utilized or dealt with knowledge engineering approaches. In this paper, an attempt was made to introduce the potential applications of a number of emerging knowledge engineering and innovative computing technologies. These technologies should find increasing applications in earthquake engineering during the coming years.

ACKNOWLEDGEMENT

The prototype systems EXQUAKE1, EXQUAKE3, WQUAKE, and OQUAKE described briefly in this paper have been developed during the course of a research project supported by the National Science Foundation under Grant No. CES-9904129.

REFERENCES

Adeli, H. Ed. (1988), *Expert Systems in Construction and Structural Engineering*, Chapman and Hall, London.

Adeli, H., Ed. (1992a), *Parallel Processing in Computational Mechanics*, Marcel Dekker, New York.

Adeli, H., Ed. (1992b), *Supercomputing in Engineering Analysis*, Marcel Dekker, New York.

Adeli, H. and Balasubramanyam, K.V. (1988), *Expert Systems for Structural Design - A New Generation*, Prentice-Hall, Englewood Cliffs, New Jersey.

Adeli, H. and Cheng, N.T. (1993a), "Integrated Genetic Algorithm for Optimization of Space Structures", ASCE Journal of Aerospace Engineering, Vol. 6 (in press).

Adeli, H. and Cheng, N.T. (1993b), "An Augmented Lagrangian Genetic Algorithm for Structural Optimization", ASCE Journal of Aerospace Engineering, Vol. 6 (in press).

Adeli, H. and Hung, S.L. (1990a), "A Production System and Relational Database Model for Processing Knowledge of Earthquake-Resistant Design", Engineering Applications of Artificial Intelligence, Vol. 3, No. 4, pp. 313-323.

Adeli, H. and Hung, S.L. (1990b), "An Object-Oriented Model for Processing Earthquake Engineering Knowledge", Microcomputers in Civil Engineering, Vol. 5, No. 2, pp. 95-109.

Adeli, H. and Kamal, O. (1990), "Automatic Partitioning of Frame Structures for Concurrent Processing", Microcomputers in Civil Engineering, Vol. 5, No. 4, pp. 269-283.

Adeli, H. and Kamal, O. (1992a), "Concurrent Optimization of Structures under Dynamic Loading, Part 1: Algorithms", International Journal of Mini and Microcomputers, Vol. 14, No. 3.

Adeli, H. and Kamal, O. (1992b), "Concurrent Optimization of Structures under Dynamic Loading, Part 2: Applications", International Journal of Mini and Microcomputers, Vol. 14, No. 3.

Adeli, H. and Kamat, M., Kulkarni, , and Vanluchene, D. (1993, "Review of High Performance Computing Methods in Structural Mechanics", Journal of Aerospace Engineering, ASCE, Vol. 5 (to be published).

Adeli, H. and Shwe, T.T. (1990), "A Knowledge-Based System for Selection Configurations in Seismic Regions", Heuristics, Vol. 3, No. 2, pp. 38-51.

Adeli, H. and Vishnubhotla, V. (1992), "Parallel Processing and Parallel Machines", in Adeli, H., Ed., *Parallel Processing in Computational Mechanics*, Marcel Dekker, New York, pp. 1-20.

Adeli, H. and Yeh, C. (1989), "Perceptron Learning in Engineering Design", Microcomputers in Civil Engineering, Vol. 4, No. 4, pp. 247-256.

Adeli, H. and Yu, G. (1993), "An Object-Oriented Data Management Model for Numerical Analysis in Computer-Aided Engineering", Microcomputers in Civil Engineering, Vol. 8, No. 3.

Burns, G.D. (1989), "A Local Area Multicomputer", Proceedings of the

Fourth Conference on Hypercubes, Concurrent Computers, and Applications.

Davis, L., Ed. (1991), *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York.

EXSYS (1988), *EXSYS Professional*, EXSYS Inc., Albuquerque, New Mexico.

Goldberg, D.E. (1989), *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, Reading, Massachusetts.

Hsu, H.L. and Adeli, H. (1991), "A Microtasking Algorithm for Optimization of Structures", International Journal of Supercomputer Applications, Vol. 5, No. 2, pp. 81-90.

Hung, S.L. and Adeli, H. (1991a), "A Model of Perceptron Learning with a Hidden Layer for Engineering Design", Neurocomputing, Vol. 3, pp. 3-14.

Hung, H. and Adeli, H. (1991b), "A Hybrid Learning Algorithm for Distributed Memory Multicomputers", Heuristics, Vol. 4, No. 4, 1991.

Level Five Research (1986), *INSIGHT 2+ Reference Manual*, Version 1.0, Level Five Research Inc., Indialantic, Florida.

MDBS (1987), *GURU - Expert System Management*, Vol. 1.1, Micro Data Base System Inc., Lafayette, Indiana

Rosenblatt, F. (1962), *Principles of Neurodynamics*, Spartan Books, New York.

Saleh, A. and Adeli, H. (1992), "Multitasking Algorithms for Optimization of Space Structures", Proceedings of Supercomputing'92, Minneapolis, Minnesota, November 16-20..

Shwe, T.T. and Adeli, H. (1991), "An Expert System for Design Code Processing", Structural Engineering Review, Vol. 3, No. 1, pp. 41-48.

Trollius (1991), *Trollius Reference Manual for C Programmers*, Research Computing, The Ohio State University, Columbus, Ohio.

UBC (1991), *Uniform Building Code*, International Conference of Building Officials, Whittier, California.

Vishnubhotla, P.R and Adeli, H., "Parallel Programming Languages and Techniques", in .Adeli, H., Ed., *Parallel Processing in Computational Mechanics*, Marcel Dekker, New York, pp. 21-32.

Yu, G. and Adeli, H. (1991), "Computer-Aided Design using Object-Oriented Programming Paradigm and Blackboard Architecture", Microcomputers in Civil Engineering, Vol. 6, No. 3, pp. 177-190.

Yu, G. and Adeli, H. (1992), "An Object-Oriented Finite Element Model for Composite Laminate Analysis", Proceedings of the American Society for Composites 7th Technical Conference on Composite Materials, University Park, Pennsylvania, October 13-15, 1992