Zeitschrift: IABSE reports = Rapports AIPC = IVBH Berichte

Band: 58 (1989)

Rubrik: Session 3: Expert systems for design and construction

SESSION 3

Expert Systems for Design and Construction

Application de systèmes experts dans le projet et la construction

Expertensysteme in Entwurf und Erstellung

Leere Seite
Blank page
Page vide

Coupled Expert Systems for Engineering Design

Systèmes experts: «couplés» en conception

Gekoppelte Expertensysteme für den Ingenieurentwurf

H. ADELI
The Ohio State Univ.
Columbus, OH, USA

Hojjat Adeli received his Ph. D. from Stanford University in 1976. He is currently a Professor of Civil Engineering at the Ohio State University. He is the author or editor of about 190 publications in various fields of computer-aided engineering.

SUMMARY
This paper reviews the development of coupled expert systems for engineering design. AI-based symbolic processing and traditional numerical processing are combined in a coupled expert system. Various approaches to coupling are discussed. Several coupled knowledge-based expert systems developed by the author and his associates are described briefly.

RESUME
Cet article passe en revue le développement de systèmes experts dits «couplés» dans le domaine de la conception. Programmation symbolique et programmation numérique traditionnelle sont combinées dans les systèmes experts «couplés». Différentes approches de «couplage» sont débattues. Plusieurs systèmes experts à base de connaissance «couplés» développés par l'auteur et ses associés sont décrits dans cet article.

ZUSAMMENFASSUNG
Dieser Beitrag bespricht die Entwicklung gekoppelter Expertensysteme für den Ingenieurentwurf. Dabei werden traditionelle numerische Methoden mit auf künstlicher Intelligenz basierenden symbolverarbeitenden Methoden kombiniert. Verschiedene Koppelungsarten werden besprochen und anhand von bereits entwickelten Expertensystemen erläutert.

INTRODUCTION

A large number of knowledge-based expert systems has been reported in the literature during the last few years. With a few exceptions, these systems are primarily concerned with symbolic processing of experiential knowledge often expressed in the form of production rules. The problem of integrated engineering design consists of the conceptual design, preliminary design, structural analysis, and the final detailed design. Thus, an integrated engineering design requires extensive numerical processing in addition to symbolic processing of heuristics and experiential knowledge. This paper reviews briefly several coupled knowledge-based expert systems developed by the author and his associates during the past few years. Readers interested in the details of these systems should refer to the references given at the end of the paper.

RTEXPERT

RTEXPERT (for Roof Truss EXPERT) is a coupled expert system for detailed design of three different types of roof trusses, namely, flat Pratt, pitched Pratt, and Fink trusses (Adeli and Al-Rijleh, 1987). RTEXPERT can advise the user on the appropriate type of the roof truss, selection of the layout of the truss, and the loading. It also presents the final detailed design of the selected truss. The basis of design is the American Institute of Steel Construction specification (AISC, 1980). The truss is designed for dead, live, snow, and wind loads in accordance with the American National Standard Institute specification (ANSI, 1982). RTEXPERT has a comprehensive graphic interface for displaying the truss configuration, cross-sections, loading, and deformed shape. Information about individual members is presented through multi-window graphics-text displays.

RTEXPERT has been developed on an IBM Personal Computer with two floppy disk drives and 512K of RAM. The knowledge base and explanation facility of RTEXPERT have been developed using INSIGHT 2+ expert system shell (Level Five Research, 1986). The mathematical computations, graphic algorithms, and data file manipulation routines have been developed in Turbo Pascal.

BTEXPERT

BTEXPERT (for Bridge Truss EXPERT) is a prototype expert system for optimum (minimum weight) design of bridge trusses (Adeli and Balasubramanyam, 1988a&b). The scope of BTEXPERT is limited to the optimum design of four types of bridge trusses, i.e., Pratt, Parker, parallel-chord K truss, and curved-chord K truss for a span range of 100-500 ft. Design constraints and the moving loads acting on the bridge are based on the American Association of State Highway and Transportation Officials (AASHTO) specifications (AASHTO, 1983).

Bridges are to be designed for combined dead and live (moving) loads. Live loads are usually specified by design specifications. AASHTO live loads are used in BTEXPERT. These loads can be classified into three categories : Two-axle truck (H 15 and H 20), two-axle truck plus one-axle semitrailer (HS 15 and HS 20), and uniform lane loadings consisting of a distributed load of uniform intensity but variable length and a single moving concentrated load. A heuristic approach has been developed for finding the maximum compressive and tensile forces in the members of a bridge truss based on the classification of the shape of the influence line diagrams (ILD's) and the type of AASHTO live loads.

The optimum design of a bridge truss consists of selecting the right combination of the cross-sectional areas of the truss members so as to satisfy all the design constraints and produce a least weight truss. For achieving this, a hybrid optimization algorithm has been developed for minimum weight design of bridge trusses subjected to moving loads (Adeli and Balasubramanyam, 1988b). In this algorithm, an efficient zero order explicit approximation is combined with a more accurate but less efficient explicit stress constraint formulation.

BTEXPERT has been developed using the Expert System Development Environment (ESDE) (IBM, 1986b & c) and the Expert System Consultation Environment (ESCE) (IBM, 1986a & c) implemented in Pascal/VS. The first program is used to develop expert systems and in particular the knowledge bases. The second program provides facilities for executing them. The two programs are collectively referred to as the Expert System Environment (ESE). The analysis and optimization algorithms have been coded in FORTRAN 77.

For performing numerical processing and for graphics interface, BTEXPERT uses procedures implemented in FORTRAN 77. Therefore, an interface has been developed in PASCAL/VS interfacing the knowledge base of BTEXPERT implemented in ESE to the interactive bridge truss optimization program implemented in FORTRAN 77. The interface consists of a number of procedures written in PASCAL/VS and use ESE utility functions. They transfer information from ESE to numerical and graphical processors and acquire information from the numerical processors and transfer it to ESE. This information may be in the form of values of control parameters, and/or the knowledge about the sequence of application of the numerical algorithm, and/or the results obtained from the numeric processors (Adeli and Balasubramanyam, 1988b).

SDL and STEELEX

SDL (for Structural Design Language) is a domain-specific expert system development environment implemented in INTERLISP for building coupled knowledge-based expert systems for integrated design of structures (Paek and Adeli, 1988a and 1988b). The complex body of knowledge needed for detailed design of a structure is fractionated into smaller and manageable knowledge sources which are organized into a hierarchy of cooperating conceptual specialists. SDL has been used to develop an expert system for integrated design of steel building structures consisting of moment-resisting frames, called STEELEX (Paek and Adeli, 1988c). STEELEX designs the beams and columns making the frame as well as the moment-resisting connections. STEELEX has a multi-window graphics interface that can display orthographic and isometric views of the structure and moment-resisting connections.

EXOPT

EXOPT is a prototype coupled knowledge-based system for large scale structural design optimization (Adeli and Balasubramanyam, 1988b). The domain of EXOPT is limited to optimization of plane trusses under arbitrary multiple loading conditions subjected to user-specified stress, displacement, and fabricational constraints. In order to make use of the trend information obtained during the optimization cycles, design variable classification and heuristic constraint deletion strategies have been implemented in EXOPT. Symbolic processing is done through the use of the IBM ESE and numerical processing is performed in FORTRAN 77.

PG-BRIDGE1

PG-BRIDGE1 is a prototype knowledge-based expert system for optimum design of stiffened and unstiffened, homogeneous and hybrid steel plate girders used in highway bridges (Adeli and Mak, 1988&1989). The basis of design is the American Association of State Highway and Transportation Officials (AASHTO) specifications (AASHTO, 1983). The plate girders are subjected to the live (moving) loads of the AASHTO specifications.

A robust mathematical optimization algorithm has been developed for minimum weight design of multispan steel plate girders used in highway bridges employing the generalized geometric programming technique (Adeli and Chompooming, 1988). The total weight of the plate girder is used as the objective or minimization function. The design constraints are based on the AASHTO specifications. The design variables are the flange width and thickness, the web depth and thickness, and the width and thickness of transverse stiffeners, and the spacing of the intermediate stiffeners. In the optimization algorithm, the nonlinear primal problem is transformed to an equivalent standard linear programming problem via double condensation.

PG-BRIDGE1 is developed using the IBM ESE described previously. Numerical processing is performed in external programs written in FORTRAN 77. In order to link the graphics interface and load the FORTRAN utility functions, an execution file ESXGLBL is invoked automatically each time the ESE is loaded.

FRAMEX

FRAMEX is a coupled knowledge-based expert system for integrated design of rectangular multistory steel buildings in which AI-based symbolic processing is integrated with numerical processing and data base management techniques (Adeli and Chen, 1989). The vertical lateral force-resisting systems in the building are limited to moment-resisting frames located on the perimeter of the structure.

FRAMEX starts with the preliminary design and presents the final detailed design. The basis of steel design is the recently-developed American Institute of Steel Construction (AISC) Load and Resistance Factor Design (LRFD) specifications (AISC, 1986). The loading on the structure can be various combinations of dead load, live load, snow load, wind load, and earthquake load. The computations of live, wind, and earthquake loads are according to the last edition of Uniform Building Code (UBC, 1988). The snow load is calculated on the basis of the American National Standards Institute (ANSI) minimum design loads specifications (ANSI, 1982).

FRAMEX has been developed on an IBM Personal Computer (PC) with 640 KB of Random Access Memory (RAM), a 360 KB floppy disk drive, and a 20 MB hard disk drive.

Symbolic processing in FRAMEX is done using the Personal Consultant (PC) Plus expert system shell (TI, 1987a&b). Developed by Texas Instrument in 1987, PC Plus is a microcomputer-based expert system environment implemented in LISP. PC Plus provides an English-like language, called Abbreviated Rule Language (ARL), an explanation facility, and the ability to interact with external programs developed in procedural languages and dBASE III data files and Lotus 1-2-3 spreadsheets. Numerical processing for structural analysis, member selection in preliminary design and redesign, optional LRFD procedural code checking, and graphics interface is done in Turbo Pascal. Figure 1 shows the architecture of FRAMEX schematically.

Data base files used in FRAMEX are classified into static and dynamic data files. The static data files contain the data which are not changed during a consultation with the system. There are four such static data files. One of them is a dBASE III data base file with extension .DBF. The remaininng

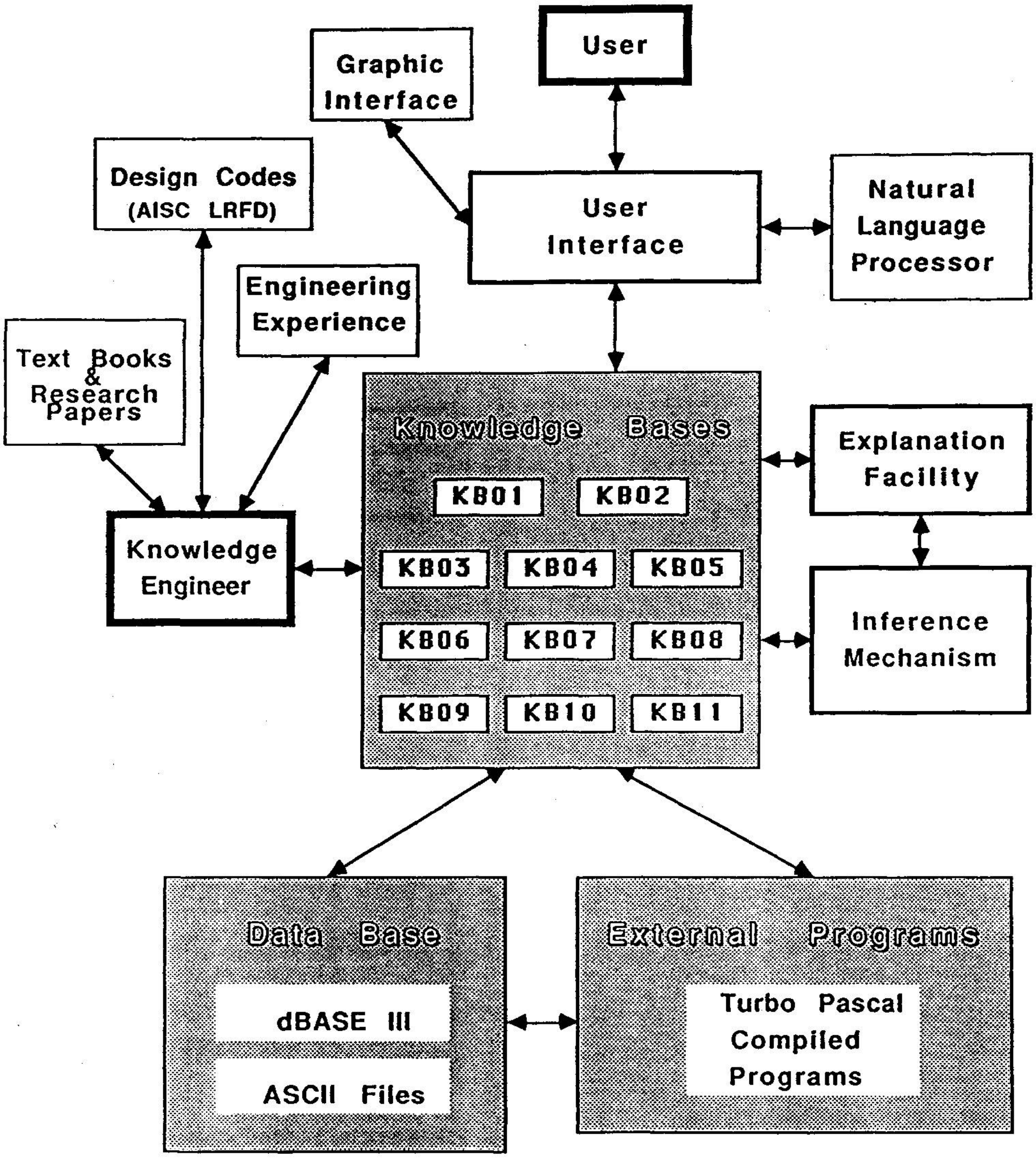


Figure 1 Architecture of FRAMEX

three data files are ASCII database files with extension .DAT. Figure 2 shows how the four static data files are used by the knowledge bases.

During the complicated design process various types and quantities of data are generated. The large quantity of data must be managed properly and efficiently. FRAMEX manages the data created during the consultation process through 30 dynamic data files. The contents of these data files are changed during a consultation with FRAMEX. Details of the dynamic data files generated by knowledge bases and Turbo Pascal programs are given in Adeli and Chen (1989). Figure 2 also shows how the dynamic data files are generated and used by various knowledge bases and external programs.

SDLS

We are currently developing a prototype Structural Design Learning System (SDLS) in a combination of Prolog and Pascal languages (Adeli and Yeh, 1990). The machine learning approach used is Explanation-Based Learning (EBL). A relatively new approach in machine learning, EBL appears to produce a more reliable generalization without the inductive bias observed in the similarity-based learning developed earlier. SDLS includes a modified implementation of EBL in Prolog, where the two stages of explanation and generalization are combined into one stage. A more formal definition of the operationality criterion suitable for the domain of structural design is proposed and used in SDLS.

REFERENCES

AASHTO (1983), Standard Specifications for Highway Bridges, 13th ed., American Association of State Highway and Transportation Officials, Washington, D.C.

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

Adeli, H. and Al-Rijleh, M.M. (1987), "A Knowledge-Based Expert System for Design of Roof Trusses," Microcomputers in Civil Engineering, 2(3), pp. 179-195.

Adeli, H. and Balasubramanyam, K.V. (1988a), "A Knowledge-Based System for Design of Bridge Trusses," ASCE Journal of Computing in Civil Engineering, Vol. 2, No. 1, pp. 1-20.

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

Adeli, H. and Chen, Y.S. (1989), "Structuring Knowledge and Databases in Expert Systems for Integrated Structural Design," Microcomputers in Civil Engineering, Vol. 4, No. 3.

Adeli, H. and Chompooming, K.V. (1988), "Interactive Optimization of Multispan Plate Girders," Microcomputers in Civil Engineering, Vol. 3, No. 3, pp. 227-244.

Adeli, H. and Mak, K. (1988), "Architecture of a Coupled Expert System for Optimum Design of Plate Girder Bridges," Engineering Applications of Artificial Intelligence, 1(4), pp. 277-285.

Adeli, H. and Mak, K. (1989), "Application of a Coupled Expert System for

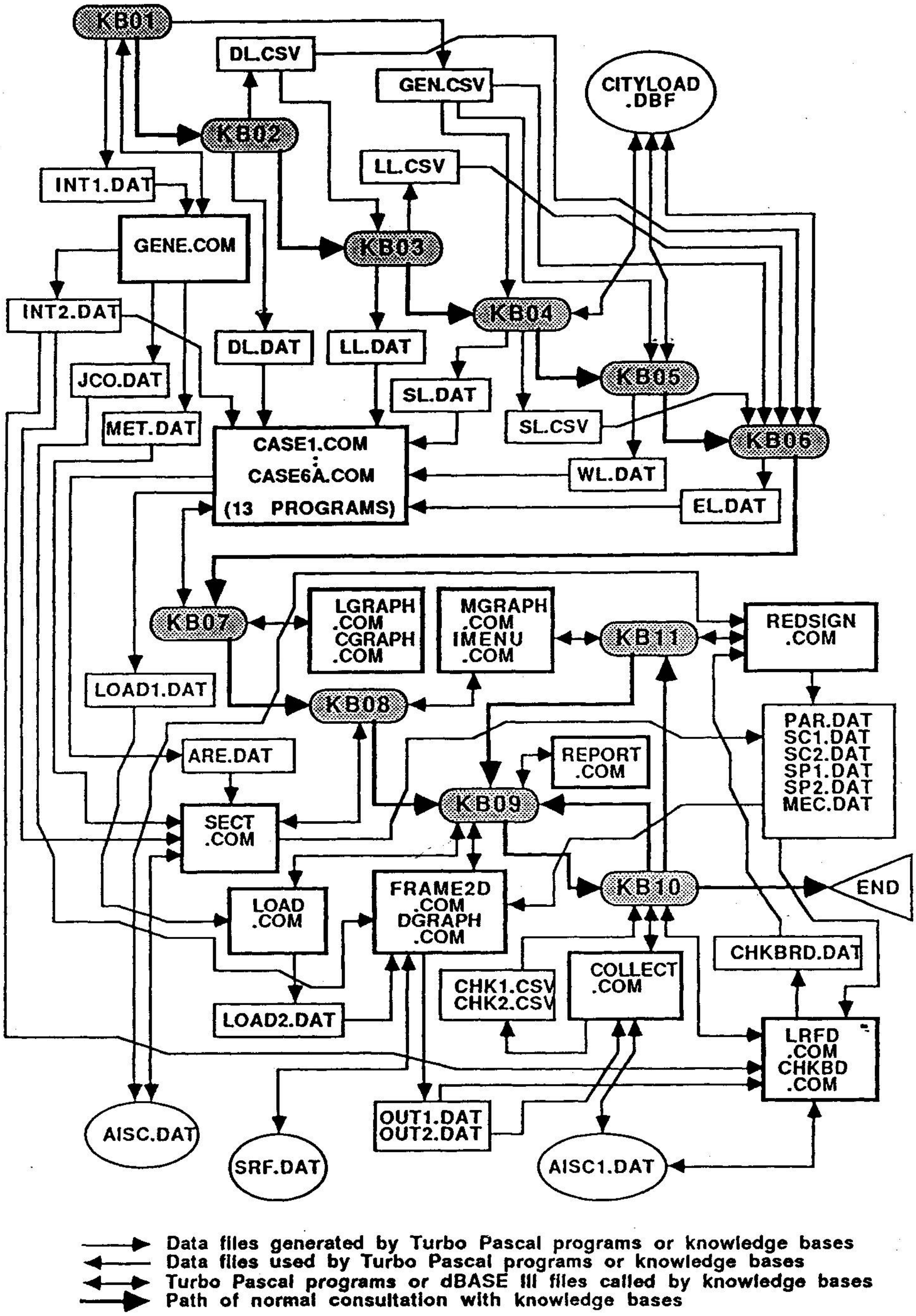


Figure 2 Path of consultation and the relationship of knowledge bases, data bases, and external Turbo Pascal programs

Optimum Design of Plate Girder Bridges," Engineering Applications of Artificial Intelligence, Vol. 2 (to be published).

Adeli, H. and Paek, Y.J. (1986), "Computer-Aided Design of Structures Using LISP," Computers and Structures, Vol. 22, No. 6, pp. 939.956.

Adeli, H. and Yeh, C. (1990), "Explanation-Based Machine Learning in Engineeeing Design," in preparation.

AISC (1980), Manual of Steel Construction, American Institute of Steel Construction, Chicago, 1980.

AISC (1986), Manual of Steel Construction - Load and Resistance Factor Design, American Institute of Steel Construction, Chicago.

ANSI (1982), American National Standard Minimum Design Loads for Buildings and Other Structures, American National Standards Institute, New York.

IBM (1987a), "Expert System Development Environment User´s Guide," SC38-7006-0, IBM Corporation, Bethesda, Maryland.

IBM (1987b), "Expert System Consultation Environment User´s Guide," SC38-7005-0, IBM Corporation, Bethesda, Maryland.

IBM (1987c), "Expert System Development and Consultation Reference Manual," SC38-7004-0, IBM Corporation, Bethesda, Maryland.

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

Paek, Y. and Adeli, H. (1988a), "Representation of Structural Design Knowledge in a Symbolic Language," Journal of Computing in Civil Engineering, 2(4), pp. 346-364.

Paek, Y. and Adeli, H. (1988b), "SDL: An Environment for Building Integrated Structural Design Expert SystemS," in Adeli, H., Ed., Microcomputer Knowledge-Based Expert Systems in Civil Engineering, American Society of Civil Engineers, New York.

Paek, Y. and Adeli, H. (1988c), "STEELEX: A Coupled Expert System for Integrated Design of Steel Structures", Engineering Applications of Artificial Intelligence, 1(3), pp. 170-180.

TI (1987a), Personal Consultant Plus - Getting Started, Texas Instrument Incorporated, Austin, Texas.

TI (1987b), Personal Consultant Plus - Reference Guide, Texas Instrument Incorporated, Austin, Texas.

UBC (1988), Uniform Building Code, Whittier, California.

Knowledge-Based System for Automatic Design of Glued Laminated Structures

Système expert pour le calcul automatique des structures en bois lamellé-collé

Wissensbasiertes System für den automatisierten Entwurf von Glulamstrukturen

Zvonimir ŽAGAR
Prof. Dr.
Zagreb University
Zagreb, Yugoslavia

Z. Žagar, born 1931 received his Civ. Eng. degree at the Faculty of Architecture, Civ. Eng. and Surveying of the Zagreb University in 1957. He worked from 1965 to 71 as senior lect. at the BT, Faculty of Arch., UST Kumasi, Ghana. He holds the Chair for timber structural design at FCI Zagreb.

SUMMARY
We have developed a knowledge system for cost optimisation design of two and three hinged glued laminated framed 3D structures. The system incorporates all the necessary standards and codes of practice. On the basis of input building's clear-area-height dimensions, the geographical location and current prices of material used and the past «learned» experience, it designs the optimally priced 3D structure with all the details included.

RESUME
Nous avons développé un système pour l'optimisation des coûts des structures en bois lamellé collé du type ferme à deux ou trois articulations avec la possibilité de minimisation du côut global du bâtiment. A partir des données de base, représentées par la portée, la longueur du bâtiment, la position géographique ainsi que le prix du bois et de l'acier, les résultats fournissent le coût optimal du système à trois dimensions. Les résultats sont conservés dans la base de données pour une utilisation ultérieure.

ZUSAMMENFASSUNG
Ein Expertensystem zur Kostenoptimierung von 2- und 3-gelenkigen räumlichen Brettschichttragwerken ist entwickelt worden. Die entsprechenden Normen und Vorschriften sind dabei berücksichtigt. Aufgrund der Innenmasse der Gebäude, der geographischen Lage, der aktuellen Baupreise sowie der früheren «Erfahrungen», entwirft das System eine wirtschaftliche Tragstruktur in allen Details.

1. Introduction

We gradually embarked upon the development of the knowledge system (KS), until involvment at the present stage. First, some team members (under late Prof. Dr. Sablic) had developed programs for automatic glulam beam table generation for the timber industry in Yugoslavia. Fig. 1. [23] [12]. In the meantime the SZS (the Standard Scecification Office) updated the JUS (Yugoslav standards for wood structures: JUS UC 9.200, and JUS UC 9. 300) valid from 1985.[15]. Also we were involved in writing those standards, we recognise the shortcommings of these standards: they have the "old form" and the oddly arbitrary naming of the variables, and some in accordance with internationally accepted standards. Some of us had the idea about writing the standards in modular form suitable for direct computational use: in FORTRAN, PASCAL, BASIC etc. This could be then distributed with the usual standard written form to various legal users [33], [34], [35].
But this idea, based on work of prof. Fenves and others [9], [10], [11], [13]. was rejected at that time.
I made an attemt to write some parts of the standards in modular forms suitable for direct computer use. But this was not done systematically. The idea of optimization was actually old and dated to the times when we did some considerably complex optimization of very large and complex structural use in the realm of architecural planning, based on generalised linear model for optimization of planning based on the works of Aguilar, deNeufville & Stafford and Stark & Nichols [2], [6], [25], [30]. We did it on the then Interdisc. Interfaculty Traffic and Transportation Study of the Zagreb University where I lectured on structural design (1971-1980). In those works we just assumed that the reif. concrete structure was already rationaly designed, and the model developed took into account multiused facilities and considered realistic design constrains such as zooning and parking regulations and requirements, space use, rentals of floor areas (depending on the use and architecural quality and marked preferences), construction costs, budget and the maximisation of profits [2]. There was a considerable disapointment, because nobody from the authorities responsible for planning and investments was interested in the use of the model. This was done in 1975 and well described [30] in 1977. The computers were rare and the knowledge to use it mostly limited just inside universities. Later at the Faculty of Civil Engineering (FCE) I made (with the help of some students) some optimization programs for planar wooden structure design based on the JUS standards. We developed some simple programs for ratioinal design of two and three hinged glulam arches, glulam (variously shaped) beams, some steel structural parts, foundations etc., some written in FORTRAN and some in BASIC [17], [31], [37], [38], [39].

2. The first steps

The appempt to write an ES or KS as a rather elementary or primitive version was a tempting foundation base for more elaborate systems. The idea to start to write an rudimentary ES (KBS) was born at the time of the Zagreb Universiade games (in 1987.), when we got the contract to design three (over 30 m spaned) glulam roofs over sports halls. So I made a program for spatial optimization of planar glulam roof structures and then started to develope it into a rudimentary KBS [39]. This immediately highlights the fundamental problems of ES: that they are usually quite knowledgeable about a limited (in that case a very narrow) domain, but have no knowledge of wider world. It should be noted that we always checked the obtained results with parallel computations on the mainframe computer using FEM and the ICES STRUDL 2 system [15]. We latter even checked some assumptions and theories of glulam beam behaviour (Mohler, Heimeshoff), and the theory of spring back effects-initial stresses due to manufacturing processes of streight and curved glulam members and the "size effect", etc.

In that then written system the various embodied theories, the standards and the codes of practice could be reveiled by the use of various HELP routines in graphical or printed/written forms on the CRT devise. The system for the design of spatially braced glulam planar structures could do the optimization quite independently form the structural engineer, the "man in the link" was accepted to give some explanations, details of codes and the theories used, just to give him the feeling of a master over the program [5], [7], [32], [39].

We have recognised the shortcommings of separate unlinked (or linked) optimisation programs as well as the 2D towards the 3D comprehensive approach in design, and the role of the partialisation of different architectural forms in the design of timber structures.

A small program which was able to find the 2D optimal solution among several predeterminated 2D forms had shown that an optimization should include comparission of architectural forms and as well as materials on the price basis. This program was able to compare the structures shown in fig. 2. [7], [37].The efforts of research work and some of the student diploma works were cricial for the development of such system.

It was enlightening to run this program for different spans, building heights, bearing capacities of soil and different prices of materials involved. In running it I learned much of the design decision making and was able toobserve the generation of the thumb-rules. To my suprise the, with the program gained solutions, were the same as the stated in books and known as "thumb rules", gained in practice by healty workmanship and past economical gains.

3. Development

We were seeking to develope a model for writing a real KBS which could be gradually expanded to more complex one. In the case of timber design we were aquanted with the Basic wood design for minicomputers [18]. We developed on the past experience a KBS for wood design on the 3D (spatial) design basis which we are trying to enlarge to other structural forms. One part of the system is operational and this is represented here .

4. The KBS for design of framed glulam structures

Such systems embody the accumulated technical and technologycal knowledge, the contemporary state of standards and codes of practice, the past "thumb-rule" experience, and the new knowledge obtained by the use of the system. In this way the knowledge is gradually increased, but because of the limited scope of the narrow active area (the one bay space system) there is a limit

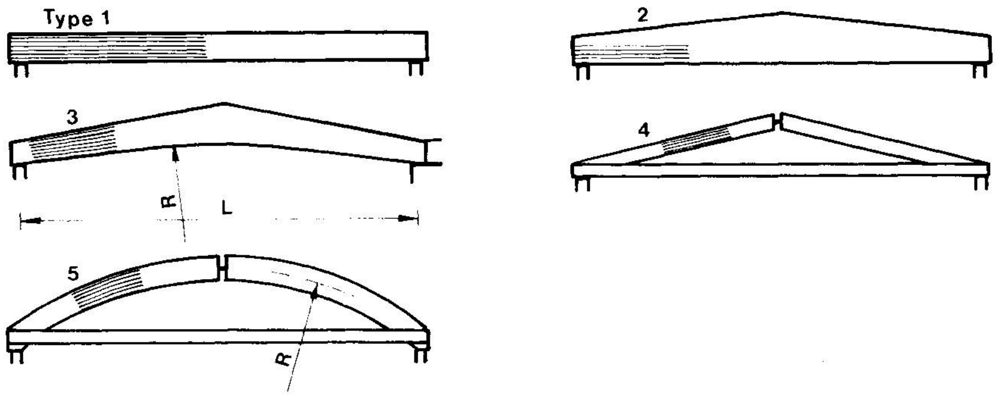


Fig. 1. The early approach to optimization of glulam beams.

for the growth of the knowledge base. This limit is obtained quickly. The acumulated knowledge saved in the data bank could be applied by practical design of glulam timber structures.
The well known ICES STRUDL system is very close to this definition but the searching of the DD2 data base file (where the past experience could be saved) is unpractical without an intelligent interface for revival of the, only to that specific problem related, data.
There have been some sporadic attempts to solve some optimisation problems in design (for example to solve the question of optimal design of simple structures: glulam two hinge arches, optimization of purlines, design of simple glulam beams, reinforced concrete one bay frames) [12], [17], [29], [31], [36], [37], [38], [39]. The described automated program for 3D optimization of one bay glulam framed building could learn and the learned experience knowledge is saved in a data base which could be expanded by interpolation or /and by further learning in future runs.
We have in this program established the "man in the link" to:
(i) help in re-education of users (students, str. engineers),
(ii) to force "experts" to critically review their knowledge and thought processes, which than could lead to new ideas for more effective solutions to problems,
(iii) to put a point to the role of contemporary structural engineers.
It was a policy from the beginning that the role of the program users was a supperficial role only, but this shouldn t be apparent to the user. This is far from deskilling the role of structural designer but we are on the way to do just this.
It could sound unbelievable that the first steps were done on a SVI 328 and then tranfered to and developed on IBM XT/AT PC, where the KBS is now operational. Now it is transfered to CONVEX. The steps described, are the total minimum cost optimisation of an architecturaly prediscribed one bay conventional glulam timber 3D workshop buildings based on two- or three- hinged glulam frame stuctures. To be able to start with this we have to rewrite in digital forms the relevant Design specifications, specification for loads and code of practice. Therefore we have proposed the writing of the "new generation" of design codes and specifications in a way of verified subroutines expressed in different languages: Pascal, Fortran, Forth and why not McBasic [8], [9], [10], [11], [11], [13], [33], [34], [35]. These subroutines should be the integral part of the new code representation of the

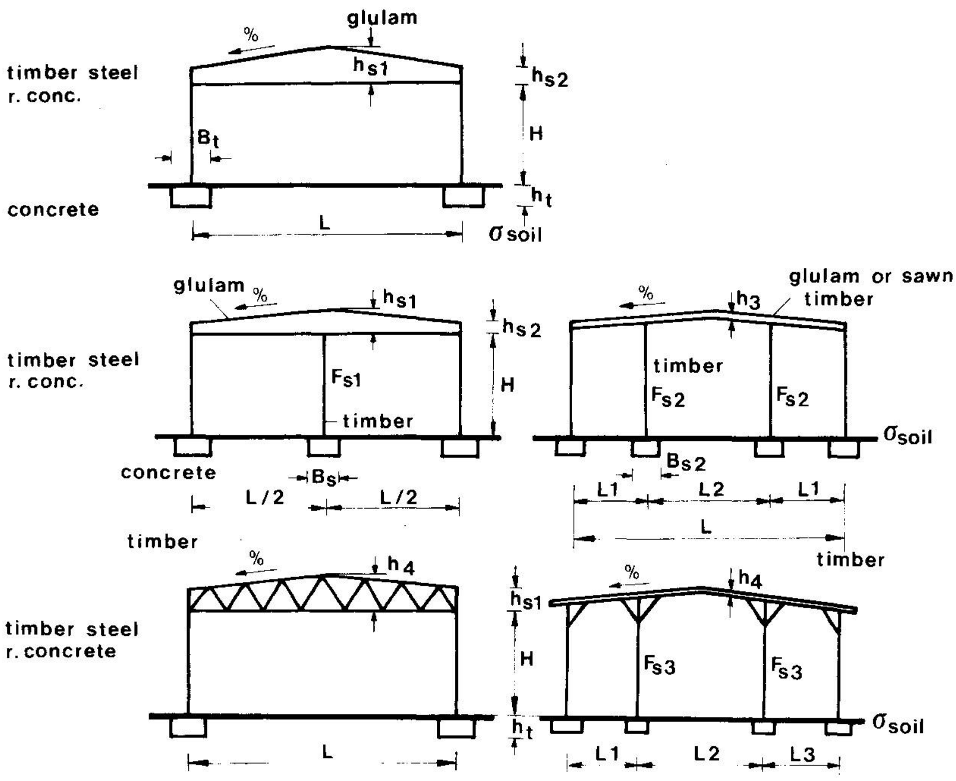


Fig.2. An early 2D optimization approach to different structural systems.

JUS codes and standards [15], closely related to ISO standards. We have imagined the whole building standards codes as a system of drawers in a cupboard. The old outdated rules scould be taken out and the new rules pushed in without affecting the whole. These standards should be distributed to the prospective legal users on floppies and suplemented by textual parts. They could be used in various programs as modules. The standards should be constantly upgraded and distributed to legal users. An institution should be in charge for testing, distributing and upgrading the rules.This work is closely related to the selection of variable's names, the choice of letters (notation) used etc., which could be obtained only by mutual agreement at the level of the Yu Standard specification Office. On this level there has not been much interest, nor financial support. So we were forced to do the task to write some of the parts of such a system in hope that somewhere in the future the SZS (Bureau for National Standards) will accept the whole idea. After that we have started to write the main managing routines for structural calculations, code managing, stability checking, price optimisation, and optimisation of mutually interlaced building parts etc.
On the basis of the architectural choice of the main outside dimensions of the building, the geografical position inside Yugoslavia (governing the snow and wind loads), the choise of exposion to wind loads (closed, open, partialy closed building etc.),

the type of roofing (and the roof's slope), the choice of timber (glulam, sawn timber), the class (1st or 2nd) and species of timber (soft or hard), the environmental exposure (pH level), the choice of supervisions (and the level of meintenance), the proposed duration of the structure, the current prices of materials and some other choices, the optimisation process starts and after a while the optimal solution will be obtained (fig. 3). The described data colection (input) is done in a converstion mode and there is no chance left to omit any question. There are several choises to input the data for wood classes and species: this could be done by the user or could be left to be decided by program according to price levels of materials used.
Starting from the purline distance the spacing among the frames, the optimisation of the steel bracing system and its form, the choice of the glulam framed system (the two- or three- hinged glulam frames), the system longs for the minimum priced feasible 3D building. When the global economy is obtained the details (the circular dowell connection between post and beam etc.) are optimised (fig. 3.), and searched for the most economical solution. Up to the last bolt, dowell and nail.
The optimal solution is saved in the data bank. This data bank is searched first, when a new run is started.
In this way the data base is growing and accumulating knowledge about such systems. After some number of runs, there is only a sporadic need for new runs of the whole system. It is clear that the various dimensions are dependent of the geometry, the geografical position of the building and other parameters, but mostly from the current prices of the material used. The to day obtained economical solution, if for some time not realised, might not be economically erected by another future price relation. This is by the way the current position with us. At present, the system is "learning".
There are some shortcommings which we are trying to correct now. One is the intelligent search routine and the range of economical decisionmaking in a stabilised and an inflatory environment. But the heuristic search should include such realities. At present we are trying to develop the heuriustic search and logical save routines and to embody in the program the typical truss structures supported on "I" steel [20], and/ or reif. concrete columns or surrounding walls (with or without openings). This is being done as a PHd and a MSc thesis and should be finished in a year or two.
For the intelligent heuristic search routine, we did among us some brainstormings to discover how an experienced structural engineer should use the past experience.
There are some problems in the (by us) hectically inflating economy: the prices are rising, but the relations of prices of different materials are changing too and are not constant. The second schortcomming is threefold: - the interface language communication, - the JUS standard embodied, and - the various variable's names (reflecting our language). For the language (for interfacing the program with the outside world) I feel that it should be written in english as a standard language. But there exists among us a strong feeling to use native croatian language. The variables should reflect the ISO (by my opinion outdated) recommandation and this could be changed easily. The JUS standards are outside the program (as a module) and could be changed and updated to level needed.
Parallel to this work we are developing an epoxi-glued with four HTB perpendiculary prestressed corner joint for framed structures instead of the expensive and time consuming circular bolted

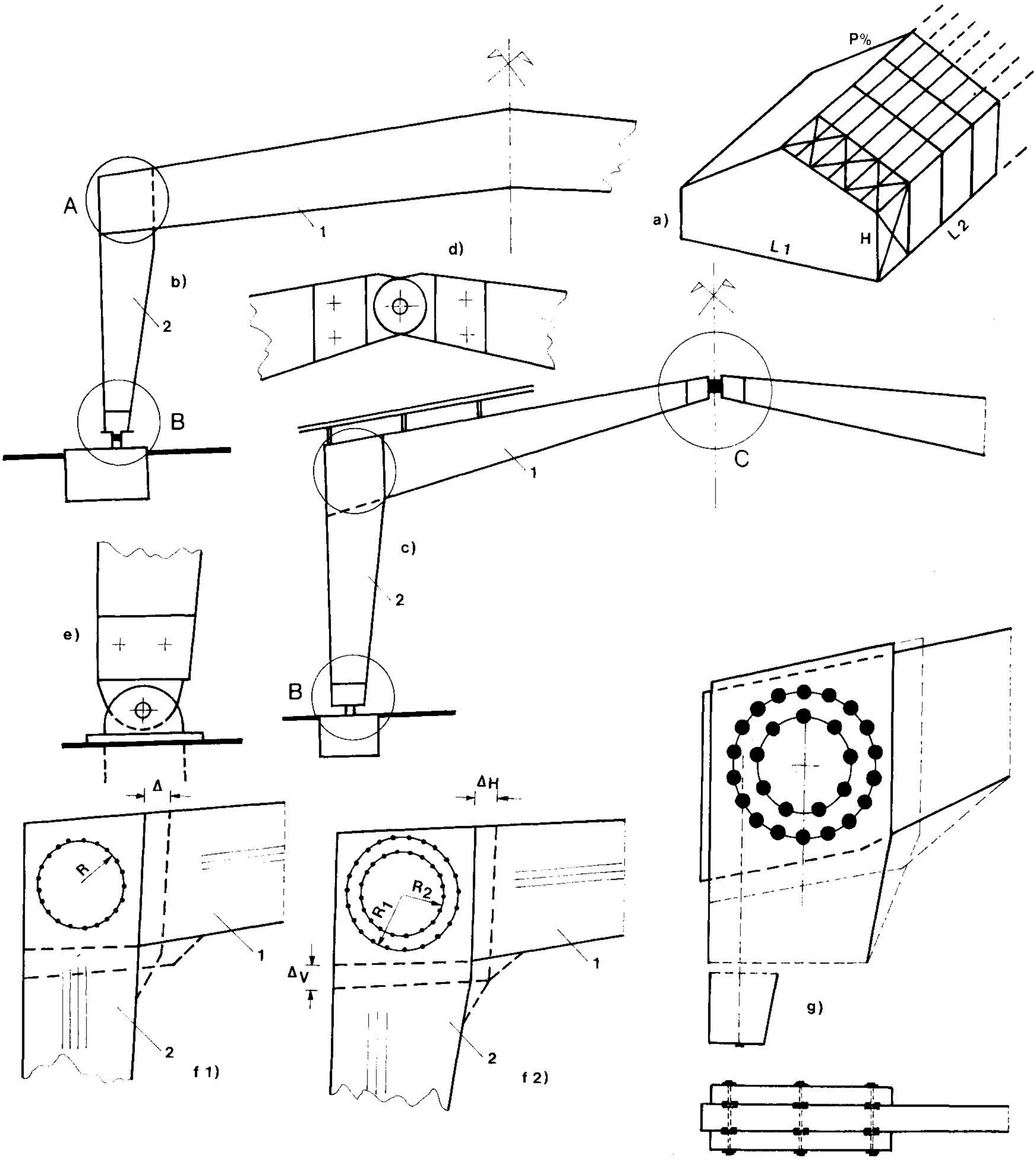


Fig.3. a) The outlines of the building, b), c) the two and three hinged glulam frame, d) the top joint of the three hinged frame, e) detail of the support hinge, f), g) details of the generated circular bolted corner joint (one or two rings).

connection already included in the KBS. This corner joint is being tested in lab and the results compared withe a large 3D FEM simulation including the anisotrophy of the wood and the epoxi glue layers, done with the IBM version of ICES STRUDL 2 system. This recent development (which could be economically feasible) should be then included as one of the possible solution of the corner joints.
As shown by this development, a number of experts with some degrees of understanding and expertise in this technique have been increased significantly.
We do hope that the glulam timber industry will be interested in further development and in the use of the developed system.
At the present time they are not very interested in joining such project and this is quite strange knowing the present economical situation at home.
In the described KBS the structural calculations are done inside the program in a close form solution modul. For the intended inclusion of truss structures we are using a outside routine using some common programs. Only the preliminary dimensions for the structural calculations are as "thum rules" embodied in the system.

5. Discussion

There is one question sticking out in the discussion: the obvious deskilling of the structural engineers and the changed role of a new bred structural engineer's generation, also the scope of their education. What of the future? The new generation of computers will manage knowledge and therefore be capable of making decisions on the ground of quantity, and maybe quality too. Such systems should give expert assistance rather than replace completly the human thougth and decission processes.
As the first runs of our system started (with all of the subroutines debuged), we have discovered a system bug so that the system was giving us plausible but strange solutions. We discovered the bug in the system on the sole basis of the only past design and manufacturers experiences,the past experience and engineering logic, at the end the diploma candidate told me "the Germans have been right".
This poses a question of how to debug ES or KBS in the future when experience will be based on the past experiences gained by ES alone and not by experienced professional structural engineers.
How far then can we trust the system (with uncovered bugs in the system), and whose is the responsibility? How much trust we should have in future expert systems? Should we be cautious in the use of expert system, or try to verify them when still there is time left and experts around? As knowledge erodes the effort should be made to obtain good experienced structural engineers, capture their knowledge and educate makers of such systems as to avoid undiscovered bugs in systems on the sole basis of their past experience, until it is not too late [21].
It seems that the quality assurace of the ES should be administred by appropriate professional and learned societies or by institutions which can be entrusted with the task: exchange, validification and updating of data banks, programs etc. This is necessary but it seems to us that it will not be done until an outcry demands it. We reported the problem through various papers and also notified the SZS authorities to take charge of it on a higher (even international) level.
As for the development of the described system (which is operable

on PC) I am not very optimistic. The progress is by us slow and mostly not financed or is financed symbollicaly.
Confronted with economic crisis lack of hardware, (CAD systems, CAD stations and lack of plotters), an inadequate and virtually nonexistant information network, the lack of research funds, the diminishing standard, that I clearly believe we do not have much chance to go further from this described first step - the very beginning.

References:

1. **Arnold A.**, Artificial Intelligence, Prentice Hall, 1986.
2. **Aguilar R. J.**, System Analysis and Design, Prentice-Hall, 1973.
3. **Bundy A.**, Catalogue of Artificial Intelligence Tools, Springer-Verlag, 1986.
4. **T. Bernold**, Expert System and Knowledge Engineering, North-Holland, 1985.
5. **Baljkas B.**, Numerical modelling of timber structures (Numericko modeliranje drvenih konstrukcija), Simp. Contemporary structures: Timber structures of today, Brijuni, 265-294, (1987).
6. **DeNeufville R., J.H. Stafford**, System Analysis for Engineers and Managers, McGraw-Hill, 1971.
7. Diploma works, **S. Ferebuš, N. Petrinić, M. Miljak, Z. Glavinić**, FGZ Zagreb (1985-1988).
8. **Forsyth**, Expert systems, principles, Chapman Hall, 1984.
9. **Fenves, Rankin, Tejuja**, The Structure of Building Specifications, Build. Science Series 90, NBS, Washington D.C., 1976.
10. **Fenves**, Recent Development in Methodology for the formulation and Organisatio of Design Specifications, Eng. Str., Vol. 1., 1979.
11. **Garret Jr.**, A Generic Standard Processor for Use in Computer-Aided Structural Design, Ph. D. Thesis, Carnegie Mellon Univ., 1985.
12. **Haiman M.**, Types of glulam systems (Tipski drveni lamelirani lijepljeni nosači), Simp. Suvremene gradjevinske konstrukcije: drvene konstrukcije danas, DGKH, Brioni, 141-174, (1987).
13. **Howard, Fenves**, Representation and Comparison of Design Specifications, Dept. of Civ. Eng., R-83-141, Carnegie Mellon Univ., 1983.
14. xxxx, **ICES STRUDL** manuals.
15. **JUS Standards**, SZS, Beograd, 1985.
16. **P. Jorrand, V. Sigurev**, Artificial Intelligence I, II, North-Holland, 1986.
17. **Magerle M., Žagar Z., M. Haiman, B. Baljkas**, Optimization of two-hinged wooden glulam arches using FDM (Optimizacija dvozglobnih lukova od lameliranog drveta metodom konačnih diferencija), The 2nd Simp. Suvremene drvene konstrukcije, ZTI, Ljubljana, 145-156, (1986).
18. **Morton, Newman P.E.**, The Structural Engineering Design Programs, **Volume 1 - Wood Design**, The McGraw Hill Professional software Series, McGraw-Hill Books Co., N.Y., 1985.
19. **McNitt L. L.**, Basic Computer Simulation, TAB Book Inc., Blue Ridge Summit, 1983.
20. **MacGinley & T.C. Ang**, Structural Steelwork, Butterworths, 1987.

21. Oakley B. W., Expert systems: artificial intelligence research comes of age, Sci. publ. Affairs 3 37-51 (1988).
22. **Roth, Waterman, Lenst,** Building Expert System, Addison Wesley, 1983.
23. **Sablić S.,** Types of current industrial glulam structures (Tipizacija kurentnih industrijski proizvodjenih nosivih elemenata od ljepijenog lameliranog drva), Simp. Wooden structures, GI/FGZ/ISZ, Cavtat, B4-6-1 - B4-8-9, (1977).
24. **Sparks L. E.,** Investment Analysis with your Microcomputer, TAB Book Inc., 1983.
25. **Stark R. M., R.L. Nicholls,** Mathematical Foundation for Design: Civil Engineering System, McGraw-Hill Book Co., 1972.
26. **C. Weisbin,** Artificial Intelligence Applications, IEE CS Press/North-Holland, 1985.
27. **Wright,R.,** Virtual Reality, The Sciences, 11-12,1987., 8-10.
28. **Žagar, Z.,** O osnovama ekspertnih sustava u graditeljstvu (About expert systems in structural and civil engineering), VIII Kongres SDGKJ, Cavtat, 1987, T-91., 247-251.
29. **Žagar, Z.,** Expertni sustavi, Simposium Suvremene gradjevinske konstrukcije (Contemporary wooden structures),Bled,1987.
30. **Žagar Z.,** Nosive konstrukcije (Structures), 1, 2, 3, Liber, Zagreb, 1979.
31. **Žagar Z.,** A program for design of two-hinged glulam arches (Program za proracunavanje dvozgl. lukova iz lameliranog drveta), Simpozij '85, SDGKJ, Dubrovnik, Book R, 216-228. (1985).
32. **Žagar Z., B. Baljkas, M. Magerle, M. Haiman,** FEM modelling of glulam beams (Modeliranje drvenih lameliranih nosača MKE), Simpozij '85, SDGKJ, Dubrovnik, Book R, 435-442 (1985).
33. **Žagar Z.,** A new generation of standards for wooden structures (Nova generacija standarda za drvene konstrukcije), Simpozium on contemporary wooden structures, ZTI, Bled, 1-10 (1986).
34. **Žagar Z.,** The role of standards in the development of CAS and ES (Uloga standarda u razvoju CAD-a i ES), 9th Internat. Simp. CAD/CAM, Zagreb, 125-130, (1987).
35. **Žagar Z.,** Expert systems (Ekspertni sustavi), Simp.: Contemporary structures: Timber structures of todays, DGKH, Brioni, 411-482, (1987).
36. **Žagar Z.,** Experten systeme, Simulation & Prognosen (on 5 1/4" diskete), SVI BASIC Lib. #19, SVI/MSX-Club Deutschland, 1988.
37. **Žagar Z.,** The optimal solutions (Optimalna rješenja), The 2nd Simp. Suvremene drvene konstrukcije ZTI Ljubljana 301-312, (1986).
38. **Žagar Z.,** Automated design of lamella roof structures (Automatski dizajn mrežastih svodova), The 2nd Simp. Suvremene drvene konstrukcije, ZTI, Ljubljana, 329-349, (1986).
39. **Žagar Z.,** A system of programes for design of glulam beams (Sustav programa za dimenzioniranje drvenih lameliranih grednih nosača), Gradjevinar 37 (1985) 12, 499-507.

Knowledge-Based Systems in Civil Engineering (from CAD to KAD)

Systèmes à bases de connaissance en génie civil

Wissensbasierte Systeme im Bauingenieurwesen

Dietrich HARTMANN
Universitätsprofessor
Ruhr Univ.
Bochum, Fed. Rep. of Germany

Dietrich Hartmann, born in 1944, is Professor for Computational Engineering at the Ruhr University Bochum since 1987. From 1982 to 1987 he was Professor for Structural Mechanics and Optimization at the University of Dortmund.

SUMMARY
Planning and design in civil engineering require an integrated approach based upon a system engineering philosophy. Although numerical methods are very significant for the solution of the problems, engineering expertise and knowledge are much more central. Knowledge based systems (KBS) provide the potential of computerizing the expertise and knowledge of experts in specified knowledge domains. In particular, the problem solving behaviour of engineering experts can be simulated. It is demonstrated by means of distinct CAD/CAE examples what types of mechanisms are required to represent and evaluate engineering knowledge. It turns out that, in the future, a comprehensively hybrid philosphy is needed to obtain knowledge aided design or engineering (KAD/KAE), respectively.

RESUME
La conception et le dimensionnement en génie civil exigent une méthode intégrée basée sur un système d'ingénieur. Les méthodes numériques sont très importantes pour la résolution des problèmes, mais l'expérience et la connaissance sont plus importantes. Les systèmes à base de connaissance donnent la possibilité d'informatiser l'expérience et la connaissance des experts dans des domaines specialisés. On démontre qu'à l'avenir, la philosophie hybride est nécessaire pour obtenir le dimensionnement ou l'ingénierie assistée par les bases de connaissance à l'aide d'exemples concrets.

ZUSAMMENFASSUNG
Planungs-, Entwurfs- und Konstruktionsaufgaben im Bauingenieurwesen, erfordern eine ganzheitliche Sicht im Sinne der Systemtechnik. Numerische Methoden sind zwar eine wichtige Voraussetzung zur Lösung derartiger Probleme; ohne Ingenieursachkompetenz und -wissen aber ist keine ganzheitliche Lösung möglich. Wissensbasierte Systeme (WBS) bieten die Chance, Expertenwissen und damit das Problemlösungsverhalten von Ingenieurspezialisten zu computerisieren. An konkreten Beispielen aus dem Bereich CAE/CAD soll gezeigt werden, dass hybride Wissensrepräsentations- und Verarbeitungsmechanismen erforderlich sind, um das CAE/CAD zukünftig in ein Knowledge-basiertes CAE/CAD (KAE/KAD) überführen zu können.

1. INTRODUCTION

Knowledge based systems (KBS), one of the recent derivates of Artificial Intelligence (AI), provide the potential to incorporate knowledge into those engineering activities that for a long time could not be represented by means of algorithms. In particular, the problem solving behaviour of engineering specialists, being sophisticated in specified knowledge domains, can be captured provided that a sufficiently narrow knowledge domain is considered. Thus, KBS enable us to improve the "brute force computing" to an "inferential computing", an approach that is needed very much in CAD/CAE.

Although, KBS (also called expert systems) are said to be AI-research they are actually not such dramatic as often considered. In fact, KBS are nothing but a new software technology that permits the formalization and representation of knowledge as well as expertise provided that adequate representation mechanisms are available. AI researchers, who are working at the top of AI used to denote KBS as a methodology that no longer can belong to AI because of the tremendous success KBS have had in the past years!

2. REPRESENTATION OF ENGINEERING EXPERTISE

As we all know, engineering expertise plays the central role in engineering much more than numerical methods. While numerical methods are a cornerstone for the accurate analysis of physical or mechanical properties, "engineering-know-how" brings together the parts that really make an integrated system in the sense of the CIM-philosophy discussed all over the world. According to FEIGENBAUM, one of the fathers of AI, expertise and knowledge is characterized by the following: Even though a lot of professional work seems to be expressed in mathematical formulas the matters that set apart experts from beginners are symbolic, inferential and are rooted in experimental knowledge. This makes evident that experts have acquired their expertise not only from explicit knowledge found in textbooks and lectures but also from experience gained by doing things again and again, from eventually learning when to go by the book and when to break existing rules.

If we are able, to formalize and computerize knowledge and expertise, engineering related activities such as

- data analysis and interpretation,
- definition of engineering objects in relation to other objects by means of semantic nets,
- classification, diagnosis, selection, etc.
- formation (planning, modelling, developing, etc.)
- assistance and training,
- evaluation and verification,
- monitoring and control, etc.

can be coupled with traditional computational techniques. Precondition to the integration of numerical and knowledge oriented models is an adequate knowledge representation as already mentioned. Since there is a

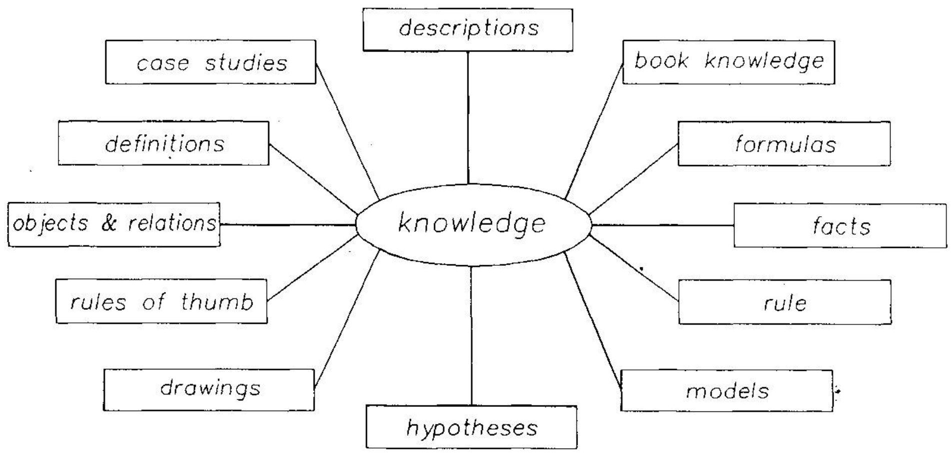


Fig. 1 Knowledge categories

wide variety of distinct knowledge categories (an precise definition is controversial but an enumeration like that in Fig. 1 may be sufficient) it is obvious that also a wide variety of representation mechanisms is mandatory. Present research of KBS indicates that rule based paradigms in association with object-oriented paradigms and blackboard techniques as well as semantic nets (frames, scripts, events, etc.) provide the versatile tool needed for engineering problems. If all of the above mentioned paradigms are combined within one single system this system is called a hybrid system. (Typical examples are KEE, ART, TWAICE, KNOSSOS, just to name a few.)

A further characteristic of KBS is the strict separation between knowledge and operational mechanisms (inference engine) that act on the knowledge and infer new knowledge from existing one. However, apart from these two fundamental components additional components are necessary for practical applications (see. Fig. 2).

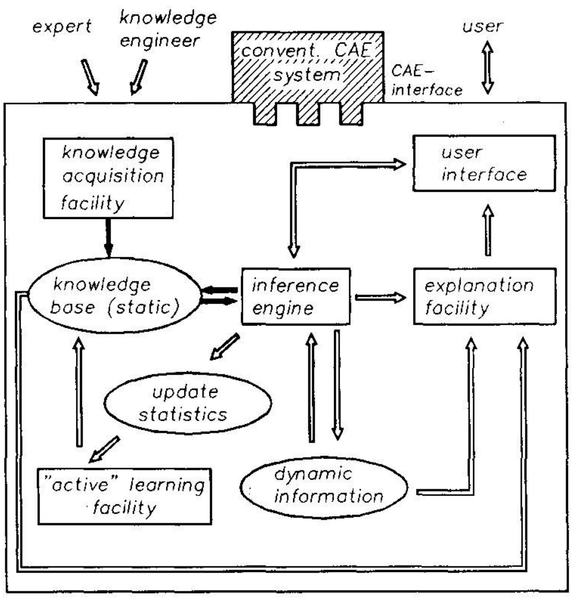


Fig. 2 KBS architecture

Thus, according to Fig. 2 we have a knowledge acquisition facility that supports all activities needed to acquire knowledge in a computer readable format and an explanation facility to make transparent conclusions and inference paths. For real world problems in engineering an interface to existing conventional CAE-software is absolutely necessary. Very recently a learning facility that allows updating of knowledge has appeared. However, active learning adaptation is still a matter of AI research.

Interfacing conventional software with KBS will yield the following scenario in the nearest future (see Fig. 3).

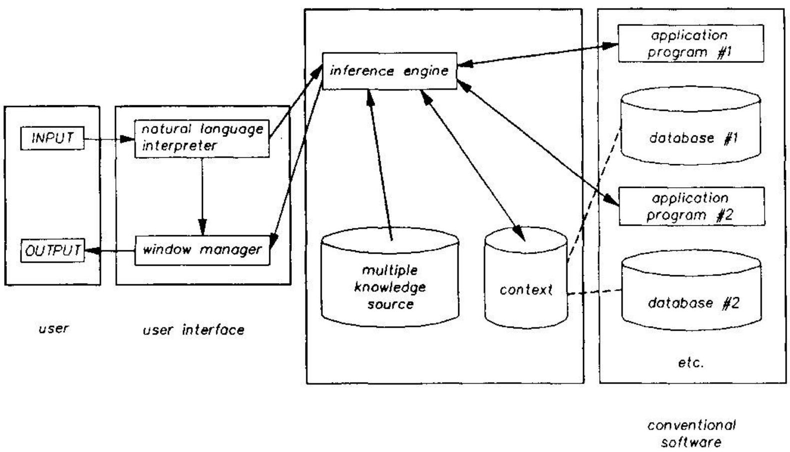


Fig. 3 Conventional CAE coupled with WBS

Fig. 3 indicates that a natural language description may be used for the problem definition. (Natural language systems and pattern recognition are also AI-disciplines that are becoming more and more important to engineering problem solutions.) Fig. 3 also demonstrates that the KBS methodology and conventional software is integrated into one computer system.

3. EMBEDDING ENGINEERING KNOWLEDGE INTO CAD

The computerization of engineering knowledge leads to substantial changes or, at least, major modifications in conventional CAD. The fundamentals of conventional CAD systems, exclusively written in traditional (procedural) computer languages like FORTRAN or C, are graphic oriented entities such as lines, arcs, polygons, cubes, etc. These graphic primitives are well suited to make impressive drawings. However, if design and manufactoring is considered, or the complete horizontal CIM-life cycle of a construction from planning and preliminary design over "final" design to fabrication including management is to take into account, then drawing is only one aspect among others. In this case, a much more sophisticated approach is needed. All over the world, researchers and software specialists in the CAD-domain are aware or getting aware of the significance that the embedding of knowledge into CAD captivates. As a consequence, numerous attempts have been undertaken to incorporate knowledge into CAD. Some of the most interesting categories of realisation will be discussed in the following subchapters.

3.1 INTELLIGENT CAD

When CIM-applications are to be addressed, a CAD system must devise its own methods for defining data "objects" or "entities" and retrieve them from a data base. Processes such as the creation of lists or data structures, however, are not supported by traditional languages. They may be simulated but then traditional programs become very large. Also, such programs are expensive to debug or modify. Therefore, an "object-oriented philosophy" has to be developed to automate the design process efficiently.

From the author's point of view, the ICAD-system [1] is the first CAD-system designed from the ground up to employ an object- or feature-based data structure. The ICAD-system has a language structure that is more suited to the way designs take shape than do traditional languages. As each part of the design is invented, the designer creates the part using standard component features, then defines the rules for connecting the part to the structure or machine. The program which describes the part is called an "object". Thus, objects may be created as they are needed, then linked to other objects in a very natural fashion. The process allows the designer to build his parametric objects part by part, testing each part as it is created. This piecewise approach parallels the design thought process very much.

In contrast to this method, in FORTRAN e.g. the entire program would have to be written, compiled, debugged, linked and loaded before any portion of it could be tested. When traditional languages are applied to automate design work, the engineering of software tends to become more complex than the design itself. This holds particularly for program modifications.

In a general sense, the process of engineering consists to large part in manipulating symbols and linking those symbols into lists in accordance with certain rules. The rules may stem from mathematical formulas or other type of knowledge (e.g. technological requirements such as "bolt X must fit into bolt hole Y"). In order to represent the knowledge there are three basic components in ICAD:

- A symbolic language for product description.
- A graphic browser for viewing and editing the product design.
- A relational query language for retrieving parts from an existing library.

The symbolic language (ICAD) is based upon LISP but easier to handle by engineers, and directly assists the engineering work. The versatility clearly shows evidence that LISP, as an AI-language, fits the demands of an advanced CAD.

In order to elucidate the ICAD system more precisely, a simple demonstration object (see Fig. 4) with a description of its construction is to be presented. The structure considered represents a simple 3-D-frame structure called "speakers platform".

On the left side the part-whole tree can be seen that depicts the tree hierachy of the construction consisting of a floor, a horizontal frame (grid) and four legs. In the lower right-hand corner one can find the 3-D-representation of the structure as a simple isometric view. At the top right is a window for inspecting the values of the attributes of any object defined.

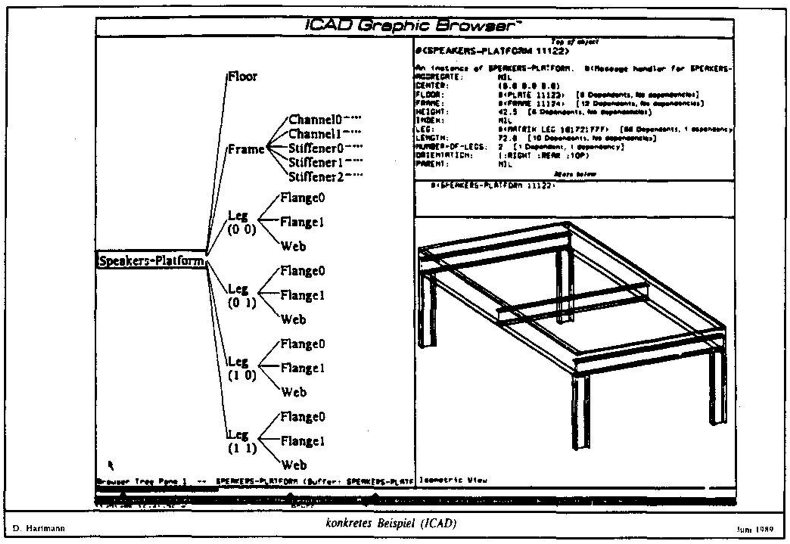


Fig. 4 Object - oriented ICAD

```
(DEFPART     SPEAKERS PLATFORM         (box)
 :DEBUG-MODE ?    t
 :INPUT  (:length  :width  :height)
 :ATTRIBUTES
 (:number-of-legs                    (celling ( the :width ) (feet 20)))
 :PARTS  ((FLOOR    :type            plate
                    :position        (:top 0.0)
                    :height          (inch 7/8))
          (FRAME    :position        (:below (:from (the :FLOOR) 0.0)))
          (LEG      :type            aisc:W6x12
                    :quantify        (:matrix :lateral the :number-of-legs) :longitudinal 2)
                    :position        (:below (:from ( the :FRAME) 0.0))
                    :orientation     (:rotate : right )
                    :length          (- ( the: height )
                                     ( the :FRAME :height )
                                     ( the :FLOOR :height )))))

(DEFPART     FRAME     (box)
 :DEBUG-MODE ?    t
 :INPUT  (:length  :width  :height)
 :ATTRIBUTES (:height                ( the :CHANNEL:any :height )
             :number-of-stiffeners   (celling ( the :width) (feet 20)))
 :PARTS ((CHANNEL   :type            aisc:PC10x3x1/4
                    :quantify        (:pair :longitudinal )
                    :orientation     (:rotate : top )
         (STIFFENER :type            (:if ( ≥ ( the : length ) (feet 5))
                                              'aisc:W6x12
                                             'aisc:L2x2x3/8  )
                    :quantify        (:series :lateral ( the :number-of-stiffeners))
                    :length          (:between ( the :CHANNEL) : web ))))
```

Fig. 5 Editor window in ICAD

Fig. 5 shows an editor window which contains the design rules for the speakers-platform and demonstrates the way the design rules for the given structure can be established through the ICAD language. Also, ICAD's similarity to LISP becomes evident. In particular, the "DEFPART"-keyword allows the user to define any number of object attributes. Attributes may be orientation, position, length relative to individual parent objects, additional information with respect to fabrication, management etc. The keyword "PART" creates the part-whole hierachy that is graphically represented in the tree display parallel to its creation.

The attributes of any object in the structure can be referred to by any other object through a symbolic reference scheme. Such a reference is stated by means of the word "THE" and contains a path from one object to another. This concept materializes a semantic network of dependencies between objects. Also, this semantic net naturally represents the taxonomy knowledge of structural components within a total construction or building. Furthermore, the application of relations between objects incorporates a form of inheritance between "parent" and "child" objects similar to the inheritance mechanism used in the frame paradigm in expert systems.

Another kind of dependency is created in the part "STIFFENER". Modifications to a reference configuration are set up in terms of production rules of the type:

IF premise THEN conclusion, or
IF event THEN reaction.

In our current example a production rule for the part "STIFFENER" is used to determinate a stiffener:

IF length of stiffener ≥ 5 feet
THEN use W6X12 beam
ELSE use L2X2X3/8 angle.

Using more sophisticated conditionals allows arbitrary complexity. Thus, besides object oriented concepts, taxonomies and semantic nets, ICAD also provides the standard rule paradigm of rule based systems. The rule paradigm provides the potential to create general constructional knowledge bases, but company specific knowledge bases are possible as well.

To summarize the features of ICAD, the ICAD-system is based upon knowledge of objects and their reaction to alternations and modifications. Instead of working with absolut data a parametrical design is specified taking into account all possible alternatives. Therefore, the design process can be shaped identical to the natural way a designer proceeds: First, a rough model is designed, then it is refined stepwise to become more and more detailled. The incremental logic of design associated with the possibility to make alternations at any time makes ICAD the most advanced CAD system the author knows.

Despite the advantages of ICAD there are some drawbacks. ICAD does not allow an interactive design procedure exclusively based on graphic modelling. Instead, the object oriented view requires a programming in terms of ICAD commands. In other words, to acquire improved semantics, programming in a LISP-like language is required. The final ICAD design, however, may be transferred to a conventional CAD-system that contains graphic capabilities of a high quality. Another drawback is the high cost of ICAD that currently runs on symbolic-workstations only (total costs about US-$ 250.0000)

3.2 KNOWLEDGE BASED PROGRAMMING TECHNIQUES

High costs are a crucial obstacle for civil engineers in particular. Therefore, it is clear that in civil engineering all attempts are made to increase the intelligence of existing CAD-systems without producing astronomically high costs. In this context, two examples may exemplify this approach, the first example is the BERT expert system, the second is the research carried out by one of the author's co-workers.

The BERT system [2] is a knowledge based system that links together a conventional CAD design for parts of a brickwork building with a rule based system capturing the standards for brickwork design. Starting from a conventional CAD drawing (AUTOCAD) relevant facts from the drawing or the internal data base are extracted and converted as a context data base for the rule based expert system. The expert system attempts to bring into conformity the given facts with the knowledge incorporated in the knowledge base. The inference engine chains the facts with the rules in a backward chaining manner in order to infer that all the standards hold. If not so, comments and suggestions are given such that an appropriate construction can be created. This cycle continues until the expert system is unable to find anything wrong.

The second example demonstrates that intelligence can also be incorporated in the interior of a conventional CAD system, by enhancing the command and menu structure of a CAD system. In this case the knowledge based character is achieved by AI-language-based programming rather than by creation of a knowledge base or inference engine. Based upon the aforementioned AUTOCAD-system, representing a worldwide quasi standard, a knowledge based pre-processor for structural analysis is in process (see [3]). Starting with an architectural CAD-model a structural system is prepared for structural analysis, where a finite element method is taken as the fundamental computational procedure. It is well known that modelling a finite element model is cumbersome, time consuming and prone to errors, particularily if large scale structures are considered. Therefore, intelligent aids for finite element modelling are desired very much. In this context, intelligence is understood in the sense that CAD-systems link knowledge about structural data and properties with the CAD-geometry. This link is accomplished through AUTOLISP, a derivate from common LISP with an adaptation to AUTOCAD. AUTOLISP is an adequate language for manipulating symbols in terms of lists. The symbols may be words (e.g. AUTOCAD words), numbers or other lists of symbols.This ability makes AUTOLISP very powerfull for augmenting AUTOCAD with intelligent mechanisms.

Just to give a short impression of how AUTOLISP constructs work the structural system in Fig. 6 is considered.

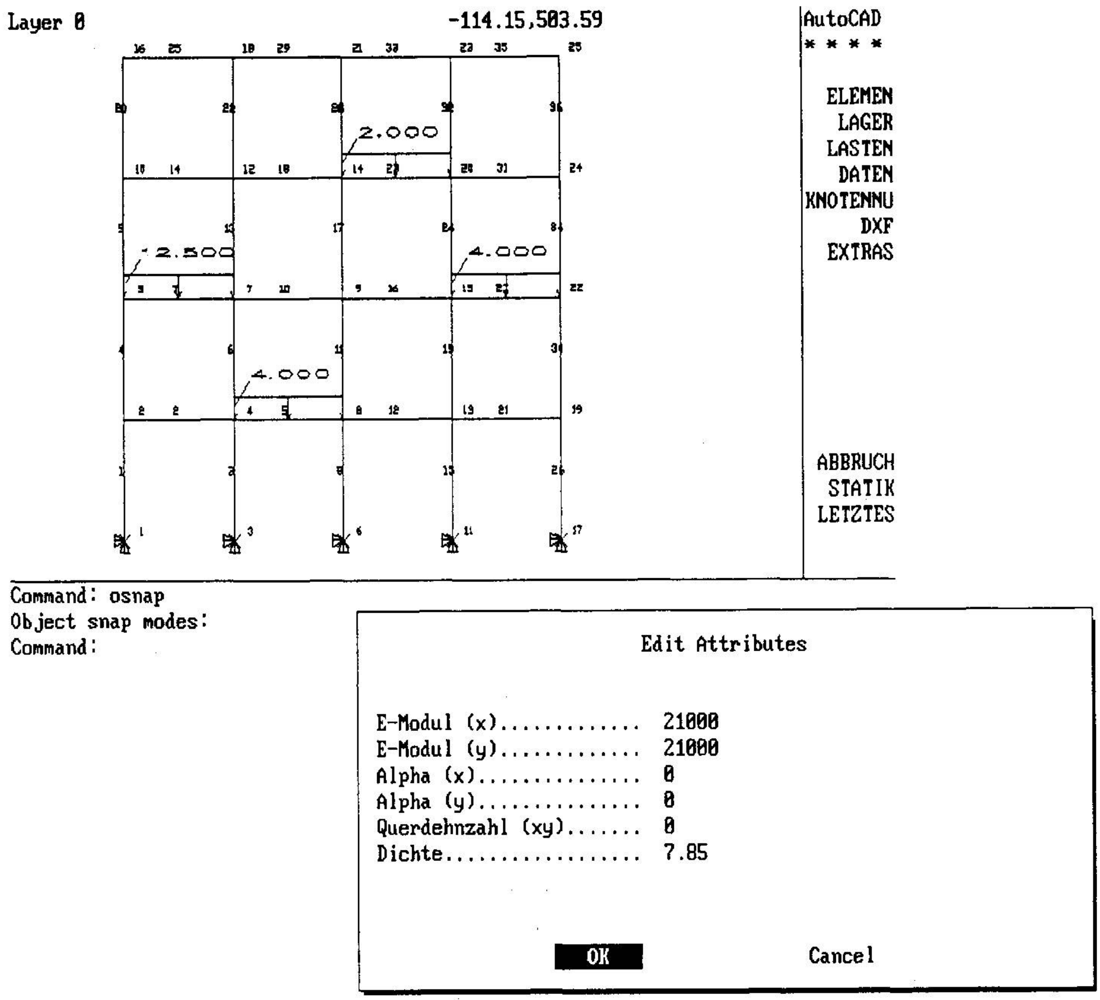


Fig. 6 "Intelligent" AUTOCAD

Whereas in conventional CAD only graphic primitives (lines, arcs, strings, etc.) can be identified without any information on operational steps subsequent to the architectural design (like structural analysis, management, fabrication, etc.) AUTOLISP constructs embed corresponding knowledge into the geometrical entities used. The AUTOLISP module for the above example is shown in Fig. 7. It can be seen that through this module the internal geometry is linked with "computational information".

```
.
.
(if (= ELEMTYP "BIEGE2D")
  (progn
    (while (= EMODULX nil)
      (setq EMODULX (getstring "\nE-Modul : "))
    )
    (setq EMODULY EMODULX)
    (while (= ALPHAX nil)
      (setq ALPHAX (getstring "\nAlhpa : "))
    )
    (setq ALPHAY ALPHAX)
    (while (= NUXY nil)
      (setq NUXY (getstring "\nQuerdehnzahl (xy) : "))
    )
    (while (= DICHTE nil)
      (setq DICHTE (getstring "\nDichte : "))
    )
  );progn
);if

(command "layer" "m" (strcat "DAT" (itoa LNR))
  "c" LNR (strcat "DAT" (itoa LNR)) "")
(command "insert" (strcat PFAD "material") "0,0" (/ FAKTOR 10) "" "0"
      ELEMTYP (itoa LNR) EMODULX EMODULY ALPHAX
      ALPHAY NUXY DICHTE)
.
.
```

Fig. 7 AUTOLISP example

The structural data are converted in terms of AUTOCAD block data structures that can easily be evaluated for a subsequent finite element analysis. The evaluation is performed by means of a PROLOG programm (another AI language) to provide a standard format (FEDIS) accepted by a variety of finite element codes. (Of course, the structural knowledge could also be applied in the BERT fashion; a separate expert system could be created to check the appropriateness for finite element input, for instance with respect to input requirements).

4. KNOWLEDGE BASED CONCEPTS IN STRUCTURAL ANALYSIS AND DESIGN

Besides intelligent pre-processors for structural analysis of the type discussed in the previous chapter there are further domains that are well suited for knowledge based systems. However, at first it should be explicitly pointed out that computational methods themselves are not a matter of knowledge based systems. Since computational methods are founded on consistent theories, that are definitely accurate within prescribed application limits, there is absolutely no necessity for knowledge based approaches. That is to say that knowledge based systems are exclusively successful in cases only where

- no accurate theoretical concept is available,
- a diffuse complexity is present or
- a solution has to be based on permanently available expertise acquired in years of training.

Typical categories of problems that qualify are:

- selecting appropriate solution methods,
- monitoring computational processes and
- assisting and consulting in order to navigate complicated phases in computation.

To exemplify the potential that the knowledge based approach presents, again, two application domains are dealt with.

The first application for structural analysis and design is a knowledge based system created for assisting and consulting a mechanical engineer to identify bifurcation or limit points of a given stability problem [4] according the theory of linear elasticity. Stability problems are characterized by the fact that a specified load level may results in various equilibrium conditions depending on the nature of the problem (snap through problem associated with limit points; bifurcation problems yield primary and secondary equilibrium paths, see Fig. 8).

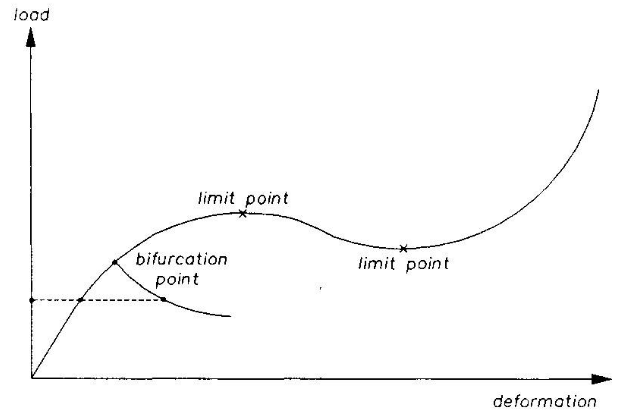


Fig. 8 Stability problems

A knowledge based approach provides the possibility to identify the nature of the critical points (whether they are limit or bifurcation points). The identification process is based upon explicit rules captured in a knowledge base in which new knowledge can be added if detected. In order to draw conclusions during the computational process (e.g. whether a critical point is occuring, what category of point is detected, what method is to be used to procede, etc.) numerical output (e.g. determinant of tangential global stiffness matrix, eigenvalues and eigenvectors) is converted into qualitative facts needed for the inference mechanism applied (modus ponens and backward chaining).

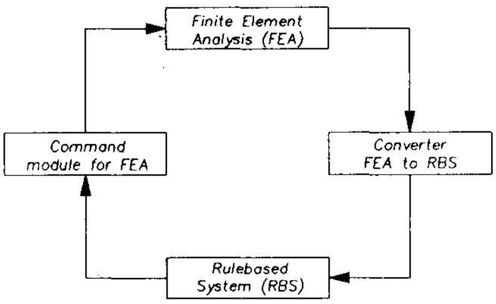


Fig. 9 Architecture of consulting expert

The v. Mises two-bar system illustrates the capability of a knowledge based navigation in complicated computational scenarios:

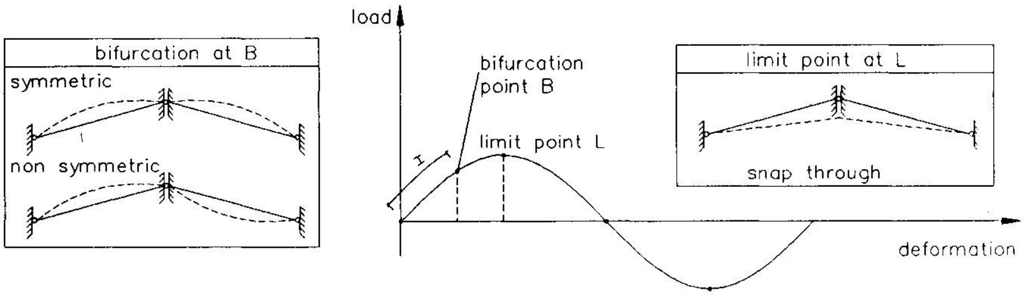


Fig. 10 v. Mises stability problem

The rule based system (RBS) identifies critical points and does consulting in the following fashion (short form) according to Fig. 10:

computational phase	RBS reaction
within part I of the computation:	according to rule # XXX instability has to be expected, confidence 100;
after exceeding point B	according to rule # YYY instability active; confidence 100; explanation: negative diagonal element in the tangent stiffness matrix measures taken: check on category of instability thru eigenvalue evaluation; switch over to arc-length method and back iterate to point B
after a while according to eigenvalue and eigenvectorcheck	instability due to bifurcation
etc.	etc. .....

Although the sample test is elementary it demonstrates that, along with knowledge based systems, computational mechanics is developing from the rather "brute force technique" to a more "inferential computing", as mentiosed at the beginning.

The second application for structural analysis and design addresses the post-processing of structural analysis. Post-processing is not only restricted to the customarily used graphical representation of computational results, in addition, the structural component design is ascribed to post-processing.

The major part of structural component design is knowledge based because the design itself has its roots in standards that contains a diversity of knowledge formats (facts, rules, tables, formulas, comments, figures, etc.). In this context, the research carried out at the civil engineering department of the CMU (Carnegie Mellow University, Pittsburg, PA, USA) under supervision of Prof. Fenves, deserves particular mention. The Standards Processing Expert (SPEX) [5] links numerous distinct knowledge sources within the design process (book knowledge, knowledge on objects, experience, standards) by virtue of a blackboard (see Fig. 11). A blackboard is a central medium with which separate knowledge sources (they may even be written in different languages) communicate. The inference engine schedules the flow of conclusions in the blackboard and monitors the various activities of the knowledge source.

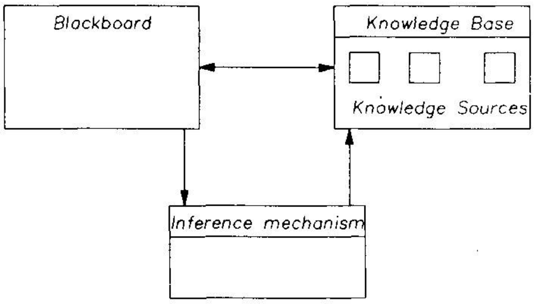


Fig. 11 Blackboard architecture

Current research of one of the co-workers of the present author (see [3]) focuses on the computerization of the new DIN18800, part 2 (stability problems in steel structures). The given standards are transferred to a rule based system utilizing PROLOG-production rules. Currently, also an object oriented expert system shell (TWAICE) is examined. Just to give an insight in the PROLOG rule base a rudimentary PROLOG representation of a production rule (in PROLOG called implication) is given in Fig. 12

DIN 18800 Teil 2 (Entwurf)

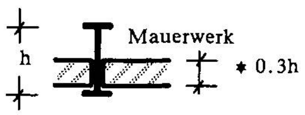


Bild E303.
Aussteifung durch Mauerwerk

3.3.2 Behinderung der Verformung
Behinderung der seitlichen Verschiebung
308 - Ausreichende Behinderung der seitlichen Verschiebung ist vorhanden bei Stäben, die durch ständig am Druckgurt anschließendes Mauerwerk ausgesteift ist, dessen Dicke nicht geringer iat als die 0.3fache Querschnittshöhe des Stabes.

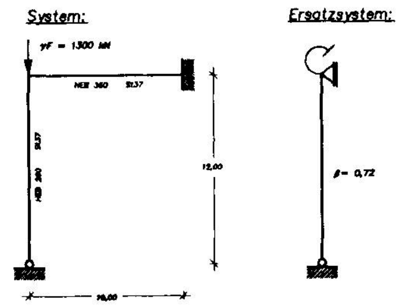
PROLOG

```
% . . . Aussteifung durch Mauerwerk ( 3.3.2.308 )
nachweis (abschnitt_3_3_2,Achse,Nquer) :
,  global (querschnitt,i (_))
,  eingabe_num ('Mauerwerksdicke am Druckgurt','cm',Dicke)
,  global (profilhoehe,H)
,  Dicke >= 0.3 * H
,  Nquer is 0
.
```

Fig. 12 PROLOG production rule example

Based upon production rules of the above type a DIN-adequate check on stability is performed through the backward chaining mechanism incorporated in the PROLOG interpreter (Arity-PROLOG). A PROLOG session in Fig. 13 indicates how the knowledge based system works for a elementary test sample.

In the test sample (the frame system is transferred to a single bar object with a centric compressive load according to the "Einzelstabverfahren" in DIN 18800) the input for the PROLOG post-processor is interactively accomplished (for practical application the input has to be taken from an output file of a finite element run or a corresponding structural analysis programm).




```
?- din18800 t2(Y,Z).
Material [st37,st52,ste460,ste690] :  st37.
Querschnittstyp [kreis,kreisrohr,rechteck,i(geschweisst),kasten(geschweisst),
ipb,heb] :ipb.
Profil [100,120,140,160,180,200,220,240,260,280,300,320,340,360,400,450,
500,550,600,650,700,800,900,1000] :  300.
Knicklaenge um y [m] :        12.
Knicklaenge um z [m] :        12*0.72.
Gamma-fache Schnittkraefte links
    Normalkraft [kN]  :        1300.
    Moment um y [kN*m] :  0.
    Moment um z [kN*m] :  0.
Gamma-fache Schnittkraefte rechts
    Normalkraft [kN]  :        1300.
    Moment um y [kN*m] :  0.
    Moment um z [kN*m] :  0.
*****************************************************************
***  Ergebnisse:                        *****
*****************************************************************
Abschnitt 3.2.1: nquer(y) -> 0.66506616
           nquer(z) -> 0.95088999
Abschnitt 3.2.2: nquer(y) -> 0
           nquer(z) -> 0
?-
```

Fig. 13 PROLOG session for checking expert system

5. CONCLUDING REMARKS

In the preceding chapters distinct application fields of knowledge based concepts have been tackled. It was intended to demonstrate that knowledge based approaches have many facets and may be materialized in different ways. Despite the diversity of candidate representation formats and practical realisations, conventional CAD/CAE will definitely augmented and modified from "simple" 1st-generation CAD towards more sophisticated 2nd-generation CAD.

REFERENCES:

1. ROSENFELD L.,
   Breaking through the Complexity Barrier Internal ICAD-Paper, ICAD, Incorporated, Cambridge, Massachusetts, USA, 1986

2. BOWEN J.A:, CORNICK T.C., BULL S.P.,
   BERT - An Expert System for Brickwork Design. Internal Paper, Department of Computer Science and Construction Management, UniversitY of Reading, England, 1986

3. HARTMANN D., CASPER E., LEHNER K., SCHNEIDER H.-J.,
   Einsatz von Expertensystemen im Umfeld von Finite-Element-Applikationen. Tagung Finite Elemente in der Baupraxis, RU-Bochum, März 1988

4. TARNOW N.,
   Das Expertensystem GENIUS, Interaktive Steuerung nichtlinearer, struckurmechanischer Lösungsalgorithmen im Finite-Element-Programm PCFEAP Diplomarbeit am Institut für Baumechanik (Prof. Stein), Universität Hannover, 1987

5. GARRET J.H., FENVES S.J.,
   A Knowledge Based Standards Processor for Structural Component Design. Report # R-86-157, Carnegie Mellon University, Dept. of Cicil Engineering, Pittsburgh, Pennsylvania, Sept. 1986

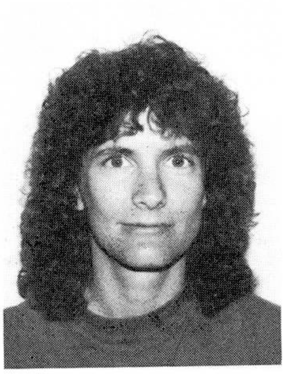
Leere Seite
Blank page
Page vide

Synthesis of Structural Systems

Synthèse du projet de construction

Synthese des Bauentwurfs

Mary Lou MAHER
Associate Professor
Carnegie Mellon University
Pittsburgh PA, USA



Mary Lou Maher received a BS from Columbia University in 1979 and a MS and PhD from Carnegie Mellon University in 1981 and 1984 respectively. Her teaching and research interests include computer aided design and AI applications to structural design.

SUMMARY
This paper discusses an expert system approach to the synthesis phase of structural design. In addition to formalizing the design knowledge for designing structural systems, a synthesis algorithm is developed. The use of such an algorithm in developing expert systems for structural design facilitates the development of a knowledge-base. This approach is illustrated with applications to the design of structural systems for buildings.

RESUME
Cet article expose une des approches par système expert de la phase de synthèse dans la conception d'une ossature. En plus de la formulation de la base de connaissance pour la conception des ossatures, un algorithme de synthèse est développé. L'utilisation d'un tel algorithme dans le développement des systèmes experts pour la conception d'ossatures facilite le développement d'une base de connaissances. Cette approche est illustrée par des applications concernant la conception d'ossatures de bâtiments.

ZUSAMMENFASSUNG
Die Arbeit diskutiert eine Möglichkeit eines Expertensystems für die Synthesephase des Gebäudeentwurfs Zusätzlich zur Formalisierung des Wissens für die Konstruktion von Gebäudesystemen wird ein Synthese-Algorithmus entwickelt. Die Verwendung eines solchen Algorithmus erleichtert die Schaffung einer Wissensbasis bei der Entwicklung von Expertensystemen für die Konstruktion. Dieser Weg wird anhand von Anwendungen für Hochbauten dargestellt.

1. INTRODUCTION

Structural design includes the synthesis of a structural system that satisfies a set of requirements. Synthesis can be considered at several levels of abstraction, where more information about the requirements as well as the evolving design description is available as the process proceeds. In this paper, the focus is on the early stages of design where the design knowledge is largely qualitative. During the early stages, or preliminary design, the major components or subsystems are identified and their composition is evaluated. Although the identification and composition may make use of associated quantitative models, the designer typically reasons about these models in a qualitative manner.

To support the designer in the identification and composition of components of structural systems requires both synthesis and evaluation methods. Such methods can provide a systematic approach to design, allowing the designer to pursue more alternatives and to evaluate the alternatives based on a discourse of criteria and value. The use of an expert systems approach for the exploration of alternative structural systems maintains a separation of method and knowledge, allowing the designer to guide the methods with qualitative or empirical knowledge without sacrificing the benefit of a systematic approach.

In this work, the synthesis of structural systems is based on a constraint directed search through a design space that is decomposed into components, subsystems and constraints. Evaluation of alternative structural systems is based on the concept of Pareto optimality, where multi objective optimization provides a basis for identifying a set of optimal solutions among a set of feasible solutions. Both synthesis and evaluation are integrated in a single model for producing alternative design descriptions for a given set of requirements. This model has been implemented as an environment for developing expert systems, where the experienced designer defines a knowledge base and the designer uses the resulting knowledge base to produce design solutions.

2. SYNTHESIS

There are many books that provide definitions and elaborations of the design process; in structural engineering such books include [3], [4], [2] and [1]. The design process can be considered as comprising different phases, synthesis being one of these phases. Although the phases may not be addressed hierarchically for the entire design process and are often carried out recursively, there is an inherent order in which designers approach a design problem. The following represents one formalism of the design process.

- *Formulation* involves identifying the goals, requirements and possibly the vocabulary relevant to the needs or intentions of the designer.
- *Synthesis* involves the identification of one or more design solutions within the design space elaborated during formulation.
- *Evaluation* involves interpreting a partially or completely specified design description for conformance with goals and/or expected performances. This phase of the design process often includes engineering analysis.

Formulation occurs at some level of abstraction and provides enough information to begin a synthesis process. The result of formulation is usually a set of design specifications. For example, the design of a 30 story office building with a regular 25 foot grid represents a partial set of specifications. Synthesis involves identifying the form of the design solution. For the office building, the result of synthesis may be a set of steel rigid frames along the grid lines with a reinforced concrete floor slab. Evaluation, during the early stages of design, is usually based on a subjective assessment of relevant criteria. For example, alternative structural configurations may be evaluated according to cost estimates, ease of construction, and stress-strain requirements. The knowledge used during synthesis and evaluation of preliminary designs is not well articulated. Experienced designers resort to trial and error less frequently than novice designers when searching for an appropriate or satisfactory form, suggesting that the use of an expert system to represent 'experience' may improve design synthesis and evaluation. The problem is how to represent this experience.

During synthesis a designer considers a design space which contains the knowledge that is used to develop the design solution. For structural design, a design space may include different types of framing systems, floor systems, wall systems, and materials. A human designer does not explicitly identify his design space, it is implicitly developed and expanded as he gains experience. A design program, however, does contain an explicit representation of the relevant design space. The formalization of the knowledge in the design space is of interest when considering an expert system approach to structural design.

3. EDESYN

EDESYN (Engineering DEsign SYNthesis) is a software environment for developing expert systems for design. The development of EDESYN was modelled after the expert system "shell" concept. An approach to developing an expert system for structural design was implemented as HI-RISE [5]. This approach was generalized and expanded to facilitate the development of expert systems for design. The design method is implemented as an algorithm to serve as an inference mechanism. The design knowledge is structured to provide a formalism for developing a knowledge-base.

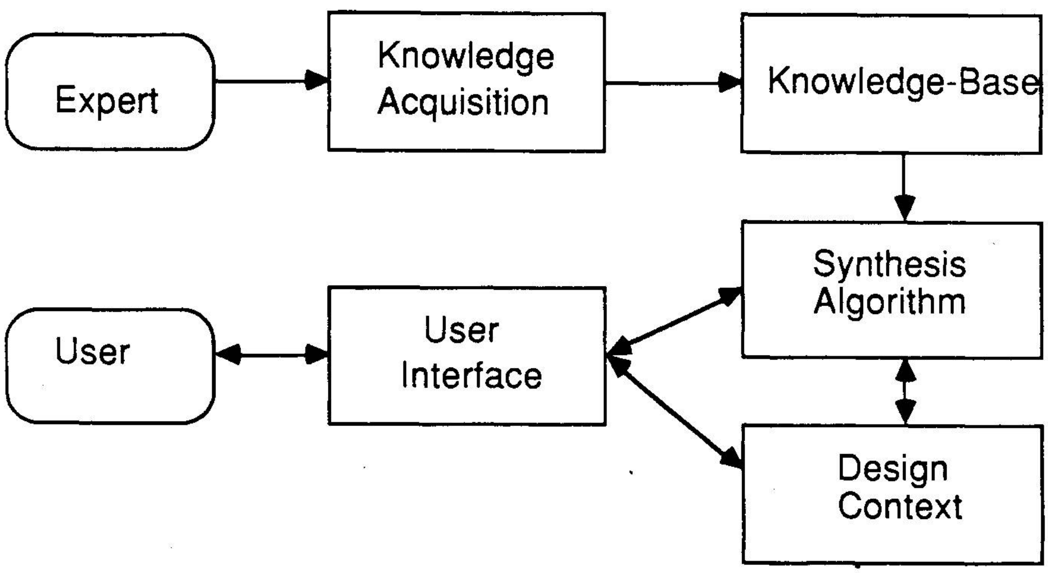


Figure 1: Architecture of EDESYN

EDESYN consists of five main modules: design knowledge-base, synthesis algorithm, design context, user interface, and knowledge acquisition facility, as illustrated in Figure 1. When using EDESYN, the knowledge acquisition facility is invoked first. During knowledge acquisition, the domain specific knowledge is read from files prepared by a domain expert. The domain specific knowledge is stored in the knowledge-base and the synthesis processor is invoked. The user then provides the problem specific information through the user interface to initialize the design context and guides the synthesis of design solutions to augment the context.

The knowledge-base includes decomposition, planning, constraint, and evaluation knowledge. The decomposition knowledge is specified as systems and subsystems, where each system is comprises a set of attributes. An attribute may be another system (i.e. subsystem), representing a synthesis node in a goal tree, or a simple attribute, representing a terminal node. The synthesis node is specified by another system. The terminal node is specified as a selection from a set of discrete alternatives or the evaluation of a Lisp function. The planning knowledge is associated with the system to identify the relevant attributes for the current design situation and the order in which the attributes should be considered.

An example of a system definition for designing the lateral load resisting system for a building is:

```
(system lateral
        3D-lateral one-of (core orthogonal-2D)
        2D-lateral subsystem 2D-lateral
planning
        If stories < 5 Then 2D-lateral
end system)
```

The design of a lateral load resisting system is described by the 3D lateral system and the 2D lateral system. The 3D lateral system can be selected from a set of alternatives and the 2D lateral system must be synthesized. The planning rule indicates that buildings with less than 5 stories should only have one attribute, i.e. the 3D lateral system is not appropriate.

The constraints are specified in the knowledge base as elimination constraints, where each constraint is a combination of design decisions and design context that is not feasible. The constraints are used during the synthesis process to eliminate infeasible alternatives. Examples of constraints in the structural design knowledge base are:

```
IF
stories > 30
3D-lateral = orthogonal-2D
THEN not feasible

IF
2D-lateral-x/material = steel
2D-lateral-y/material = concrete
THEN not feasible
```

The first constraint eliminates a 2D-orthogonal lateral system for buildings with more than 30 stories. The second constraint ensures that a concrete system is not built in the y direction if the lateral system in the x direction is defined to be steel.

The evaluation knowledge is specified by a set of criteria for each synthesis node or system. A criterion is described by a label, a weighting factor, a non-dimensionalizing factor, a normalization factor, and a function to determine the value of the criterion for a design solution. Example criterion for the lateral system are stiffness, compatibility, cost, and ease of construction. The value for each criterion is assessed using qualitative knowledge about structural systems since there is not enough information during preliminary design for a quantitative analysis. For example, stiffness could be assessed in a relative manner, where the designer knows that in most cases a braced frame structure is stiffer than a rigid frame structure.

The synthesis algorithm uses the design knowledge in the knowledge base to produce feasible design solutions consistent with the context. The overall algorithm is based on a constraint directed depth first search through the systems in the knowledge-base. The attributes of each system are assigned all legal values, where a legal value is one that does not get eliminated by the constraints. All feasible combinations are generated for each system, using the planning rules to define and order the attributes. After the alternatives for a system have been synthesized, the evaluation mechanism is invoked. The alternatives are compared for each criteria to produce a set of non-dominated solutions, which are then ranked using the preferences specified by the weighting factors. At this point, the solutions are presented to the designer along with the evaluation information and the designer chooses one solution for further consideration.

The design context initially contains the requirements and specifications associated with a particular design problem. For example, the intial context for a structural design problem includes the number of stories in the building, the occupancy, the structural grid, etc. The context expands as synthesis proceeds to include a tree of alternative solutions, where each node in the tree represents a solution for an attribute of a system. Along with the solution tree, a hierarchy tree is maintained to associate each attribute in the solution tree with the system for which it was generated.

The user interface is implemented using a multi-window, menu driven interaction style. During the design synthesis process, the user can view and change the design specifications, monitor the synthesis process as a tree of solutions is generated, and view a single solution in more detail.

The knowledge acquisition facility transforms the information provided by the expert to the frame based representation of the knowledge base. The expert provides the following design knowledge: preconditions, decomposition, constraints, evaluation criteria, and functions. The design knowledge is specified in a simple syntax and stored in files. Preconditions are specified as a set of names, default values, and allowable ranges. For example, one precondition may be wind load and its default value 30 psf, and its allowable range > 0.0. Decomposition knowledge includes the systems, subsystems, attributes, and planning rules. The constraints are specified as infeasible combinations of elements. Each evaluation criterion is sepcified by a name and a procedure for assigning a value using the goals and elements associated with the current solution. Functions are specified as Lisp functions that use the current state of the design solution to calculate the value of a parameter.

4. STRYPES AND STANLAY

EDESYN has been used to develop two expert systems for structural design: STRYPES and STANLAY. STRYPES generates alternative combinations of structural systems and materials for a given building. STANLAY accepts a feasible combination of structural systems and materials for a given building and generates alternative layouts and approximates the load requirements for the structural components. The knowledge bases for each of these expert systems is described below.

The knowledge-base for STRYPES is described by the decomposition knowledge and the constraints for recomposition. The decomposition knowledge is illustrated in Figure 2. The generation of alternative structural system types and materials is decomposed into the lateral and gravity load resisting systems. For the lateral system, a selection of alternative 3D systems and 2D systems in each direction are combined. The 3D systems are selected from 2D orthogonal systems and a 3D core system.The 2D systems are selected from rigid frames, braced frames and shear walls. For the gravity system, a selection of alternative 2D-horizontal systems and support conditions are combined. For example, a possible gravity system is a reinforced concrete slab supported on 4 edges without intermediate floor beams. Another possible system is a steel deck supported on two edges with intermediate floor beams.

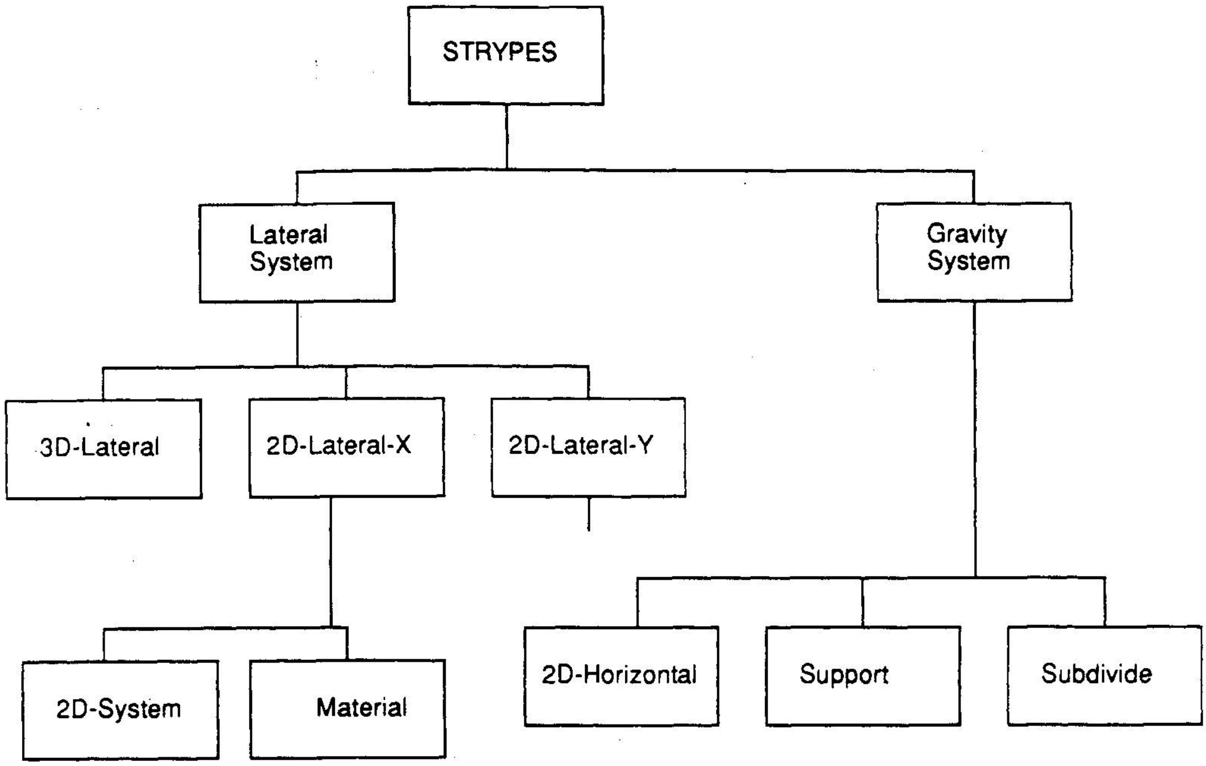


Figure 2: STRYPES Decomposition Knowledge

```
Gravity-System

2D-Horizontal   one-of   ( concrete steel-deck
                                panels waffle )
Support   one-of   ( 0-edges 2-X-edges 2-Y-edges
                                4-edges )
SubDivide   one-of   ( none X-direction Y-direction )
```

Figure 3: Gravity System in STRYPES

An example of a system definition in STRYPES is illustrated in Figure 3. The system represents the Gravity-System node in the decomposition tree. The alternative gravity systems are determined by combining selections from different 2D horizontal types and the number of edges supported and the decision to subdivide in one direction. The alternatives formed depend on the constraints and the design context. The use of a particular 2D horizontal type may depend on the lateral system and on the span of the structural grid. These constraints are generalized and stored in the knowledge-base.

The constraints on recomposition in STRYPES eliminate infeasible alternatives to reduce the number of solutions considered. Some constraints are based on design heuristics, eliminating alternatives that an experienced engineer would not consider. For example:

IF
lateral-system/3D-lateral = orthogoanl-2D
2D-lateral-system/2D-system = shear-wall
stories > 35
THEN not feasible.

This constraint eliminates the use of 2D shear wall systems for buildings with more than 35 stories. Other constraints eliminate unusual combinations of materials and systems. For example:

IF
2D-lateral-system/2D-system = shear-wall
2D-lateral-system/Material = steel.
THEN not feasible.

This constraint eliminates shear walls made entirely of steel.

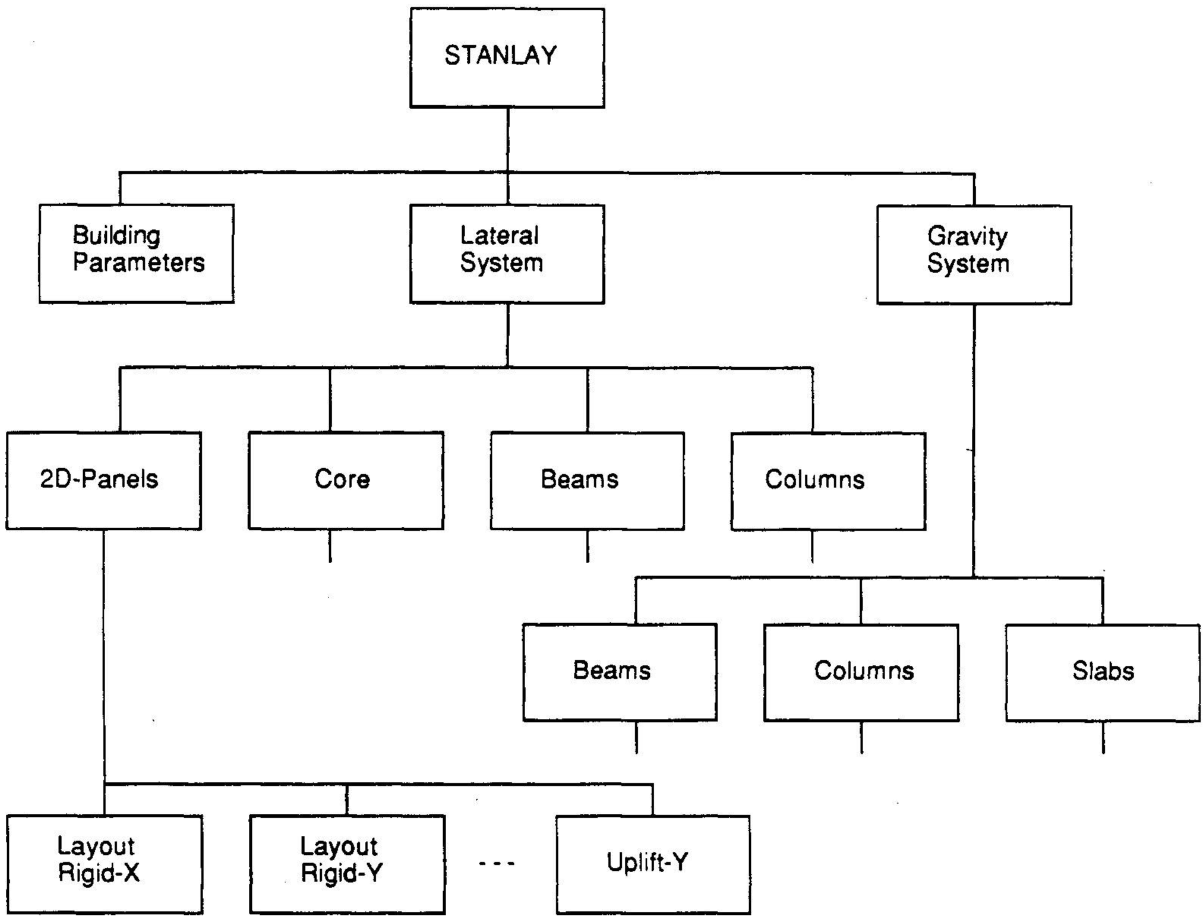


Figure 4: STANLAY Decomposition Knowledge

The decomposition knowledge in STANLAY is illustrated in Figure 4. The layout and load distribution is decomposed into three major decision groups: building parameters, lateral system, and gravity system. The building parameters system calculates and infers additional information about the building given the input conditions. The lateral system is considered by system and component type. The 2D-Panels system places the appropriate systems on the structural grid and distributes the lateral load to each panel. The 2D-Panels system generates alternative placement schemes. The core system locates the walls around the service shaft and determines the lateral load acting on the core. The beams and columns systems distribute the loads to each of the components using approximate analysis techniques. The gravity system, similar to the lateral system, distributes the gravity loads to the components using approximations.

```
2D-Panels

layout-rigid-X  one-of  (edges edges+1  . . . )
layout-rigid-Y  one-of  (edges edges+1  . . . )
- - -
Mover-X  function . . .
Mover-Y  function . . .
Uplift-X  function . . .
Uplift-Y  function . . .

Planning Rules:

IF  (2D-Lateral-  X = rigid-frame)
AND  (2D-Lateral-Y = rigid-frame)
THEN  (layout-rigid-X layout-rigid-Y  . . .)

IF  (2D-Lateral-X = braced-frame)
AND  (2D-Lateral-Y = rigid-frame)
AND  (TotalLength Y-Bays) > (TotalLength X-Bays)
THEN  (layout-braced-X layout-rigid-Y  . . .)
 - - -
```

Figure 5: 2D-Panels System in STANLAY

An example of a system definition in STANLAY is illustrated in Figure 5. The system represents the 2D-Panels node in the decomposition tree. The attributes of the 2D-Panels system include layout information and load information. The layout attributes are selected and ordered by the planning rules. The load attributes, i.e. overturning moment in each direction (M_{over}) and uplift forces, are computed by Lisp functions. The layout attributes have values that represent alternative placement schemes, e.g. edges indicates that the panels are placed on the edges of the building only, edges+1 places a panel in the center of the building in addition to the edges. The combination and use of the placement schemes are checked by constraints for consistency with building geometry and intended occupancy. Other constraints in STANLAY check the load attributes for each of the subsystems and components for appropriate magnitudes.

5. CONCLUSION

EDESYN provides a formalism for synthesis of structural systems that facilitates the incremental development of a knowledge-base for preliminary structural design. The expression of design knowledge as systems with attributes, planning rules, constraints, and evaluation criteria is easy to work with and expand. The development of STRYPES, STANLAY, and other expert systems using EDESYN has led to a better understanding of the knowledge required for preliminary design and the representations needed for expressing this knowledge in an expert system.

REFERENCES

1. Cowan, H.J. and Wilson, F., Structural Systems, Van Nostrand Reinhold, 1981.

2. Fraser, D.J., Conceptual Design and Preliminary Design of Structures, Pitman, 1981.

3. Holgate, A., The Art in Structural Design, Oxford University Press, 1986.

4. Lin, T.Y. and Stotesbury, S.D., Structural Concepts and Systems For Architects and Engineers, John Wiley and Sons, 1981.

5. Maher, M.L., HI-RISE: A Knowledge-Based Expert System For The Preliminary Structural Design of High Rise Buildings, PhD Thesis, Department of Civil Engineering, Carnegie Mellon University, 1984.

Leere Seite
Blank page
Page vide

Expert System Applications in Construction Materials Technology

Systèmes experts en technologie des matériaux de construction

Expertensysteme in der Baustoff-Technologie

Karl-Johan SERÉN
Research Scientist
Technical Research Centre
Espoo, Finland

Karl-Johan Serén, born 1958, received his MScTech degree in mechanical engineering at Helsinki University of Technology. For five years he has been involved in research projects concerning automation in the concrete industry, with special interest in application of advanced electronic data processing techniques, such as knowledge-based systems.

SUMMARY
This paper deals with the application of expert system technology in construction materials technology. Two expert system applications are described. The problem domain of the first system is diagnosis of brickwall damage. The application area of the second system is repair of concrete balconies. Both systems run on microcomputers. Moreover, other expert system applications dealing with construction materials developed or under development in Finland are presented in brief.

RESUME
Le présent rapport explique l'application de la technologie des systèmes experts à la technologie des matèriaux de construction. Deux applications sont décrites, l'une destinée à la diagnose des défauts des murs en briques, l'autre à la réparation des balcons en béton. Les deux systèmes marchent sur micro-ordinateurs. En conclusion, d'autres applications du système expert développées ou en voie de développement en Finlande dans la technologie des matériaux de construction sont discutées.

ZUSAMMENFASSUNG
Dieser Vortrag befasst sich mit der Anwendung der Expertensystemtechnologie auf die Baustofftechnologie. Zwei Anwendungen werden beschrieben. Bei der ersten Anwendung handelt es sich um die Diagnose der Schäden an Ziegelwänden. Die zweite Anwendung behandelt die Reparatur von Betonbalkonen. Beide Systeme laufen auf Mikrocomputern. Abschliessend gibt es eine kurze Zusammenfassung anderer Expertensystemanwendungen, die in Finnland für die Baustofftechnologie entwickelt wurden oder in Entwicklung sind.

1. INTRODUCTION

1.1 General

Expert systems have a great application potential in the field of construction materials technology. Material related decision-making spans over the whole construction process, from design to actual usage of the structure, and a considerable amount of economical assets are bound to these decisions. The decision processes related to construction materials, for instance selection of materials, diagnosis of material-based structural damages and selection of correct working and repair procedures, involve a lot of heuristics and judgment knowledge of conceptual nature. These processes are generally impossible to model using traditional algorithmic computer programs. Therefore it is quite natural that the expert systems technology has been gaining a lot of interest lately in this context. It may be noticed that one of the first generally known applications of expert or knowledge-based systems in civil engineering, SPERIL-I [3], was from this domain of application (assessment of structural damages).

One aspect worth considering is that the conceptual decision-making process, which can be modelled by rule-based inferencing in expert systems, is often linked to other types of data processing. For instance data bases can be used for convenient storing and updating of basic material data. Furthermore, some kind of numerical processing as well as graphical presentation of data is often needed. This makes the integration capabilities of expert system development software a key issue. In this respect the future of expert systems looks promising, because the newest microcomputer-based expert system shells have quite good interfacing capabilities (see for instance [6]).

1.2 Civil engineering expert system applications in Finland

Research work concerning expert systems in civil engineering and construction has been carried out in the Technical Research Centre of Finland (VTT) from mid-1984 onwards. Apart from some surveys and feasibility studies, for instance concerning expert systems in the concrete industry [7], the research activities have concentrated on actively developing small and mid-sized expert system prototypes. General information about these systems and experiences gained from developing them can be found in [4, 8]. Some of the prototypes are related to construction materials technology: an expert system for the selection of ready mix concrete [9], an expert system for repainting of wooden facades [4], an expert system for diagnosing brickwall damages [5] and an expert system for repair of concrete balconies [7]. The two last ones of these systems will be presented in this paper.

2. CASE 1: BRICKWALL DAMAGE EXPERT SYSTEM

2.1 General background

The *Brickwall Damage Expert* is a typical diagnostic expert system intended for analyzing damage to brickwalls. The system was developed by one of VTT's research scientists, Kalle Kähkönen, during a two month visit to the Loughborough University of Technology in England. The development work was carried out under the supervision of Dr. Allwood at the Department of Civil Engineering which provided both know-how, hard- and software. The source of the knowledge incorporated into the system is a book written by Elbridge [2], which deals with common defects in British buildings and is especially intended for practical application and people concerned with building maintenance. The Brickwall Damage Expert is quite restricted and it can be considered as a demonstration prototype. The system is described in detail in [5].

2.2 Development software

The Brickwall Damage Expert was developed using the *Savoir* expert system shell. Savoir is basically a bayesian inference network type of expert system shell. The three main components of the knowledge representation language are: a) questions, referred to as "Questions", b) rules, referred to as "Variables" and c) demons, referred to as "Actions". In addition the shell has the capability to interface to external functions by a module called the "trap handler". Savoir is equipped to deal with both classical logic and uncertain reasoning. The main probability operator provided is the bayesian inference rule to which is added some fuzzy logic operators. There is no default control strategy. Instead the knowledge engineer can build a control procedure to meet the specific requirements of a domain by defining the "Actions". The shell supports both backward- and forward-chaining. The end-user interface can be quite freely constructed by the knowledge engineer. The Savoir shell is available for a variety of computers including IBM PC, DEC PDP-11 and IBM mainframes. Further information about Savoir can be found in [1].

2.3 Problem domain issues

During preliminary discussions it became clear that the brickwall damage diagnosis process should be focused on the causes of cracks. The secondary goals should aim to provide guidelines on remedial work. However, the finding of the main goals (causes of cracks) is quite a demanding task, including probabilities and thus the causes became the most important factor in developing this expert system. The repair guidelines are linked to the particular causes and the feature of presenting corresponding repair instructions can easily be added to the system later.

The knowledge was analyzed into a tree structure by the use of a computer aided drawing tool starting from possible causes of cracking and linking every possible symptom leading to each specific cause. Twelve different causes of brickwall damage were found from the source book and the seven which seemed to be the most important were selected to be used as goals of the system (fig. 1).

2.4 System structure

The knowledge base consists of nine separate knowledge files from which the system is built up during compilation (fig. 1). One of these files is called the main knowledge file into which other knowledge files are included during compilation. Other tasks of this file are: a) starting of the consultation, b) leading of the inference process in an appropriate direction and c) displaying of the conclusions. One file called "the initial question file" contains the initial questions to be asked at the beginning of the consultation. Other files include the knowledge related to the calculations of the probability of a certain cause. As a whole the knowledge base consists of 45 questions, 33 variables and 19 actions taking up 36 KBytes of memory in source code form and only 2.5 KBytes of memory in compiled form. The system runs on an ordinary IBM PC -type of microcomputer.

The user interface of the Brickwall Damage Expert is based on the standard user interface provided by the Savoir shell. It consists of overlapping windows each of which has a pre-defined function, size, colour and place on the screen when the window is displayed. The system includes 36 pre-defined questions for defining the symptoms of brickwall damage. During the asking of questions the system calculates the probabilities of all relevant causes. Once the relevant questions are asked a display is shown with the most probable cause of the damage and the probabilities of the other causes (fig. 2).

3. CASE 2: EXPERT SYSTEM FOR REPAIR OF CONCRETE BALCONIES

3.1 General background

The expert system for repair of concrete balconies (*Concrete Balcony Repair ES* for short) is another quite typical diagnostic expert system based on classification of knowledge. The system was developed during a preliminary feasibility study in the Concrete and Silicate Laboratory of VTT and at present it is a quite rough demonstration prototype. The system is intended to be used as an aid in preparing the repair planning documents, but it may also be used by the contractor to aid in preparing the working plans. The knowledge for the system was provided by a research scientist specialized in repair and rehabilitation of concrete structures.

3.2 Development software

The Concrete Balcony Repair ES was developed using the *Xi Plus* expert system shell. Xi Plus is a rule-based shell with some interesting features [10]. In addition to ordinary backward-chaining rules (If-Then rules) forward-chaining can be used. Furthermore, a wide variety of inference control statements and rules can be used, for instance so called "Demons" (When-then rules) which can be applied as a kind of meta-level control. The knowledge representation includes the possibility to assign class membership to an identifier by the use of the "is a" relation and to use variables in the rule statements as well as attributes in the form of "<attribute> of <identifier>". Moreover, interfacing capabilities to external programs, spreadsheet data files and graphics (GEM) are provided. Knowledge bases can be linked together in a divided structure to provide for a more economical memory usage and to make the updating of the knowledge easier. The Xi Plus shell runs on IBM PC and compatible microcomputers.

3.3 Problem domain issues

When analyzing the domain knowledge two general classes of damages in concrete balconies could be identified: a) surface damages and b) cracking. Each of these two classes has a set of possible causes for damages with corresponding symptoms or properties in a similar manner to the Brickwall Damage Expert. The repair methods can in general be determined by the causes of damage with additional information concerning the level of damage.

Accordingly, the problem domain could be divided into two general classes of damages, with two contexts each: a) diagnosis of the damage and b) determination of damage level and corresponding repair method. Each of the general classes has five possible damage causes. 18 different types of repair methods could be identified. It may be noticed that no probabilities were used and that the damage causes are stated explicitly. The use of probabilities is often questionable, because they usually represent a subjective view of one expert and some experts find it hard to define any numerical probability range for different cases.

3.4 System structure

In accordance with the analysis of the problem domain the knowledge base was divided into three smaller units: a) a small general knowledge base containing only two rules for loading appropriate sub-knowledge base according to user selection, b) a sub-knowledge base for surface damages containing 18 rules (fig. 3) and c) a sub-knowledge base for cracking containing 10 rules (fig. 4). The two rules in the general knowledge base are so called "Demons" mentioned in sub-chapter 3.2. Each sub-knowledge base has two contexts the execution of which is controlled by control statements in one rule each: a) diagnosis of the damage and b) determination of the repair method.

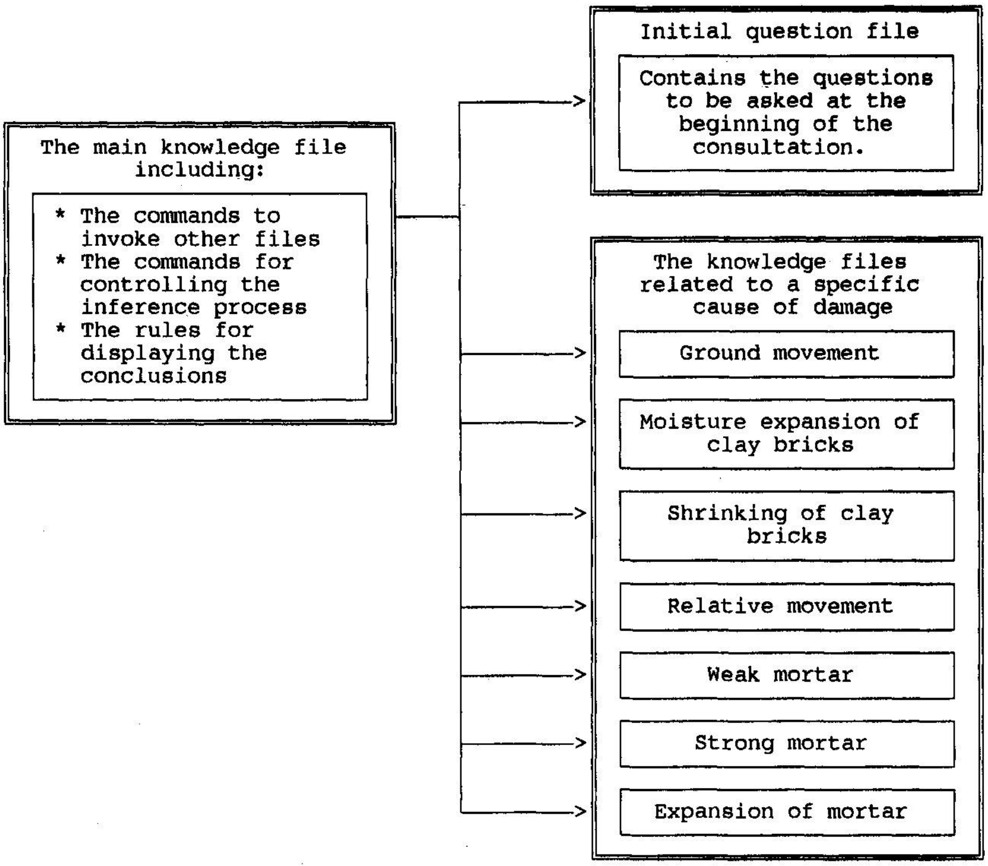


Fig. 1. The structure of the knowledge base of the Brickwall Damage Expert [5].

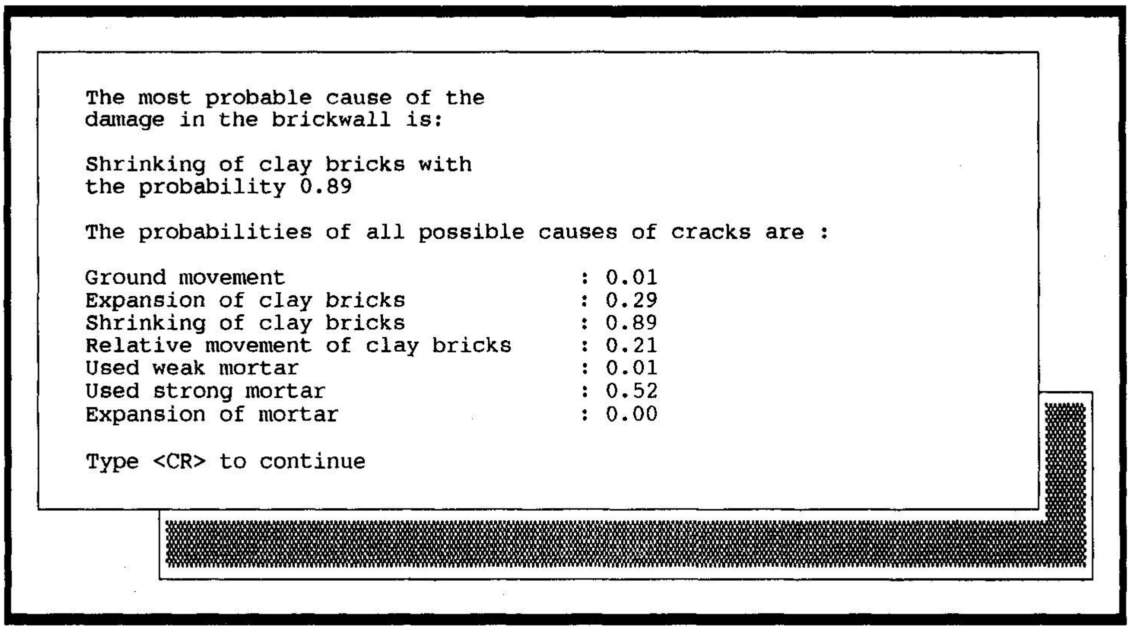


Fig. 2. The concluding display of the Brickwall Damage Expert. Example from [5].

The user interface of the Concrete Balcony Repair ES is based on the standard user interface of Xi Plus which is divided into windows. The questions presented by the system are automatically generated by the Xi Plus inference engine in the form of menu type queries based upon question definition statements in the beginning of each knowledge base. The results are displayed to the screen in the form of normal text output.

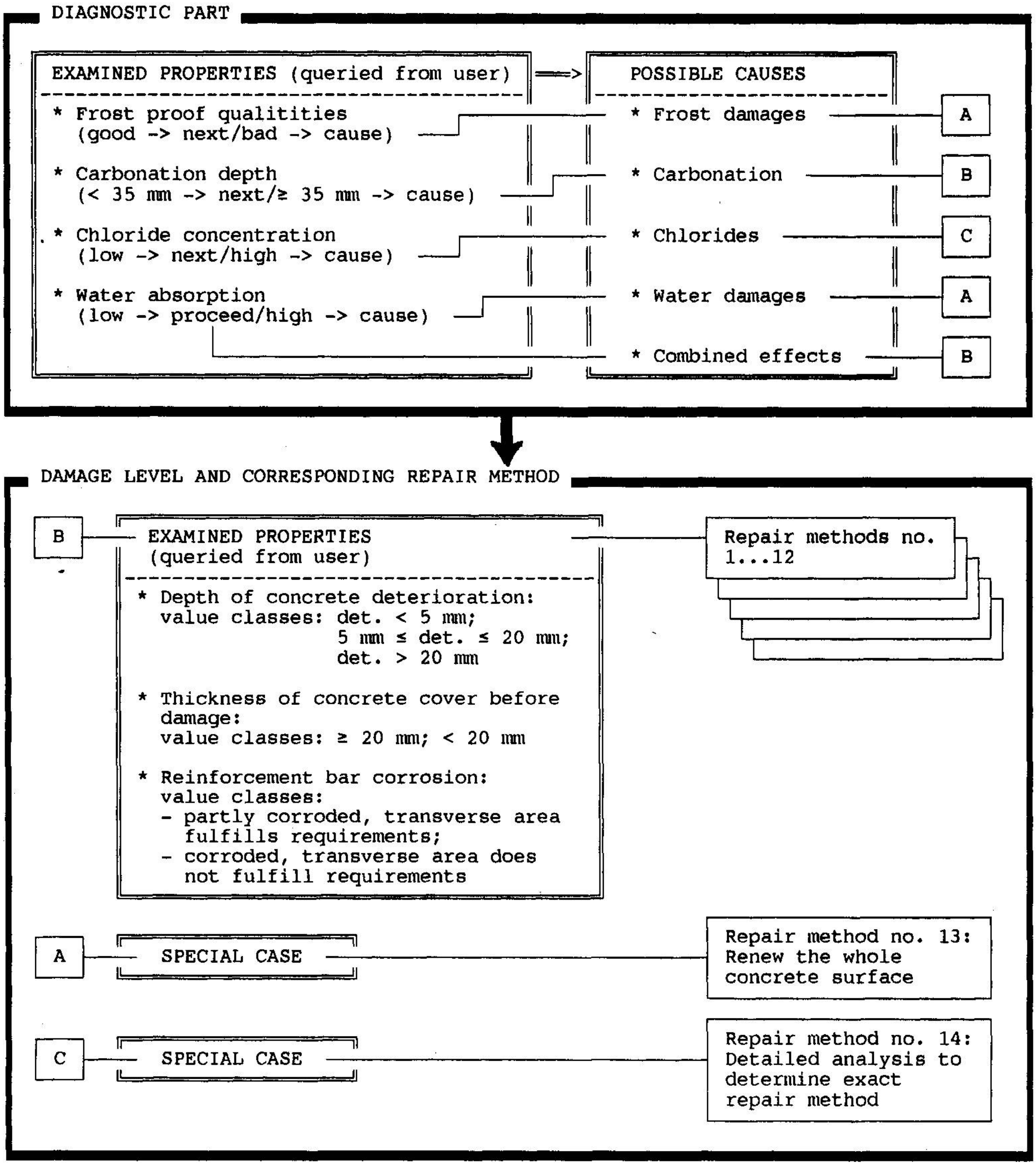


Fig. 3. Structure and function of sub-knowledge base for surface damages (Concrete Balcony Repair ES).

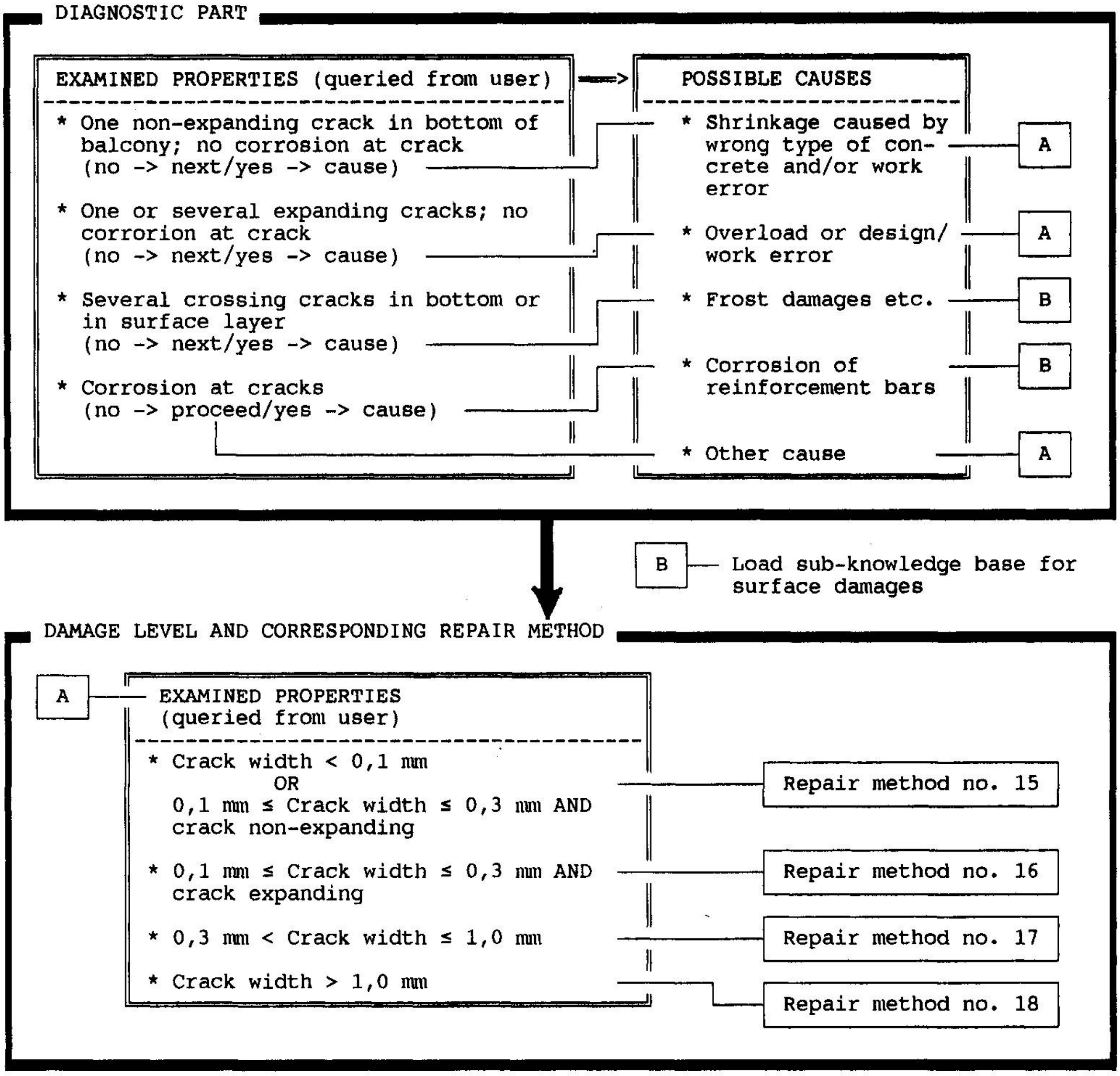


Fig. 4. Structure and function of sub-knowledge base for cracking (Concrete Balcony Repair ES).

4. ON-GOING RESEARCH WORK CONCERNING EXPERT SYSTEM APPLICATIONS IN CONSTRUCTION MATERIALS TECHNOLOGY IN FINLAND

Several research projects concerning civil engineering and construction industry expert system applications are going on or in preparation in VTT. Although the main focus at the moment is on application of knowledge-based systems in construction management, some project are related to construction materials.

In connection to a major three year research program called "Information and Automation Systems in Construction" the use of knowledge engineering approaches in construction materials technology will be studied. The feasibility of a framework interfaced to a product model database for selecting materials is under investigation. As a pilot-study the framework will be applied to renovation of facade cladding and surface coating (paint, plaster, clinker tiles etc.). This pilot-study will include condition assessment of facade structures. Some preliminary studies have shown that a generic framework is difficult to extract because of the very differing needs of different potential user groups (materials researchers, designers, contractors etc.).

Lately, however, some promising results have been attained by applying an object-oriented approach to the knowledge analyzing, by which it seems possible to develop a conceptual framework for the domain. The implementation environment will be the Hypercard hypermedia running on Apple Macintosh II with a software link to the Nexpert Object expert system shell.

A project concerning the development of a methodology for damage assessment of concrete structures is in preparation in the Concrete and Silicate Laboratory of VTT. In connection to this project an expert system aiding the assessment procedure will be developed.

5. CONCLUSIONS

The experiences show that microcomputer-based expert system applications are well suited for narrow problem domains which incorporate classification type of knowledge structures. However, a lot of work remains to be done before this type of expert systems can be taken into everyday use. The vast majority of the known expert systems of this kind have been developed in universities or research institutes and to the authors knowledge none of these have been taken into practical production use. The two example applications described in this paper are typical in this sense. The two systems have not gone through any critical evaluation and they could hardly be used in practice due to their roughness and restricted knowledge bases.

In the future more attention should be paid to the actual needs of different potential end-users. The expert system software technology itself can be considered mature enough for practical applications. The main difficulty is related to the difficulty to model the actual deeper knowledge of the decision processes involved in this kind of applications. This is a key factor in larger system applications with real practical viability. For the end-user it makes no difference what kind of computer techniques a system is based on, as long as the system is easy to use and behaves in a sensible manner. The implication of this is that a lot of effort should be directed to develop systems with complete and robust knowledge bases and good user interfaces, which could tailored for different user groups.

ACKNOWLEDGEMENTS

The main part of the work described in this paper was funded by the Technical Research Centre of Finland (VTT). Several persons have contributed to the results presented. The author would especially want to thank research scientist Kalle Kähkönen (Laboratory of Urban Planning and Building Design of VTT) who developed the Brickwall Damage Expert; senior research scientist Kalle Tanskanen (Concrete and Silicate Laboratory of VTT) who provided the knowledge for the Concrete Balcony Repair ES; and techn. stud. Jukka Kuisma who assisted in developing the Concrete Balcony Repair ES. All uses of soft- and hardware trademarks are also acknowledged.

REFERENCES

1. ALLWOOD, R. J. et al. Evaluation of expert system shells for construction industry applications. Loughborough 1985, Loughborough University of Technology, Department of Civil Engineering. 114 pp. + app. 31 pp.

2. ELBRIDGE, H. J. Common defects in buildings. London 1976, Department of the environment, Property Service Agency, Her Majesty's Stationery Office. 486 pp.

3. ISHIZUKA, M., FU, K. S. & YAO, J. T. P. Computer-based systems for the assessment of structural damage. Proceedings of the IABSE Workshop on Informatics in Structural Engineering. Bergamo, 6 - 8 Oct. 1982. Zürich 1983, International Association for Bridge and Structural Engineering, IABSE Reports, Volume 40. Pp. 89 - 98.

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

5. KÄHKÖNEN, K. Diagnostic expert systems development for construction. A case study. Espoo 1987, Technical Research Centre of Finland, Research Notes 705. 47 p. + app. 16 pp.

6. PEDERSEN, K. Connecting expert systems and conventional environments. AI Expert 3(1988)5, pp. 26 - 35.

7. SERÉN, K.-J. Expert systems for the manufacturing process in precast concrete building component factories and concrete construction. Fourth Int. Symp. on Robotics and Artificial Intelligence in Building Construction. Haifa, 22 - 25 June 1987. Haifa 1987, Building Research Station - Technion I.I.T. Vol. II. Pp. 771 - 787.

8. SERÉN, K.-J. Expert systems in construction: State of the art in Finland. First French-Finnish Seminar on Information Technology in Construction. Valbonne - Sophia Antipolis, 4 - 5 Oct. 1988. Valbonne - Sophia Antipolis 1988, Centre Scientifique at Technique du Bâtiment (CSTB), ES/88-249. 13 pp.

9. SERÉN, K.-J. An expert system for choosing the type of ready mix concrete. In: Nordic Concrete Research, Publication no. 7. Oslo 1988, The Nordic Concrete Federation. Pp. 259 - 272.

10. Xi Plus - The Expert System. User manual. Slough 1986, Expertech Ltd.

Leere Seite
Blank page
Page vide

Knowledge-Based Design of Steel Portal Frames for Agricultural Buildings

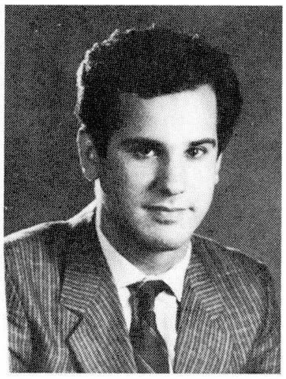
Cadres métalliques pour bâtiments agricoles dimensionnés à l'aide d'une base de connaissance

Wissensbasierende Projektierung von Stahlrahmen für landwirtschaftliche Gebäude

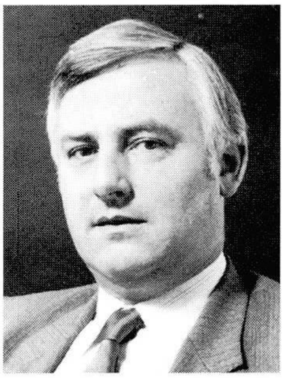
João BENTO
Civil Engineer
Technical Univ. of Lisbon
Lisbon, Portugal

Bruno FEIJÓ
Aeronautics Engineer
PUC/RJ
Rio de Janeiro, Brazil

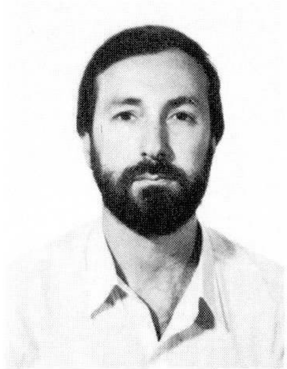
Patrick J. DOWLING
Prof. of Civil Engineering
Imperial College
London, UK



João Bento received his Engineering degree 1983 and MSc in Structural Engineering, 1987 at the Technical University of Lisbon were he is Lecturer. His research interests are steel/composite structures and, more recently, expert systems for structural design. He is also Visiting Lecturer in the Expert Systems Lab Imperial College.



Bruno Fejió received his Engineering degree at ITA, São Paulo, 1975 and his PhD at Imperial College, 1988. For many years he worked with computers in engineering design. He is now Lecturer in Computer Science in PUC/RJ. He is also Visiting Lecturer in the Expert Systems Lab. Imperial College.



Patrick Dowling is a member of Britain's National Academy of Engineering, the Fellowship of Engineering. He is Chairman of the Drafting Panel of Eurocode 3 and Editor of the Journal of Construction Steel Research and Director of the Consulting firm Chapman & Dowling Associates Ltd which drafted the Brazilian Steel Bridge Code.

SUMMARY
A simple conceptual model of the structural design process has been proposed by Feijó (1988). This paper deals with the improvement and implementation of that model. A first implementation, however, demands a domain of application that is both simple and specific. These criteria are certainly met in the design of steel portal frames for agricultural buildings, especially through the rather heuristic style of design which has evolved. A new routine for knowledge acquisition in design problems is also proposed and some aspects of the implementation of the model are discussed.

RESUME
Les bases d'un modèle conceptuel d'une procédure de dimensionnement des structures ont été proposées par Feijó (1988). Cet article concerne le développement et la mise en oeuvre de ce modèle. Une première mise en ouvre requiert un domaine d'application à la fois simple et spécifique. Tel est le cas du calcul des cadres métalliques pour bâtiments agricoles, grâce surtout aux caractéristiques heuristiques de ces projets. Une nouvelle méthode pour l'aquisition de la connaissance est proposée et quelques aspects plus intéressants de la mise en oeuvre du modèle sont discutés.

ZUSAMMENFASSUNG
Feijó (1988) hat ein einfaches Begriffsmodell für die Berechnung von Strukturen vorgeschlagen. Dieser Artikel behandelt die Entwicklung und Verwirklichung jenes Modells.
Diese erfordert jedoch ein gleichzeitig einfaches wie auch spezifisches Anwendungsgebiet.
Die von Grund aus empirischen Stahlrahmen für landwirtschaftliche Gebäude entsprechen sicherlich diesen Kriterien. Eine neue Methode für dem Wissensgewinn bei der Berechnung wird vorgeschlagen und einige Aspekte der Verwirklichung des Modells werden kommentiert.

1. INTRODUCTION

This paper concerns the development of a model of the design process drawn on an idea proposed by Feijó [1]. The domain of steel portal frames for agricultural buildings is adopted because it is both simple and specific.

First, a definition of design is presented followed by the assumption that most of the design activity is computation. Secondly, the model SAE of the design process is presented. Thirdly, some aspects of the implementation of this model are discussed. Finally, some results of the investigation into the domain of application and of the knowledge elicitation process are presented.

2. A DEFINITION OF DESIGN

The definition of design presented in this paper is an adaptation of the ideas proposed by Vinod Goel and Peter Pirolli [2]. Design is then defined by

$$D(Space, Inv, Struct)$$

where Space is a design problem space (2.1), Inv is a set of invariants of design (2.2) and Struct is a structure of the design problem space (2.3).

2.1 Design problem space

Research in design methodology has revealed that designers create a central case (a prototypical case) C_0 in the early stages of the process. Moreover, the designer is faithful to the central case during the whole process. In this context, design is an evolutionary process that refines the central case towards the artifact specification S(E,A,R), i.e.

$$I \to C_0 \to C_1 \to \cdots \to S(E,A,R)$$

where I = input specifications, E = entities, A = attributes of the entities and R = relationships between entities.

The sequence of states $\{C_i\}$ represents one of the many routes through a design problem space. This draws on the concept of plans of Newell and Simon [3], who pioneered the computational modelling of mental processes.

The states C_i are unpredictable but motivated variations of the central case C_0. This is equivalent to the radial category with prototype effects proposed by Goel and Pirolli [2]. Also, this evolutionary aspect of the process can be found in other models of design [4].

The sequence of states C_i may go into the design problem space, get lost and never emerge with the artifact specification. Constraints are needed to keep the search (for a possible route) to a manageable size and to help navigation in the problem space. In other words, design should be constraints driven.

The design problem space is supposed to have a set of design characteristics which explain movements in the problem space and support the evolution of the states C_i. A simplified version of this set, adapted from Goel and Pirolli [2], is as follows:

d1: Extensive Problem Structuring (i.e. finding missing information). The following steps are used by designers: Studying the design briefly; Soliciting information and clarification; Applying constraints: legislative (e.g. design codes), technical, pragmatic (e.g. money); Applying personal knowledge; Negotiating constraints (to fit personal ideas);

d2: Extensive Performance Modelling (simulation also included);

d3: Personalized Evaluation Functions (stopping rules also included);

d4: General Evaluation (i.e. a generated or focused component is continuously evaluated in three contexts: local context; current context of the complete artifact; future context - i.e. projecting the complete artifact in its final stage);

d5: Dynamic Commitments (i.e. making, negotiating and propagating commitments);

d6: Solution Decomposition (moreover, modules are not isolated, but interconnected at a function level);

d7: Abstraction Hierarchies (i.e. working with modules and entities at various levels of detail. Descending too soon or not descending at all this hierarchy is a common mistake of novice designers.);

d8: Use of Symbol Systems (e.g. bubble diagrams, rough sketches and natural language);

Design characteristics represent cognitive needs which should be satisfied by any CAD system. In this context, hypertext tools might be offered to help problem structuring (d1), object oriented languages might help the implementation of d4, d5, d6 and d7, and intelligent user interface techniques might help the use of symbol systems (d8).

2.2 Invariants of design

Goel and Pirolli [2] propose the differentiation between design problem solving from non design problem solving by identifying major invariants in the design process. The authors adopt the following list of invariants:

inv1: Large and Complex problems;

inv2: Input as goals and intentions (statements) and output as an artifact specification S(E,A,R);

inv3: Temporal separation of design phase and delivery of the artifact (usually long);

inv4: Delayed or limited feedback from the world (in the sense that, after delivery, reactions from the world can hardly guide the designer in the current project – only in the next similar project);

inv5: Independent functioning of the artifact, i.e. the designer cannot be there to explain its significance or perform its function (this is the case of a bridge designer and the project of a bridge);

inv6: Costs (penalties or benefits) associated with every action;

inv7: No right/wrong answers, only better/worse ones;

inv8: Many degrees of freedom in design problem statements (i.e. many open questions).

In this context, music composition is a better example of design activity than spontaneous conversation. In the later, there is no separation between design phase and delivery (they are simultaneous), the problem solver actually constructs the artifact rather than specifying it, there is no delay feedback and, finally, costs are not associated with actions.

2.3 Structure of design problem space

The structure of the design space is a relation that assigns invariants to the characteristics of the design problem space, i.e.

$$\begin{aligned} \text{inv8} &\rightarrow \text{d1} \\ (\text{inv3, inv4, inv5, inv6}) &\rightarrow \text{d2} \\ \text{inv7} &\rightarrow \text{d3} \\ \text{inv3} &\rightarrow \text{d4} \\ (\text{inv3, inv2}) &\rightarrow \text{d5} \\ \text{inv1} &\rightarrow \text{d6} \\ (\text{inv1, inv2}) &\rightarrow \text{d7} \\ \text{inv1} &\rightarrow \text{d8} \end{aligned}$$

The association of characteristics with particular movements in the problem space and the interconnection between characteristics are not considered part of the definition of structure. These are specific aspects of a model of the design process.

3. DESIGN AS COMPUTATION

Design is a cognitive process involving deduction, induction, experience, creation and intuition. These are complex ways of reasoning that are difficult to treat with a mathematical formalism. However, the authors believe that some problems of cognition can be formulated with some precision in terms of the mathematical theory of computability [5]. On the other hand, they recognize the limitations of this approach because there are aspects of mental life that cannot be represented mathematically. In this particular, the reader is addressed to the classic works of the critics of artificial intelligence, such as Hubert Dreyfus [6], John Searle [7] and Terry Winograd [8].

There are various computational methods for considering the ways of reasoning mentioned above. Formal proofs in first-order logic capture an important aspect of deduction. Also some forms of induction (where induction is a systematic way of reasoning that increases the given information) can be easily adopted, e.g.

High_rise_building(a)
Steel_framework(a)
High_rise_building(b)
Steel_framework(b)
...

might induce

For all x, High_rise_building(x) $\Rightarrow$ Steel_framework(x)

Other forms of induction are also possible [9][10]. Furthermore, learning techniques can handle some aspects of design experience. However, very few methods can be proposed for creation (e.g. analogy across different domains). Creation and intuition are still in the area of competence of human designers.

Artificial reasoning is associated with the problem of knowledge representation. As far as design is concerned, every characteristic of the problem space is associated with one or more types of knowledge (knowledge about design codes, personal knowledge, knowledge about evaluation, knowledge about abstractions, and others).

A taxonomy of personal knowledge is proposed by Goel and Pirolli [2]. They identify three kinds of personal knowledge: procedural knowledge (which cannot be made explicit - i.e. it is not open to introspection); abstract conceptual knowledge (principles, laws and heuristics which are situation specific knowledge; knowledge of patterns (forms or patterns directly evoked by a designer). Furthermore, Goel and Pirolli [2] claim that personal knowledge is organized in two levels: a general level and a domain specific level. The second represents a more organized and complete knowledge than the first and it is built on top of it. The authors believe that most of these types of knowledge can be satisfactorily represented by first-order logic and frames.

4. THE MODEL SAE

In the model SAE, a state C_i is represented by a set of propositions in first order logic, such as:

Tension_member(b1)
There exists x, supports(x,b1).

The evolution of the states C_i is carried out by a recursive process of three subprocesses *Synthesis - Analysis - Evaluation* which are themselves recursive processes. At least one process S-A-E is required to move from one state to another. Furthermore, these processes are organized in levels of abstraction. The central case is generated by a higher level process and detailing is a low level process.
In the case of structural engineering design, the following processes can be identified: Higher level processes (e.g., conception); Preliminary design; Structural analysis; Detailing. The model SAE is shown pictorially in Figure 1.

Design in the model SAE is recursively specified by the following sentences in first-order logic:

design (X, end).
design (X, Level)
  if **synthesis** (X, Level) and
  analysis (X, Level) and
  evaluation (X, Level, New-Level) and
  design (X, New-Level).

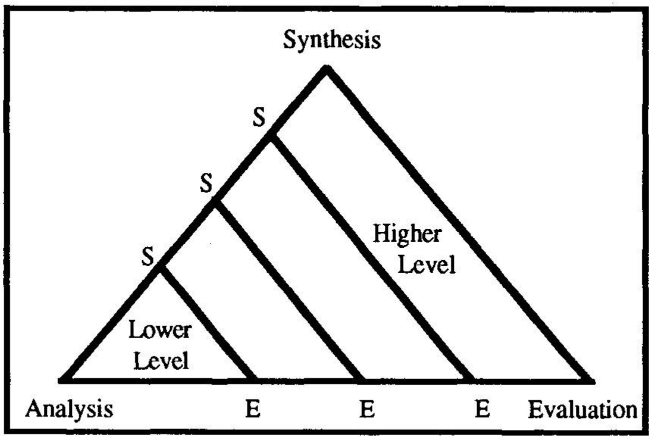


Figure 1

A subprocess of synthesis, analysis or evaluation represents a specific form of knowledge (i.e. how to synthesize, to analyze or to evaluate). In turn, this knowledge can evoke other subsets of knowledge associated with this type of subprocess. Furthermore, that specific form of knowledge can call a new design process for a new entity or module (at a specific level of abstraction).

Synthesis (the "so what ?" process in design) is the crucial problem because it involves creation. The degree of artificial intelligence of a CAD system is proportional to the degree of independence that each process of synthesis has from human intervention. Interactive computer graphics and intelligent user interface techniques can be understood as the link between the incomplete artificial synthesis (null in conventional CAD systems) and the external human synthesis (Fig. 2). The peculiarities of synthesis activities will be focused further on when the problems involved in knowledge elicitation in design problems will be discussed.

5. PRINCIPLES FOR AN ARCHITECTURE OF THE MODEL SAE

The proposed architecture starts adopting the framework of multi-expert systems. This framework consists of a set of Artificial Design Assistants (each assistant corresponding to a specific domain of expertise) and a strategy of cooperation. The assistants are related with subprocesses, specific sub-domains and levels in the model SAE. These assistants are supposed to provide design scripts, advices and answers to the user or another assistant. The objective of the cooperation is to reach an artifact specification S(E,A,R) with no inconsistency. In the present work, the strategy of cooperation is not presented.

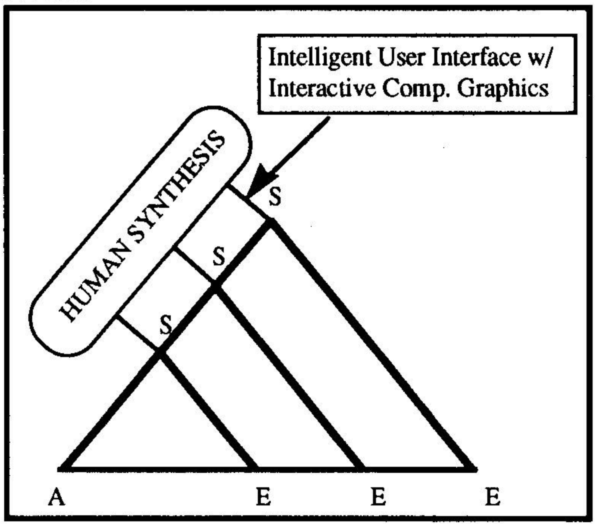


Figure 2

The proposed architecture is integrated with a set of conventional CAEngineering systems, such as CADraughting and Finite Element packages, as illustrated in Figure 3. In this particular, an effective interface between first-order logic and procedural programming should be developed [11].

The Database of Design Facts (DDF) represents the current state C_i and is classified into three areas as follows: Design Requirements (these facts represent a design request which includes hard constraints and basic assumptions); Design Options (these facts are soft constraints which represent a design script); Data (these facts are temporary data retrieved from the conventional database or generated by a subprocess). Synthesis processes may change the classification of any particular design fact.

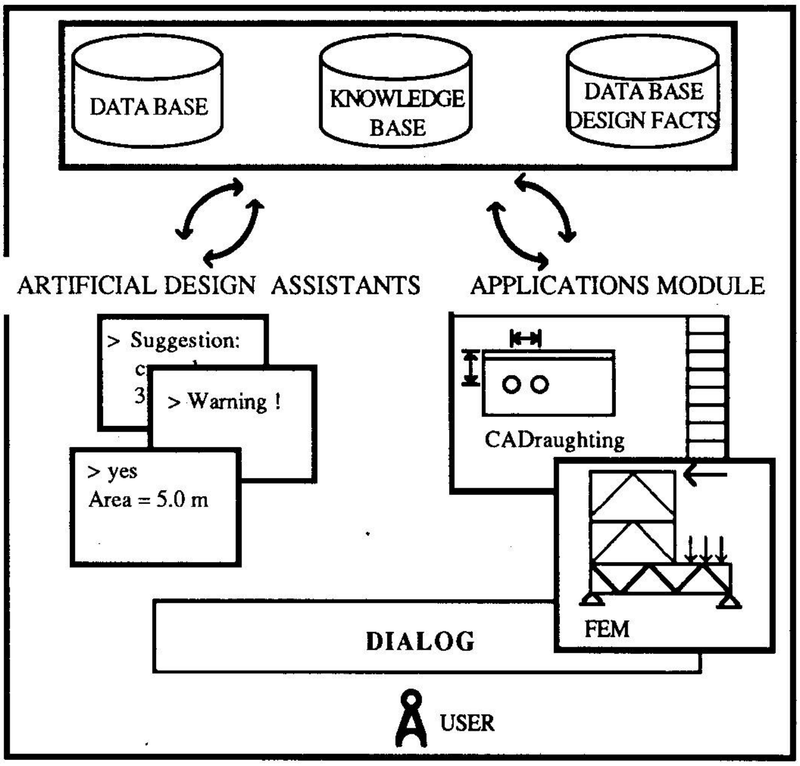


Figure 3

5.1 The role of facts

The evolution of the recursive process in the model SAE is closely aligned to the changes in the Database of Design Facts (DDF). Analysis processes can be triggered by changes in the DDF provided by synthesis processes. Evaluation processes can be triggered by new design facts provided by analysis processes (moreover, if any analysis process fails, the reason for failure will also be present in DDF).

Synthesis processes can be triggered by facts which evaluation processes identify as causing deviations.

This evolution can be exemplified by the following simple example, in which structural analysis is assumed to be the current design level:

> – Under a sub-process of analysis, a steel portal frame is beeing analyzed by means of the use of a FE package; the output of the FE analysis will generate new design facts, e.g. displacements; evaluation will then be triggered by the arrival of these new design facts; during evaluation the displacements may be found to have exceeded the displacement limits defined in the knowledge base; this occurrence will then trigger a synthesis process to allow the system to make a decision about the deviation identified; assuming the synthesis process provided a solution for this problem by redefining the depth of the haunches in the beams of the portal frame, then a change in the DDF would occur (e.g., *hauch_depth(frame, 1.5)* would become *hauch_depth(frame, 2.3)*); this would trigger a new sub-process of analysis...

In the present model facts are, as described, of great importance in driving the evolution during the process of design, towards the delivery of the design itself. This draws the authors' attention for the way of dealing with facts at implementation (6.2) and at knowledge acquisition (7.3).

6. IMPLEMENTATION OF THE MODEL SAE

6.1 Domain of application

Extracting, structuring and formally representing experts personal knowledge is extremely difficult. Furthermore, the definition which is being presented for the representation of the structure of the design problem space is formal and theoretical.

It is mainly due to this reasons that a specific and simple domain of application had to be selected for a first implementation of the model SAE. Steel portal frames for agricultural buildings, being simple but peculiar structures which are designed rather heuristically, seemed to be an attractive solution meeting those requirements.

6.2 Programming requirements

From a programming point of view, the more relevant aspects under consideration in the implementation of the proposed model are related to the need for an integrated use of several cooperating expert systems (Artificial Design Assistants – ADAs), conventional procedural applications (Applications Module – AM) and databases.

First order logic and frames are the formalisms upon which each of the ADAs is being built. It is, hence, a logic programming style which is being used to build them. It has been stated that the strategy for cooperation between the ADAs will not be presented in this article.

Concerning the procedural applications involved in the system, the main problem being tackled is their effective integration with the logic systems (the ADAs) both in terms of its mutual invocation and in terms of data interchange. A number of effective localized solutions for these problems have been proposed and already implemented [11], and the need for the development of a global logic $\leftrightarrow$ procedural programming interface was identified.

As concerns the presence of databases providing support to the applications module (CAD, FEM, etc.) and the ADAs, special attention has to be taken to the possible ability for logic systems to handle data – namely ground clauses (facts) – stored in secondary memory (e.g. databases) in the line of the contributions from Cotta [12]. This may be found to be of great importance for the improvement of the efficiency of the overall system; namely, in terms of response time for inferences involving different expert systems and procedural programs as well as in terms of the possible physical size of the knowledge bases.

7. KNOWLEDGE ACQUISITION

In spite of the simplicity and specificity of the chosen domain, the current project had a bottleneck in its knowledge elicitation component associated with the implementation of the Automatic Design Assistants (the expert systems). This is a statement common to most expert systems development reports. In fact, in the current situation, the traditional scarce availability of knowledge engineers is worsened by the relatively low number of available experts involved in the design of agricultural buildings. Furthermore, a pure design problem is being considered as opposed to typical and well covered non design problems such as diagnosis ones.

The knowledge elicitation methods more widely used may be separated into two main groups:

- psychological techniques [13];
- induction [14].

The first group comprises some of the most popular and better tested methods, such as:

- interviews;
- protocol analysis;
- multidimensional scaling (e.g., Kelly grids);
- concept sorting.

Although some of this methods evolved into very well structured routines, and even into computer programs to be directly used by the expert(s) (such is the case of Kelly grids, for instances) some very pragmatic authors as Parsay and Chignell [15] claim that "each knowledge acquisition method developed in the late 1970s and early 1980s has been a variant on the general theme of *talking to the expert*".

Induction is, the authors believe, reasonably inadequate as a major knowledge acquisition technique in a structural design problem, although the system should contain some rule induction capabilities (see 3.). Rule induction defenders seem to agree [14]. This idea can be partially supported by some of the adopted invariants of design (2.1) – inv1, inv2 and inv8.

7.2 A new kind of knowledge engineers

In the psychological techniques group, interviewing and protocol analysis seem to form the core of the so called "team approach". In this situation a team formed of knowledge engineers and domain experts work together to produce (and validate) a knowledge base (or the expert system itself). Under such a working environment a number of successful and well tested routines and methodologies have been used to elicit and organize experts' knowledge mostly in diagnosis problems. Some outstanding products emerged of such an approach. This is the case of PROSPECTOR a geology consultant for hard rock mineral exploration which has already predicted the occurrence of very important molybdenum ore deposits, Duda et al. [16] and XCON (formerly R1), an assistant for computer systems configuration which actually replaced humans in its domain of expertise, McDermott [17].

However, acquiring knowledge in a design project needs a special approach as synthesis plays a major role in design problems (5.1), the knowledge involved in synthesis activities being extremely hard to elicit. The problem with knowledge about synthesis is that experts are very often unaware of the structure of their own knowledge. As the decisions they made seem so obvious to them, they are not able to isolate and identify something close to "design rules" which can be programmed in a knowledge based system. It is the role of the knowledge engineering team and methodology to make the acquisition of that knowledge possible.

Due to the mentioned difficulties, the authors believe that, especially in design problems, the "team approach" should evolve into a situation in which pure domain experts need to be slightly educated on expert systems technology and pure knowledge engineers (traditionally, computer scientists) should give place to domain-almost-experts with a strong background of cognitive science, design theory, knowledge engineering and expert systems technology. This is the situation of the current project, in which structural engineers deeply involved in expert systems research, have been interviewing and observing structural engineering experts involved in the design of steel portal frames for agricultural buildings. This leads to a better efficiency in terms of elicitation of the experts' personal knowledge giving deeper access to abstract conceptual knowledge and knowledge of patterns (3.).

7.3 A knowledge elicitation methodology driven by facts

The knowledge elicitation methodology developed and used in the current project was originally based in the work by Grover [18], who first proposed the establishment of a Knowledge Acquisition Documentation Series. Such formalized documentation, by providing strong guidance and self monitoring during the all knowledge acquisition process, would be a first preventive step against the usual intuitive interpretation of psychological techniques such as interviewing or protocol analysis.

Grover's proposal for a knowledge acquisition (KA) cycle identifies three separate stages, as pictorially described in Figure 3 (reference [18] has a complete description of this cycle):

- Domain definition;
- Fundamental knowledge formulation;
- Basal knowledge consolidation.;

Considering the specific case of knowledge acquisition in a structural design context, a modified version of Grover's KA cycle is presented (Figure 5).

7.3.1 Domain definition

Being the knowledge acquisition team formed as described above (7.2), most of the Problem Definition items of Grover's routine become either unnecessary or of less importance, namely those concerning bibliography and glossary.

On the other hand, keeping in mind that a design system is being built it is of vital importance, during this stage, for the expert(s) to become briefly acquainted with the model of design being implemented.

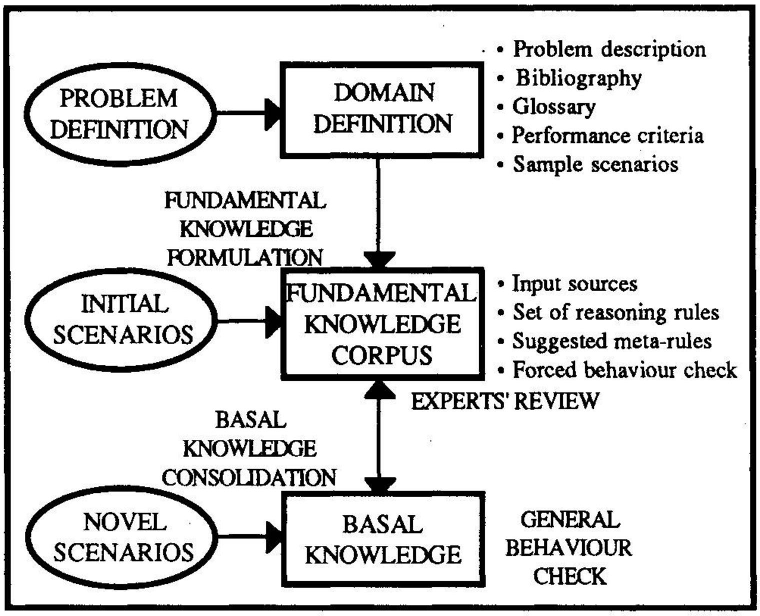


Figure 4

Also at this level of the knowledge acquisition process, the expert(s) should provide a list of questions to be answered before each designs starts, as discussed below (7.3.2).

As opposed to Grover's suggestion, the authors believe that performance criteria should not be explicitly discussed with the expert(s) at this early stage, to avoid unreasonable expectations.

7.3.2 Informal knowledge formulation

As mentioned before, some of the experts' personal knowledge can hardly, or not at all, be made explicit. Most experts would not identify a design rule they use even if confronted with a situation where they would use it. On the other hand, facts, as design requirements or constraints, are extremely easy to extract, i.e. the expert(s) can easily identify which questions need to be answered so that they may start or proceed with a particular design. The list of possible answers to those questions can be formalized as a set of first order logic ground clauses representing facts, e.g:

Question	Answer	Fact
• Which code to use in the design ? (BS5502, Eurocode3, ...)	the code to use is BS5502	code to use(BS5502);
• Are there dominant openings ? Where ? (sides, roof)	the roof has dominant openings	has(dominant_openings, roof);
• Who is going to build the foundations ? (contractor, farmer,...)	the farmer will built the foundations	builder(farmer, foundations)

For the informal knowledge formulation, the expert(s) can be confronted with the list of facts elicited during Domain Definition. It becomes then easier for the expert(s) to explicate which rules use these facts. This behaviour corroborates the organization of personal knowledge (3.) proposed by Goel and Pirolli [2].

A practical example of such a result is described:

– During initial interviews between a knowledge engineer and an expert in the design of agricultural buildings, the expert identified a number of questions he needed to place to his clients, in order to commence the design. Later on, at the Informal Knowledge Formulation phase, two of such questions seemed to be of no use for the design:

- Is there a soil survey for this building site ?
- Who is going to build the foundations, the farmer or a contractor ?

The answer for this questions could have produced the following facts (among others):

A soil survey was made for this site → exists(soil_survey);
The farmer will build the foundations → builder(farmer, foundations);
A contractor will build the foundations → builder(contractor, foundations);

Confronted with these possible facts, the designer immediately provided his use of that information, and a new design rule could actually be written:

– If there is no soil survey for the building site and the farmer will build the foundations himself, then the safety factor for the portals' design should be the highest possible in the set of design codes being used

or, design safety factor(highest) if don't know exists(soil_survey) and
builder(farmer, foundations)

Following Grover's proposal [18], a validation of this informal knowledge base should be attempted by producing, for specific scenarios, the behaviour expected by the expert. Only then, the elaboration of a formal knowledge base should proceed.

7.3.3 Knowledge base formulation

Starting from a validated natural language knowledge base, this phase should aim at establishing a formal one, assuming the authors' commitment to first order logic and frames as the knowledge representation formalisms to use (as discussed in paragraph 3.).

At this stage, adequacy of the elicited knowledge as regarding the model SAE is to be enforced. Major attention should also be devoted to the adequacy of the current knowledge base with external procedural programs and cooperating expert systems [11].

An iterative reorganization of the informal and formal knowledge bases should finally take place, as corroboration and validation activities suggest. This should include validation by external experts and end users of the system.

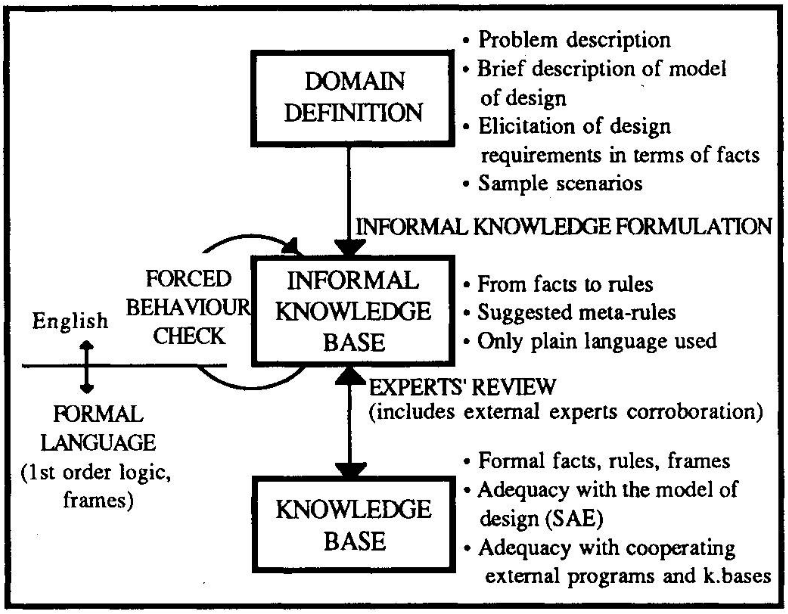


Figure 5

8. CONCLUSION

As fundamental pieces of the implementation of an intelligent computer aided (structural) design system (ICAD), a formal definition of design and a computable model of the design process were presented and briefly described in this work. First order logic and frames were identified as privileged knowledge representation formalisms.

Some major implementation issues were raised and discussed, namely the need for selecting a specific domain of application to pursue implementation and development of the model of the design process.

Some typical knowledge elicitation problems were identified and a structured routine used and enhanced during this project was described.

The ideas and methods presented in this article are being tested by their current application to an ICAD system for assisting in the design of steel portal frames for agricultural buildings.

ACKNOWLEDGEMENTS

The authors would like to thank the partial financial support provided by the Steel Construction Institute (UK) and the Calouste Gulbenkian Foundation (Portugal), in the final stages of this project.

Special thanks are due to Mr. Gordon Rose, Head of Structural Engineering at the Ministry of Agriculture, Fishery and Foods (UK).

REFERENCES

1. FEIJÓ, B., Fundamental steps towards an intelligent CAD system in structural steel, PhD Thesis, Expert Systems Laboratory, Dept. Civil Eng., Imperial College of Science, Technology and Medicine, London, 1988.

2. GOEL, V. and PIROLLI, P., Design within information-processing theory: the design problem space, AI Magazine, spring 89, 1989, pp. 19-36.

3. NEWELL, A. and SIMON, H.A., Human Problem Solving, Prentice-Hall, 1972.

4. VETH, B., An integrated data description language for coding design knowledge, in ten Hagen, P.J.W. and Tomiyama, T. (eds), Intelligent CAD Systems I, Spring Verlag, 1988, pp. 295-313.

5. JOHNSON-LAIRD, P., The Computer and the Mind: an Introduction to Cognitive Science, Fontana Press, 1988.

6. DREYFUS, H.L., What Computers Can't Do: The Limits of Artificial Intelligence, (revised edition), Harper Colophon Books, 1979.

7. SEARLE, J.R., Minds, brains and programs, Behavioral and Brain Sciences, 3, 1980, pp. 417-424.

8. WINOGRAD, T. and FLORES, F., Understanding Computers and Cognition: a New Foundation for Design, Ablex, 1986.

9. OKI, A. and FEIJO, B., Explanation based learning and civil engineering expert systems, Technical Report ESL, Expert Systems Laboratory, Dept. Civil Eng., Imperial College of Science, Technology and Medicine, London, (to appear).

10. DOYLE, J., A truth maintenance system, Artificial Intelligence, 12, 1979, pp. 231-272.

11. BENTO, J. and FEIJÓ, B., On the interaction between procedural and declarative programming styles for expert systems in structural engineering design, Technical Report ESL, Expert Systems Laboratory, Dept. Civil Eng., Imperial College of Science, Technology and Medicine, London, (to appear).

12. COTTA, J.C., DB-Prolog: A portable system based on simple and appropriate solutions, LNEC Thesis for Informatics Specialist, LNEC Proc. 71/13/7492, National Laboratory of Civil Engineering, Lisboa, 1986.

13. GAMMACK, J.G and YOUNG, R.M, Psychological Techniques for Eliciting Expert Knowledge, Research & Development in Expert Systems, Cambridge University Press, 1985, pp. 105-112.

14. BLOOMFIELD, B.P., Capturing Expertise by Rule Induction, The Knowledge Engineering Review, Vol.1 n. 4, 1986.

15. PARSAY, K and CHIGNELL, M., Expert Systems for Experts, John Wiley & Sons, Inc., New York, 1988.

16. DUDA, R.O., GASCHNIG, J., HART, P.E., KONOLIGE, K., REBOH, R. BARRETT, P. and SLOCUM, J., Development of the PROSPECTOR consultation system for mineral exploration, Final Report, SRI Projects 5821 and 6415, SRI International, Inc., Menlo Park, California, 1978.

17. McDERMOTT, J., R1: The formative years, AI Magazine, Vol.2, pp. 21-29, 1981.

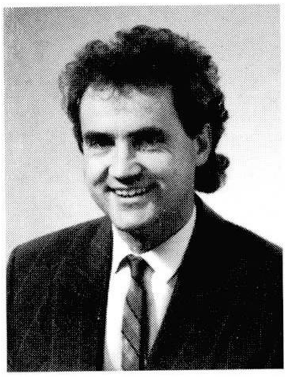
18. GROVER, M.D., A Pragmatic Knowledge Acquisition Methodology, Proceedings of the 8th International Joint Conference on Artificial Intelligence (IJCAI), pp. 436-438, 1983.

Architectural Pre-Processor for Engineering Expert Systems

Architecture d'un pre-processeur pour les systèmes experts d'ingénieurs

Präprozessor für architektonische Expertensysteme

Gerhard SCHMITT
Professor Dr. Ing.
Swiss Fed. Inst. of Technology
Zürich, Switzerland



Gerhard Schmitt, born in 1953, received his Architecture degree from the University of California at Berkeley and his PhD from the Technical University of Münich. Since 1984 Professor at Carnegie Mellon University, USA, he now holds the chair for CAAD. at ETH..

SUMMARY
In this paper, we present three issues important for the development of knowledge-based Architecture and Engineering design environments: first, intelligent interaction and feedback mechanisms between engineering and architectural design and their computational treatment, introducing classes of design and classes of tools. Secondly, the critical presentation of ARCHPLAN, a working prototype within IBDE, the Integrate Building Design Environment. Thirdly, thoughts on the role of criticism between programs and their implementation.

RESUME
Dans cet article nous présentons trois domaines importants pour le développement d'environnements de conception à base de connaissances en architecture et en ingénierie: Premièrement, et les méchanismes intélligents d'interaction et d'information du retour (feedback) ainsi que leur traitement interne par ordinateur, en introdusant un concept de classes de conception et d'outils. Deuxièmement, la présentation critique d'ARCHPLAN, un prototype fonctionnant déjà dans l'environnement de conception IBDE (Integrated Building Design Environment - environnement intégré de conception des bâtiments). Troisièmement, des réflexions sur le rôle de la critique entre les programmes et leur implementation.

ZUSAMMENFASSUNG
In diesem Artikel stellen wir drei Gebiete vor, die für Entwicklung von wissensbasierten Architektur- und Ingenieur-Planungsumgebungen von Wichtigkeit sind. Erstens, intelligente Interaktions- und Rückkoppelungsmechanismen, sowie deren computerinterne Behandlung, wobei das Konzept von Entwurfsklassen und zugehörigen Entwurfswerkzeugen vorgestellt wird. Zweitens die kritische Präsentation von ARCHPLAN, einem bereits funktionierenden Prototyp innerhalb von IBDE (Integrierte Gebäude-Entwurfs-Umgebung). Drittens, Gedanken und Rolle von Kritik zwischen Programmen sowie deren Implementation.

Abstract

High quality design under time constraints in architecture and the building industry is becoming one of the critical requirements in the development of new products. In their respective domains, knowledge based systems in diagnostics, maintenance, and construction have reached a high degree of sophistication, whereas less examples exist in the area of design and the integration of interdisciplinary knowledge. One of the problems is the loss of vital, qualitative information in the process of transferring an architectural design to further engineering synthesis and analysis. While only in the fewest cases the structural engineer is expected to deliver the architectural design or the architect is expected to complete the structural analysis, the two activities are closely related by sensitive interdependences.
The purpose of this paper is to present a number of intelligent interaction and feedback approaches between civil engineering and architecture and to suggest their computational treatment within an environment of knowledge-based systems. It is philosophical in the sense that it proposes and critically describes attempts rather than solutions.
The paper is divided into three parts. Part One addresses feasible design processes and their representations. Part Two describes an Integrated Building Design Environment and the architectural preprocessor necessary to begin the engineering design as a prototype system. Part Three presents thoughts on criticism mechanisms between knowledge-based processes.

1. Part One: Feasible Design Processes and Representations in Architecture and Civil Engineering

Effective communication and exchange of information between disciplines often encounters a language and representation barrier. The use of different models to describe and reason about the world complicates matters and in many cases prevents designers from obtaining crucial feedback from other disciplines. There are, at the moment, three possibilities to escape this dilemma:

- Reliance on known algorithms and rules. An extensive body of knowledge exists in both the engineering and architecture literature describing solutions to known design problems. A practical example is the book "Entwurfslehre" by Neufert [1] which represents a significant collection of design knowledge and has been used and updated for the last 35 years. Similar books exist in civil engineering [2]. Books of this kind generally do not, however, help in solving new or unexpected design problems.
- Principled representation. The successful search for a generalized and principled representation that holds true for two or more disciplines could solve many of the communication and feedback problems between architecture and engineering. To solve similar representation problems is the central goal of qualitative physics [3], where models based on principled representations are extensively used [4]. Qualitative physics formalizes first-principles knowledge about physical phenomena. The resulting library of *domain models* provides knowledge for instantiating qualitative causal models of a physical system, such

as a steam plant, a mechanism, a structural system, or a building. It provides a uniform representation of a large class of phenomena, possibly covering several domains.

- Reasoning based on cases. The traditional knowledge representation paradigm of Artificial Intelligence is that of general production rules. Applied to design, it suffers from two shortcomings: (a) it is difficult to construct a coherent set of rules for representing an extensive body of knowledge, and (b) it is not clear how to formulate the knowledge needed to produce a complete design in the form of general rules. Furthermore, there is strong psychological evidence that people do not reason from general rules alone, but often refer to the memory of previously solved similar problems, as shown by Schank [5] and Akin [6]. This observation has led to the paradigm of case-based reasoning (CBR), which also helps to eliminate some of the deficiencies of rule-based systems.

While principled representation and case-based reasoning for design are the subject of on-going research, the first approach is computationally straight-forward and applied in some integrated system projects underway in Stanford [7], Carnegie Mellon [8], and Liege [9] The ARCHPLAN preprocessor described in Part Two is also based on this model and concentrates on sections of the architecture and civil engineering domains to explore the use of one representation and one model to facilitate the exchange of crucial information that goes beyond syntactic specifications. Levels on which common representations and models are desirable are:

- Low level representation. This includes syntactic representations of objects and functions to describe the geometry and purpose of designed artifacts. As differences in describing geometry do exist, standards that allow the transfer of descriptions between applications without information loss are of particular importance. For product modeling, one of the standards proposed is the STEP (STandard for the Exchange of Product model data) model [10].
- Intermediate level representation. This level mainly describes larger knowledge entities and may combine syntactic and functional descriptions in one representation. Commonly used are procedures, rules, frames, or objects, of which frames have emerged as the most flexible.
- High level representation. This includes models of expression for semantics or structures of design knowledge such as chunks and prototypes.

Beside these representations, compatible models of the design process itself in the respective disciplines are needed to go beyond sequential and thus cumbersome design simulations. Proposed by Gero, Maher, and Zhang [11], three different types of design processes in both architecture and engineering are proposed: routine, innovative, and creative design. Although such a simplification is not always acceptable and will not completely describe real-world design, it is useful to divide the otherwise too extensive field of design into manageable parts. The following is a brief description of the three types of design.

1.1. Routine Design

Routine design is a goal-directed activity, characterized by prototype refinement or instantiation of designs from a catalogue of parameterized examples. Beginning with a given prototype of, for example, a piece of furniture or equipment, the designer adjusts a number of parameters to the specifications of the design program. The parameters, typically geometric properties or materials, are normally well understood and are manipulated either in the designer's memory or with advanced modeling systems. The functional requirements of the design are known and the semantics or the teleology, (the purpose of each element) of the design are not changed but accepted from previous examples. Routine design relies heavily on instantiation of designs from a catalogue of parameterized examples which are considered relevant for the design problem at hand. Without doubt, routine design is a good preparation for innovative and creative design and its importance must therefore not be underestimated. The refinement of a standard floor plan is a good example. It could be claimed that most great architects started their career with routine design [12]. The example described later in the paper refers to this type of design.

1.2. Innovative Design

According to Faltings [13], this type of design could be described as prototype combination, which makes it a prime example for case-based reasoning. According to Gero, innovative design is achievable with prototype modification [11]. In both cases, the designer has a general idea of the desired object and the design process is, as in routine design, a goal-directed activity. However, the design process cannot be completed with routine design because the functional description or the object properties are not achievable utilizing a given prototype. Therefore, the combination of two or more prototypes which each have some of the desired properties is necessary. An example would be the development of intelligent office buildings for which some new information infrastructure needs are still unknown. Case-based reasoning and explanation-based learning systems are of particular interest in innovative design because they may selectively capture desired qualities of existing buildings and avoid their shortcomings. Once these qualities have been discovered, an existing prototype may indeed be modified to incorporate innovations.

1.3. Creative Design

Creative design is the development of new solutions that may only be partially defined at the outset. Both functional requirements and the object's properties are not completely known. It is possible that a unique solution may be found to a problem in which case the result would be an archetype. In most cases, prototype creation is necessary, which later can be combined and modified (innovative design) and instantiated (routine design). An example is the invention of a new machine, such as the personal computer. Once sufficiently understood and formalized, creative design becomes part of the mainstream and can be proceduralized. In some cases, architectural language aspects of once creative design, such as the Barcelona Pavilion or Richard Meier's residences, can be proceduralized or implemented in shape grammars [12].

1.4. Levels of Design and Levels of Representation

Until recently, commercial CAD programs typically allowed manipulation of design at the level of geometry or routine design: once a design is completed with traditional means and input into the computer, manipulations are performed on the geometric model. Macros containing simple checking rules for stair angles or distance calculations could be seen as intermediate level representations of routine design knowledge. And parameterized prototypes of furniture and stairs that interactively generate more complex, but well defined building elements, are examples of higher level representations in routine design. The existence of this complete set of representation and manipulation tools, developed to support day-to-day design activities, is one reason for the popularity of computers in routine design. In order to build as powerful tools to support innovative and creative design, the following table might serve as a preliminary attempt to relate types of design and levels of representation:

Level of Representation	**Routine Design**	**Innovative Design**	**Creative Design**
Low	Geometry	Syntax	Semantics
Intermediate	Rules, Frames	Rules, Frames	Prototypes
High	Prototypes	Object Semantics	Structures

2. Part Two: The IBDE Prototype

Based on the overview of design processes and representations in Part One, this section describes the Integrated Building Design Environment (IBDE) which eventually should have the capability to perform routine and innovative design at a level of competence and completeness approaching that of a human designer and out-performing a designer in terms of consistency and time required to complete a design.

IBDE integrates 7 independent, knowledge-based computer programs. Their declarative representation of knowledge permits rapid development and modification. The prototype integrated environment serves as a test bed for examining the following issues:

- Integration of multiple disciplines, such as architecture, engineering, and HVAC design. A positive result is a more realistic simulation of the final building in the design phase and the elimination of coordination problems that usually occur during construction and can lead to costly changes.
- Discipline specific multiple views of a building design. This issue is important because it allows complete and consistent observation of the design and prevents the discovery of conflicts at too late a stage.
- Improvement of communication between disciplines. Although more a side effect of the project, it has developed into one of its most positive aspects. An integrated design environment helps eliminate most common misunderstandings about engineering and architectural design.

integrated design environment helps eliminate most common misunderstandings about engineering and architectural design.

- Automation of sub-processes to various degrees. Wherever possible, and whenever enough knowledge is available for parameterization, processes may be automated. Examples are layout of elevator core areas and an HVAC design sub-system.
- Hardware and software independence. Traditionally hardware dependent applications can be used in a distributed environment. Parallel execution of individual processes is possible.

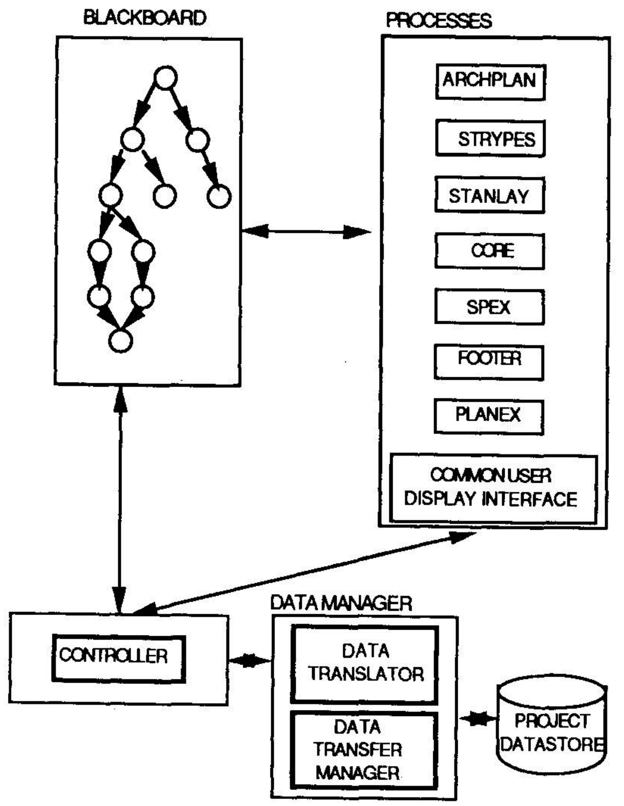


Figure 1: Schematic view of the IBDE architecture.

While the systems produces a complete and consistent design within reasonable time (two to three hours) and describes the building to a high level of detail, the system could be improved. One of the most important necessities is the implementation of critiques that act between processes. The blackboard, which is now used for posting status messages, needs to take on a more significant role. The individual processes need refinement as well: as of now, IBDE can only simulate rectangular office buildings with interior cores. Finally, development work is necessary on the common display interface and on the individual program interfaces to provide the designer with a more friendly discipline specific and unified view of the project.

2.1. ARCHPLAN - The Architectural Preprocessor

ARCHPLAN - ARCHitectural PLANning expert system - is the first of the seven knowledge-based processes and provides necessary input for IBDE to proceed. ARCHPLAN assists in the development of the conceptual design of high-rise office buildings. Input describes the site, the client's program, budget, and geometric constraints. The output produces three-dimensional functional, circulation, cost and massing information. ARCHPLAN uses prototype refinement to develop individual solutions from a generic prototype and may therefore be seen as an example for a routine design program. The user can freely move between four modules which are the site, cost and massing module (SCM), function module, circulation module, and structure selection module. Knowledge is stored in algebraic form and as heuristic rules. The program is implemented in common LISP with object-oriented extensions [14].

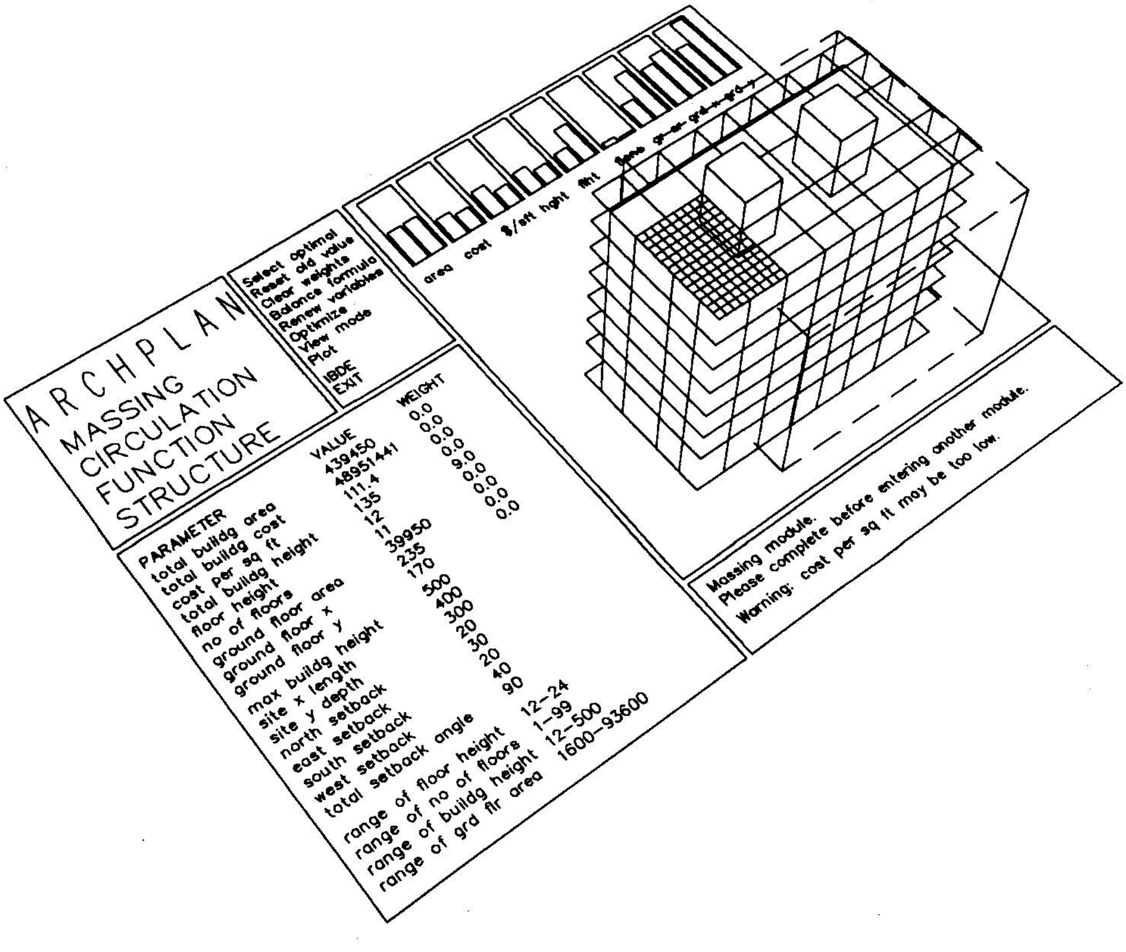


Figure 2: ARCHPLAN - designer's view of the extended site, cost, and massing (SCM) module

ARCHPLAN interaction begins with the *site, cost and massing module* (SCM). The designer finds a set of default values in an interactive window which can be modified at will. After a building site is determined, preliminary design begins with the development of a massing model that will accomodate a given budget and a range of other parameters listed in Figure 2. Cost, site and massing options are interdependent concerns. Site characteristics are considered to be facts and therefore fixed, whereas building requirements are more flexible. The user describes the degree of commitment to a certain requirement, such as floor-to-floor height or ground floor area, by entering a certainty factor. The SCM module also contains simple optimization options: minimum cost, maximum daylighting, or a combination of the two. Cost data are based on the Means catalogue, a prominent summary of building cost data in the United States. The calculated cost is a total per square-foot number and includes interior and exterior construction as well as finishings. Design factors are represented as objects. Relations and constraints between factors are expressed within those objects [15].

Critical issues are the restriction to rectangular building shapes and the fact that relations between individual knowledge objects can be changed only by modifying the source code. User-defined additions of new considerations and non-monotonic reasoning for the design phase are planned for the next release.

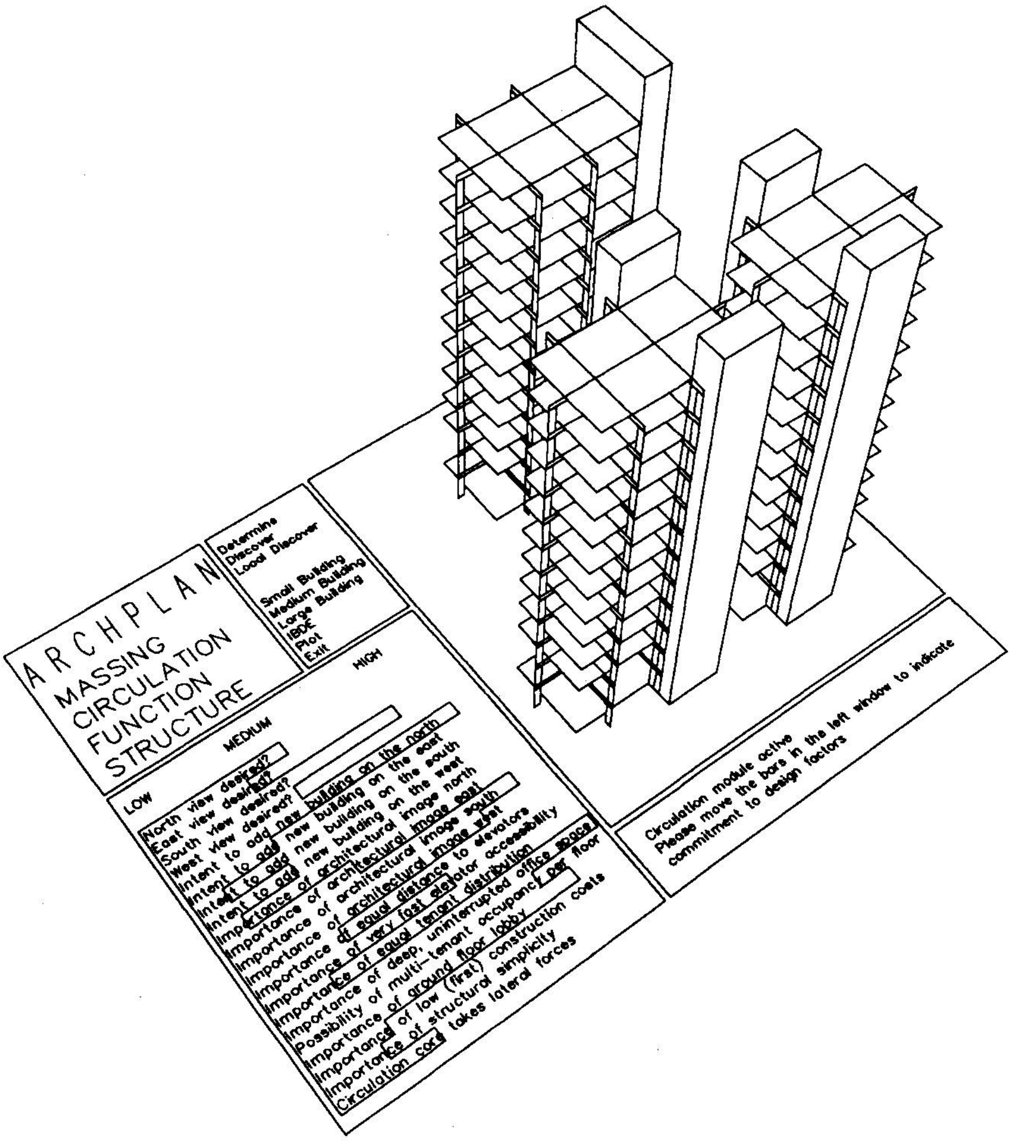


Figure 3: ARCHPLAN - designer's view of the extended circulation module

The *function module* assists in the vertical and horizontal distribution of building functions within the basic massing volume. Examples of building functions are office, retail, atrium, mechanical and parking space. Each function has particular requirements and affects the layout, appearance, and cost of the building. ARCHPLAN proposes a three-dimensional layout scheme which is displayed in solid or wire frame representation. Functional decisions are made and reflected locally, unless the constants in the global building description object are violated. In this case, the program backtracks and control is passed back to the SCM module where the designer can choose either to automatically adjust the design description to the information received from the function module or to implement changes manually. Critical issues are the shallowness of the knowledge used to determine three-dimensional functional layout. Although a number of existing buildings were used as examples, we did not discover all reasons behind a particular layout. The function module can therefore not explain its decision but executes them algorithmically.

The *circulation module* addresses the problem of moving occupants and equipment from floor to floor and within floors and guaranteeing the safe evacuation of the occupants in emergencies. Circulation also has a major impact on the internal functioning and on the architectural expression of a high-rise building. The two extreme cases for the placement of vertical circulation are the internal (service and elevator core in the centre of the building) or the external solution (service and elevator cores attached to the outside of the building). ARCHPLAN concentrates on creating vertical circulation proposals based on variations of these two prototypes. The user manipulates the relative importance of each factor leading to the final design proposal by sliding graphical bars, a more user-friendly but less exact interaction than typing in weighting factors as in the site, cost, and massing module. Figure 3 shows a view of the user interface for the circulation module.
Critical in this module is again the quality of knowledge that leads to a particular placement and size of vertical elements. Future versions must include user-definable criteria and their spatial consequences. The module does not provide for the decrease in the number of elevators in the upper levels. This calculation is handled by CORE, the elevator layout program developed by Flemming [16].

The *structure module* presents the user with a choice of eight structural systems, some of which may apply for the proposed design. The program compares the state of the design object with the characteristics of each structural type and decides which type is not applicable and available for the present design. Adjustments to the structural grid are still possible at this point. As these decisions are more competently handled in STRYPES and STANLAY, the structure module is available only when ARCHPLAN is used in stand-alone mode.

The program has gone through several revisions and is presently ported to a SUN 4 environment. In the absence of an established paradigm for architectural *design* programs, documentation of the experimental LISP source code is sometimes sketchy. ARCHPLAN is interesting for mainly two reasons:

- It makes architectural design knowledge explicit as it implements decision-making design processes in various representations. Thus, critique may be voiced openly and representation and knowledge manipulation methods for design can be improved.

- It addresses important user interface issues. The common display user interface employs direct graphical object manipulation techniques to give immediate feedback. Architectural decisions are visualized rapidly and facilitate the commencement of design from either very little or very detailed client information.

3. Part Three: On Criticism Between Processes

The introduction of criticism mechanisms between programs is based on the assumption that (a) one program alone will not be able to provide a solution to a complex problem, such as the design of a building, and that (b) a sequential execution of related knowledge-based programs will render the design process too cumbersome and does not allow for the exploration of enough alternatives.

Common representation of design objects and design processes is one enabling concept for the exchange of and the reaction to design criticism. It is assumed, as is the case in the IBDE project, that a number of programs in a knowledge-based environment (named controller, A, B, . . . in the following) are to interact, exchange information and criticism, in order to improve the final design. Design criticism may be grouped into several levels:

- The first level (on/off) is that program B, using input from program A, cannot start. Program B posts the message "Unable to commence". It is the responsibility of program A, another program, or the controller to react to the message and provide new input so that the computation can begin. This is a low level operation comparable to the range checking in data base input operations.
- The second level (impasse) is that program B, using partial input from program A, encounters an impasse in its reasoning process that it cannot resolve with the knowledge it has access to. Program B posts the message "Impasse". It is the responsibility of program A, another program, or the controller to react to the message and provide new input so that the computation can proceed.
- The third level (quality) is that program B reaches a solution that is acceptable, but has severe, definable drawbacks. Program B posts the message "Improve Quality (in a specific area)". It is the responsibility of the preceding program(s) or of the controller to provide new input.

The formulation of and the reaction to criticism becomes more difficult from the first through the third level. In the IBDE project, these levels could map to the following scenarios:

- First level. STRYPES, the structural system configurer (in the above example, program B) needs not yet existing input from ARCHPLAN (in the above example, program A). STRYPES posts the message "Unable to commence - respect constraints in structural grid - longest span less or equal to 35 ft - shortest span greater or eqal to 25 ft" on the blackboard, the controller initiates the execution of ARCHPLAN; it re-starts STRYPES once ARCHPLAN has completed a configuration.

- Second level. CORE, the space planner for the service core, attempts to fit the necessary elevator banks and service areas into the space defined by ARCHPLAN. CORE's knowledge base is unable to fit the required functions into the given area. Rather than continuing and creating an inconsistency or unilaterally updating the project data store, CORE posts the message "Impasse - core size must be greater or equal to 550 sqft - smallest side longer than 10 ft". The controller notifies ARCHPLAN which then should re-execute its circulation module. Once CORE's critique is satisfied, the impact on the other processes of changing the size and possibly the location of the elevator core in ARCHPLAN must be checked: if necessary, parts of STRYPES, STANLAY, CORE, SPEX, FOOTER, and PLANEX must re-execute as well.
- Third level. CORE has produced a layout for the elevator zone. A change in the city zoning regulations requires re-execution of the circulation and function module in ARCHPLAN. Although technically feasible, the orientation and individual layout of the core zone is no longer satisfactory for the new situation. This decision is either made manually after visual control of the core layout and the new situation, or by an additional design quality knowledge module in ARCHPLAN which regularly checks the architectural impact of solutions proposed by CORE and the other processes. ARCHPLAN posts the message "Improve Quality - re-configure core layout - core entrance should face east" on the blackboard. The controller restarts CORE, given the new information concerning orientation, and CORE proposes a new solution which is passed back to ARCHPLAN. As in the second level, impact on the other processes must be checked at this point. In re-executing processes, level one critiques have highest and level three critiques have lowest priority.

One major problem in this scenario is the circularity of critique, re-execution, and propagation of new results, possibly ending in endless loops of program execution. The settling of conflicts created by reactions to critique from other processes is not a mechanical, value-free activity, but involves judgement. This calls for high level decision knowledge in the controller or in the individual processes which should possibly allow for temporary inconsistencies in the building representation. We have not found a solution to this important problem yet. However, a system of this type would make any interdisciplinary cooperation of programs more realistic and possibly improve the results.

4. Conclusions

In writing an architectural preprocessor for engineering expert systems similarities and differences between the two disciplines become apparent. The same would probably hold true if more than two areas were involved. The first approach was to use knowledge representations that were already tested in architecture and civil engineering and to explore if there was a smallest common denominator for representing architecture and engineering design. With the exception of the Tartan grid as representation of geometry and frames as containers of knowledge, almost everything was different. This lead to the present architecture of IBDE, in which a global data store is the common depository of building data which is used as needed by the individual processes, and the common user display interface which interactively displays the content of the global data store. The advantage of this loosely coupled approach is a high degree of freedom in the processes; a drawback is the inevitable loss of high level, application specific information which may be crucial for the meaningful completion of other processes. The existence of the blackboard eases this situation somewhat and the introduction of critique mechanisms between processes is another means to achieve the appropriate balance between global and local information preference. It shows, however, even more the need for a general building description language which would be capable of operating on a design and construction model of growing complexity.

5. Acknowledgements

The IBDE project is supported by the National Science Foundation of the United States in the Engineering Design Research Center at Carnegie Mellon University in Pittsburgh. Professors Steven Fenves, Ulrich Flemming, Chris Hendrickson, and Mary-Lou Maher are the developers of IBDE and the individual knowledge-based processes shown in Figure 1. Special thanks to research assistants Chen-Cheng Chen, Chia Ming Chen, and Shen Guan Shih for their development work in ARCHPLAN.

6. References

1. NEUFERT, Ernst. *Bauentwurfslehre - Grundlagen, Normen, und Vorschriften.* Viehweg Verlag, Wiesbaden. 1980.
2. SCHLEICHER, F. *Taschenbuch für Bauingenieure - Tome I und II.* Springer Verlag, Berlin. 1955.
3. HAYES, Patrick. *The Naive Physics Manifesto.* ISCCO Working Paper. 1978.
4. FORBUS, Ken. *Qualitative Process Theory.* Artificial Intelligence, Volume 24, 1984.
5. SCHANK, R. *Dynamic Memory - A Theory of Reminding and Learning in Computers and People.* Cambridge University Press, 1982.
6. AKIN, Ömer. *Psychology of Architectural Design.* Pion, London. 1986.
7. STANFORD University. *CIFE - Center for Integrated Facility Engineering.* Project description summary. Terman Engineering Center, Stanford University, Stanford, CA. 1989.
8. FENVES, S. U. Flemming C. Hendrickson M. Maher and G. Schmitt. *An Integrated Software Environment for Building Design and Construction.* Proceedings of the 2nd Int. Symposium on CAD in Architecture and Civil Engineering, ARECDAO '89, Barcelona, Spain. April 1989.
9. DUPAGNE, A. *Models of Building Representation Referred to Designing and Evaluating Process.* Proceedings of the 1st. Convegno Nazionale, Consiglio Nazionale delle Ricerche, Progetto Finalizzato Edilizia, Sorrento, 1989.
10. GIELINGH, Ir Wim. *Computer Integrated Construction, a major STEP forward.* Proceedings of the 2nd International Symposium on CAD in Architecture and Civil Engineering, ARECDAO '89, Barcelona, Spain. April 1989.
11. GERO, John, Maher, Mary Lou and Zhang, Weiguang. *Chunking Structural Design Knowledge as Prototypes.* The Architectural Computing Unit. Department of Architectural Science, University of Sydney, Australia. January 1988.
12. SCHMITT, Gerhard. *Microcomputer Aided Design.* John Wiley & Sons, New York. 1988.
13. FALTINGS, Boi. *Qualitative Kinematics and Computer-Aided Design* in: Proceedings of the Second IFIP WG 5.2 Workshop on Intelligent CAD, Cambridge, UK. September 1988.
14. SCHMITT, Gerhard. "ARCHPLAN: An Architectural Front End to Engineering Expert Systems", in Rychener, Michael (ed.) *Expert Systems for Engineering Design.* Academic Press, New York. 1988.
15. GOSLING, James. *Algebraic Constraints.* PhD Thesis, Department of Computer Science, Carnegie Mellon University, Pittsburgh. 1983.
16. FLEMMING, Ulrich, Coyne, R., Glavin, T., and Rychener, M. "A Generative Expert System for the Design of Building Layouts - Version 2", in Gero, John (ed.) *Artificial Intelligence in Engineering: Design.* Elsevier (Computational Mechanics Publications), New York. 1988.

Case Studies in Construction Automation

Etudes de cas dans l'automation des constructions

Fallstudien für die Automation in der Bauindustrie

Fabian HADIPRIONO
Civil Engineering Dept.
Ohio State University
Columbus, Ohio, USA

Fabian Hadipriono, born 1947, earned his doctorate degree at the University of California, Berkeley. He is active in research dealing with failure, reliability, and safety analyses. He is now teaching at The Ohio State University.

SUMMARY
Increasing complexities and highly specialized construction operations necessitate automation in construction industries. The application of knowledge-based decision support and expert systems is particularly timely. Examples of the application of these systems are presented in the area of construction operation quality control, equipment selection, and value engineering.

RESUME
L'augmentation de la complexité et le haut degré de spécialisation des opérations de construction nécessite l'automatisation de l'industrie des constructions. L'application de systèmes de soutien à la décision basés sur la connaissance et de systèmes experts est particulièrement opportune. Des examples d'applications de ces systèmes sont présentés dans le cadre des opérations de contrôle de qualité des constructions, du choix des équipements et de la conservation de la valeur des constructions.

ZUSAMMENFASSUNG
Die zunehmende Komplexität der hochspezialisierten Bauvorgänge erfordert im Bauwesen eine gewisse Automation. Die Anwendung von wissensbasierten Entscheidungshilfen und von Expertensystemen ist angezeigt. Beispiele derartiger Anwendungen auf den Gebieten der Qualitätssicherung, der Geräteauswahl und der Werterhaltung werden beschrieben.

1. INTRODUCTION

Construction expertise is commonly achieved through several years of on-site construction experience. However, rapid technological advances currently characterizing the construction industry do not allow engineers the luxury of slowly acquiring their qualifications on the job. Furthermore, the variety of the constructed facilities, materials, and technologies makes it impossible for one to achieve the expertise in various construction fields during one's lifetime. Some of the expertise is transferred or disseminated to others, while much more is lost.

In addition to this, construction projects are often characterized by many complicating realities, such as: differing cultures, customs and languages; remote sources of materials; shortages of skilled labors; local construction regulations; and large geographical distances between the construction sites and the home office. All of these realities lead to problems which require fast, effective solutions. The problems involved in such construction activities must be solved under the direct pressures of meeting a rigid quality standards and time schedules, budgets, and safety requirements. Managers must accomplish their goals while using limited resources on projects which differ widely from any experienced before.

As a result, adequate substitutions for construction experience are fast becoming a necessity. One promising way to overcome the above problems is through construction automation, such as through the use of knowledge-based decision support or expert systems. These systems fall in the area of the artificial intelligence (AI). Earlier studies related to expert systems have been presented in international conferences. Fuzzy reasoning expert system developed for assessing damage level of protective structures based on the repairability, functionality, and structural integrity criteria was presented in conferences in China [6] and Japan [4]. Another expert system developed for assessing the causes of construction failures in a concrete beam was presented in a Robotic Symposium in Israel [3]. An application of AI to Value Engineering was presented in an International Symposium in Indonesia [1]. This paper evaluates the applications of knowledge-based systems developed by the author for various construction purposes, such as, for construction operation quality control, construction equipment selections, and construction value engineering.

2. CONSTRUCTION OPERATION QUALITY CONTROL

In order to substitute field experience with adequate quality control, one has to recognize the factors which impact the performance of construction operations. These factors are classified into three categories, i.e., factors concerning site activities, factors concerning home office activities, and factors concerning the construction business environment. An expert system developed for this purpose is called the Integrated Management Information System (IMIS). Due to space limitations only factors concerning site activities are discussed here. These factors impacting construction site activities have been identified and defined earlier in another paper [2]. They are cost achievement, construction performance degree, performance of the project manager, administrative efficiency, labor control, material control, equipment control, and site management contingency.

The Cost Achievement (CA) depends upon the Contract Cost (CC), Target Cost (TC), and Actual Cost (AC). The interrelationships of CC, TC, and AC are the subfactors which can be exemplified as follows: If CC is larger than TC and TC is larger than AC, then cost achievement, CA, is very good. The Performance Degree (PD) is identified by the existence of an incentive or dispute or claims. As an example, if the construction project has built in financial incentives from the out set and no dispute arises, then the PD may be considered excellent. The performance of a Project Manager (PM) depends upon his/her competence and balance between authority and responsibility. For instance, a competent project manager whose authority equals his/her responsibility may be rated as very good, while an incompetent project manager whose reponsibility is greater than his/her authority can be rated as extremely poor. The Administrative Efficiency (AE) depends upon the organization and interrelationships among the staff and personnel. For example, an excellent AE can be achieved if a strong organization exists and if the site office is staffed with compatible and efficient personnel.

A construction project requires effective control of resource-related activities such as labor, material, and equipment needed to complete the project efficiently. The Labor Control (LC) is devided into three subfactors: productivity, wage, and working condition. Two subfactors identified within the context of Material Control (MC) are cost and handling of material, while Equipment Control (EC) depends on the productivity and cost. Site Contingency (SC) has frequently been overlooked. Potential contingencies should be identified, and then measured with the lowest possible cost. For example, a highway construction built on a government land in the middle of a farm country in South East Asia required the erection of high livestock type fence along the project. However, soon after the fence was erected, the local farmers whose life line was affected, began to protest and eventually vandalized the fence. A measure was then compromised by erecting pedestrian overcrossing bridges in the affected area. Should such a contingency be identified earlier, adequate measure can be planned at a more adequate time at a reasonable cost.

An expert system shell, 1ST-CLASS [7], was used for developing the production rules using the above factors, subfactors, and their values. Most of the rules use qualitative linguistic values. The shell was designed for IBM personal computers and other compatibles. The forward chaining process was used to infer the rules in the knowledge bases. Further inferencing is performed when a factor is selected. For example, if we are interested in the Cost Achievement (CA), the second line is selected and Figure 1 will appear on the screen showing nine production rules for CA in the form of a decision tree. Essentially, the decision tree shown in this figure is composed of the conventional IF-THEN statements. For example, the first rule (line 1 through 3) in the above may be read as follows:

IF (CC vs TC) leads to (CC is larger than TC)
AND (TC vs AC) leads to (TC is larger than AC)
THEN CA is "Very Good"

In this example, the linguistic value of CA in the THEN statement is "Very Good." The values used in the rules are: Excellent, Very Good, Good, Fairly Good, Fair, Fairly Poor, Poor, Very Poor, Extremely Poor, Undecided/Unknown. The software, IMIS, was developed for integrating information gathered from all factors described previously. This integrated information can be used to assess the performance of a construction project, to compare the performance of several projects, and to make decisions for improving the performance or progress of a project.

3. CONSTRUCTION EQUIPMENT SELECTION

Many factors must be considered in selecting equipment for use in concrete placement during building construction. These factors are based upon experience, judgment, and computational procedures. Factors, such as, equipment working space limitation, operator effectiveness, and equipment versatility are generally assessed based upon experience and judgment. Other factors, such as, equipment capacity and the minimum cost for hiring equipment are generally computed from available information. The use of a concrete operation decision support system (CODSS) is introduced here to help engineers make decisions for obtaining an optimal approach to concrete placement in building components [5]. The CODSS is a preliminary system that can be extended by adding more rules pertinent to the decision making. The expert system shell,"1ST CLASS," is again used for constructing the required knowledge bases.

The knowledge representations consist of four levels as shown in Figure 2. The first level is related to the building components that will be poured, the second level includes the constraints that may exist in the operations, the third level incorporates the selection of equipment, and the fourth level is associated with the computational procedures for finding the duration, cost, and production rate of the selected equipment. The task of placing concrete can be performed through the use of cranes and buckets, concrete pumps, pressure sprays, or conveyor belts. CODSS is limited to the use of mobile cranes, internal cranes, external cranes, and concrete pumps, since they are most likely used for concrete placing in multi-story buildings. The building components generally consist of footings, columns, beams, slabs, and walls. Each of these components may call for different equipment and approaches in concrete operations. Therefore, the choice of the component may result in an individual knowledge base.

Several factors dictate the use of concreting equipment. Five factors that are considered in CODSS are: 1. the specification that may or may not dictate the use of such equipment, 2. the working distance or working line between the crane or pump to the pour location. 3. the availability of working space that may affect the choice of concreting equipment, 4. the versatility and adaptability of the equipment selected, and 5. the operator's effectiveness in operating the equipment. Pumped concrete is usually conveyed by pressures through pipes or flexible hose and discharged directly into the desired area. The types of pumps considered here are: piston pumps, pneumatic pumps, and squeeze pressure pumps. Factors considered for pump selection are the pumping distance, concrete mix slump, and the pipe diameter. The cranes considered in this study are the mobile crane, internal tower crane, and external tower crane. In order to decide which type of cranes are most suitable for use in a project, the following factors should be considered: the building height, the ratio of the length to the width of the building, and specific requirements for an internal crane (e.g., a climbing crane may be called for use in a multi-story building since the mast of the crane can move upward as construction progresses). All cranes are assumed to have the capacity for lifting the bucket containing the concrete mix.

CODSS consists of 11 knowledge bases constructed within an expert system shell. An example of a rule tree (consisting of a set of production rules) for equipment selection based upon the operational constraints is shown in Figure 3. For instance, the third branch from the top of the rule tree can be rewritten as follows:

IF the SPEC. REQ. is optional and
the height of the building is over 600 ft.
THEN infer to CRANE

The consequent (THEN statement) will infer this branch to another rule tree related to CRANE, and so on. CODSS also incorporates 3 external programs written in BASIC that can be called into 1ST CLASS. These programs are used to compute the cost and productivity of the selected pumps and cranes. At this stage, the system is limited to concrete placing only; however, as research progresses, more activities can be accomodated by and integrated into CODSS.

4. VALUE ENGINEERING EXPERT SYSTEM

Value engineering (VE) in construction is a field of study emphasizing functional analysis of construction activities or items through a systematic and organized approach in order to obtain the required functions at minimum costs. Hence, an important part of this approach is the evaluation of the functions, where ideas, judgment, brainstorming, and services from experts are essential for the success of a VE study. In our earlier paper (Hadipriono and Chandra, 1987) the construction of a knowledge base containing production rules for the application of the Functional Analysis System Technique (FAST) was introduced. However, the application of expert systems in a VE study can be extended to the creation, evaluation, and selection of alternatives.

As an example, an item is determined as a <u>steel frame.</u> It is generally conceived that a frame consists of columns and beams. And the functions of the frame is evaluated as (using verb-noun): frame-building, support-loads, or provide-shape. The VE analyst determines which of the three functions is the most important. His/her choice becomes the basic function. The next logical step is to generate ideas that will perform this basic function besides the steel frame. This process is employed to create alternatives for performing the selected basic function. To continue our example, three reasonable alternatives are then created, they are: 1. cast-in-place concrete frame, 2. precast concrete frame, and 3. timber frame. This process of generating, evaluating, and selecting alternatives can be performed through the use of production rules. A user may use the rules by first identifying the desired item, activity, or function. The inference mechanism in the shell will then infer the desired item or function to the rules in the knowledge base. As an example, if the knowledge engineer wishes to relate the functions to the alternatives, then he/she can develop the following rules:

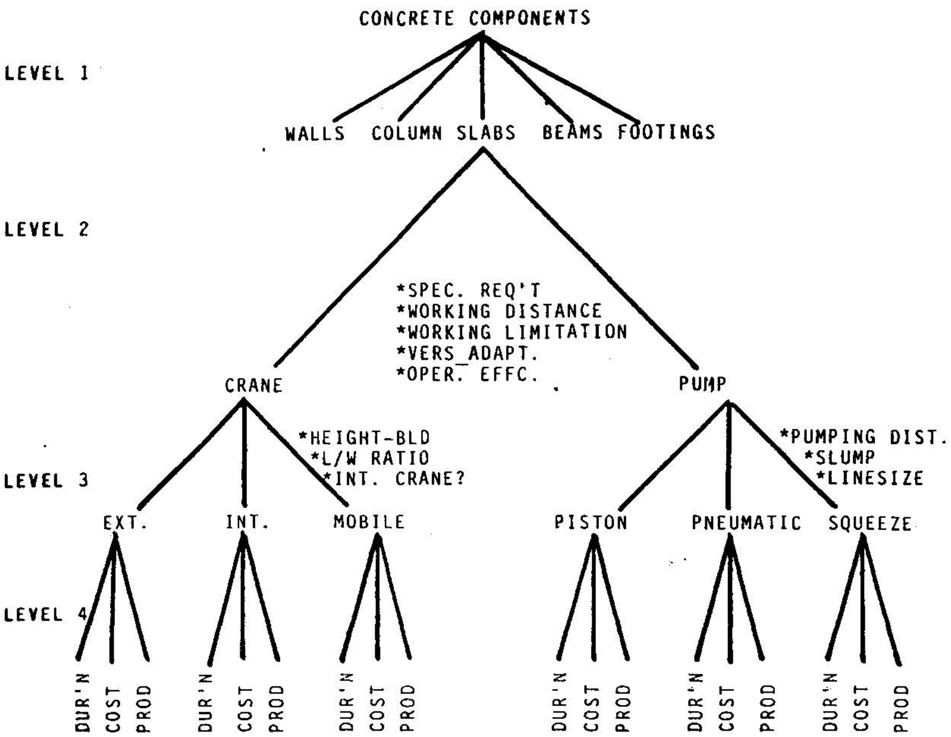
RULE A:
IF function is frame-building
THEN alternatives are:
1. cast-in-place concrete frame (0.25),
2. precast concrete frame (0.25),
3. timber frame (0.25), or
4. steel frame (0.25)

RULE B:
IF function is support-loads
THEN alternatives are:
1. cast-in-place concrete frame (0.3),
2. precast concrete frame (0.3),
3. steel frame (0.3), or
4. timber frame (0.1)

RULE C:
IF function is provide-shape
THEN alternatives are:
1. timber frame (0.4)
2. steel frame (0.3)
3. cast-in-place concrete frame (0.2), or
4. precast concrete frame (0.1)

Note that the figures within the brackets indicate the certainty factors. In this example, these factors determine the rank of the alternatives. In Rule A, these factors indicate that if the function is frame-building, then the alternatives will have the same rank. If the function is support-loads (Rule B), then the choice is equally important for cast-in-place concrete, precast concrete, and steel frames, but least important for timber frame. If the function is provide-shape, then the highest rank of alternative selection is timber frame while the lowest is precast concrete frame. Certainly, these factors could vary, depending upon the characteristics of the project, such as, location, fabrication methods, and erection techniques, and resource availability. For a more refined rules these characteristics are included in the IF statements of the rules. In general, these factors are obtained from the construction experts. For example, the first alternative in Rule C can also be interpreted as: 40% of the experts feel that timber frame is the best alternative for providing the shape of a building. With the help of experts, the knowledge engineer may continue develop and refine the production rules for evaluating each alternatives. The incorporation of menu driven options for assessing these alternatives could result in a powerful and user's friendly system. The engineer can further associate the alternatives with the cost analysis for selecting the best alternative. External programs may be needed for estimating the cost of each alternative before selecting the best one.

```
---- start of rule ----
CC vs.TC??
├CC>TC:TC.vs.AC??
│       ├TC>AC:──────────────────────────────VeryGood
│       ├TC<=AC:CC.vs.AC??
│       │       ├CC>AC:──────────────────────FairlyPoor
│       │       └CC<=AC:─────────────────────VeryPoor
│       └Und/Unk:────────────────────────────Und/Unk
├CC<=TC:TC.vs.AC??
│       ├TC>AC:CC.vs.AC??
│       │       ├CC>AC:──────────────────────Excellent
│       │       └CC<=AC:─────────────────────FairlyGood
│       ├TC<=AC:─────────────────────────────Extrem.Poor
│       └Und/Unk:────────────────────────────Und/Unk
└Und/Unk:────────────────────────────────────Und/Unk
---- end of rule ----
```




```
---- start of rule ----
SPECI-REQUI??
├crane:──────────────────────────────#CRANE
├pump:───────────────────────────────#PUMP
└optional:HEIGHT??
          ├over-600ft:───────────────#CRANE
          └under-600ft:#WORK-LIM??
                       ├space-avail:VERS-ADAPT??
                       │            ├yes-crane:OPER-EFFC??
                       │            │          ├good(p):───PROBLEM
                       │            │          └good(c):───#CRANE
                       │            └no-crane:OPER-EFFC??
                       │                       ├good(p):───#PUMP
                       │                       └good(c):───#CRANE
                       └no-space:VERS-ADAPT??
                                 ├yes-crane:OPER-EFFC??
                                 │          ├good(p):───PROBLEM
                                 │          └good(c):───#CRANE
                                 └no-crane:───#PUMP
```

5. CONCLUSION

The emergence of new construction materials and technologies calls for the automation in several construction areas. As part of the construction automation, the knowledge-based decision support and expert systems have only been recently introduced to the construction industry. However, their application is particularly timely. Such systems may take years in the making. As more and more rules are updated and added to the existing knowledge bases, the accuracy and reliability of these systems could increase. They are fast becoming a necessity for an efficient management of construction projects.

REFERENCES

1. Hadipriono, F.C. and Chandra, S., 1987, "Value Engineering Rule-Based Expert System," Symposium of Computer Application in Civil Engineering Education, Indonesia.

2. Hadipriono, F.C. and Chang, K.S., 1988, "Performance Assessment of Construciton Site Operations Using Expert System," Proceedings of the 24th Annual Conference of the Associated Schools of Cosntruction, San Luis Obispo, California, pp 58-67.

3. Hadipriono, F.C. and Lim, C.L., 1987, "Rule-Based Diagnosis of Failure Events," Fourth International Symposium on Robotics and Artificial Intelligence in Building Construction," Haifa, Israel, pp 561-574.

4. Hadipriono, F.C. and Ross, T.J., "Towards a Rule-Based Expert System for Damage Assessment of Protective Structures," Second International Fuzzy Set Association Congress, Tokyo, Japan, pp 156-159.

5. Hadipriono, F.C., Sierakowski, R.L., and Lin, C.S., 1988, "Concrete Operation Decision Support Systems (CODSS)," Fifth International Symposium on Robotics in Construction, Tokyo, Japan, pp 491-499.

6. Ross, T.J. and Hadipriono, F.C., "Fuzzy Production Rules for A Damage Assessment Expert System," Proceedings of the International Sysmposium on Fuzzy Systems and Knowledge Engineering, Guangzhou, China, pp 249-256.

7. Program In Motion (PIM), 1986, 1ST CLASS, Expert System Development Package.

Practical System for Type Selection of Bridge Crossing River

Système pratique pour le choix du type de ponts

Praktisches System zur Typenwahl einer Flussbrücke

Takayuki NISHIDO
Res. Eng.
Kawada Industries
Tokyo, Japan

Ken-ichi MAEDA
Dr. Eng.
Kawada Industries
Tokyo, Japan

Kunikatsu NOMURA
Chief Eng.
Kawada Industries
Tokyo, Japan

Takayuki Nishido, born in 1957, received his M. Eng. in 1982 at Hiroshima University. He is working at the Research Laboratory, Kawada Industries, Inc.

Ken-ichi Maeda, born in 1949, received his Dr. Eng. in 1984 at Osaka University. He is at present manager of the Research Laboratory, Kawada Industries, Inc.

Kanikatsu Nomura, born in 1941, received his B. E. in 1964 at Waseda University. He is at present director of the Design Division, Kawada Industries Inc.

SUMMARY
The authors have developed an expert system of type selection of superstructure and substructure which uses integrated knowledge of expert designers and rules from specifications in Japan as the knowledge base. The expert system adopts fuzzy set theory to express the ambiguity of design data, and utilizes online connection between workstation and host-computer to improve accuracy of substructure cost evaluation. In the conclusion, it is possible to select the appropriate type and to considerably reduce design work time by using this system.

RESUME
Les auteurs ont mis au point un système expert pour le choix de la superstructure et de l'infrastructure des ponts sur cours d'eau utilisant comme base de connaissance les acquis des concepteurs spécialisés et les réglement des normes japonaises. Ce système expert adopte la théorie du sous-ensemble flou pour exprimer l'ambiguité des données de dimensionnement et utilise une liaison directe entre les stations de travail et un ordinateur central afin d'augmenter la précision de l'évaluation des coûts de l'infrastructure. En conclusion, il est possible de sélectionner un type de structure approprié et de réduire considérablement le temps de travail apporté à la conception en utilisant ce système.

ZUSAMMENFASSUNG
Von den Autoren wurde ein Expertensystem zur Typenwahl des Ober- und Unterbaus entwikkelt, das sowohl auf das integrierte Wissen der Expertenentwickler als auch auf die japanischen Normenvorschriften als Kenntnisbasis gestützt ist. Das Expertensystem verwendet die Theorie der unscharfen Menge, um die Unbestimmtheit der Konstruktionsdaten auszudrücken und benutzt die Online-Verbindung zwischen Arbeitsplatz und Host-Rechner, um die Genauigkeit der Unterstruktur-Kostenauswertung zu verbessern. Mit diesem System können ein geeigneter Typ gewählt und die Entwurfskosten erheblich gesenkt werden.

1. INTRODUCTION

Selection of a bridge superstructure and substructure is very important in the process from its design through to erection. Unless the designer has many years of experience and wealth of knowledge, it is difficult to select a type that is economically and serviceably satisfying. Moreover, design work time as well as efficiency of the selection and construction cost evaluation varies greatly depending on the experience of the designer. If designers use the integrated knowledge of expert designers, they can expect many merits. Hence, the authors have developed a system for basic design scheme of river crossing bridges as a practical application of the expert system.

Design conditions for superstructure and substructure of a bridge vary greatly depending on whether the bridge is to be located in a mountain area, urban region, across a river, and so on. A system applicable to all locations can become bulky and difficult to develop. On the other hand, river crossing bridges account for about half of all bridges constructed in Japan each year. The authors, therefore, limited the scope of the system to river crossing bridge. For river crossing bridges, the designer must take account of the River-Crossing Structure Law in Japan, making it especially difficult to decide a satisfactory span arrangement in accordance to the Law.

This system uses an expert shell (KEE) for easy development of the knowledge base and Lisp programming language. The development of this system took about one year by two expert designers from the design division and two knowledge engineers from research divison. The designer can easily input the data using keyboard and mouse while interacting with the system. Namely, the designer can understand what data should be input by watching the monitor. Presently, the designer can only use Japanese with this system in the actual design work, but an English version of the user-interface is under development, for design scheme of river crossing bridges in foreign projects, and educational purpose for foreign students studying in Japan.

The knowledge base used in the inference and selection process of this system are from standard rules given in the Specification for Highway bridges in Japan, some manuals, conventional rules based on the knowledge of expert designers, and the above-mentioned River-Crossing Structure Law. These conventional rules was created by trial and error in discussion manner between with knowledge engineers and expert designers. This system can be utilized with foreign specifications by replacing the Japanese specification with others.

Cost evaluation of superstructure and substructure can be easily calculated with this system. Not only the total construction cost of superstructure and substructure but separate construction costs, for example, fabrication, painting and transport for steel bridges, abutment and pier can be calculated. Moreover this system evaluates economy, erection workability, maintenance and running comfort as total assessment. Although the data in the knowledge base to calculate construction cost are revised at intervals of several years in Japan, the revision of these data can be easily performed.

In particular, the pile type selection process adopts fuzzy set theory to express the ambiguity of design data. In calculating pile construction cost, the use of the charts in manuals may cause some calculation error. In this system, the workstation (Nihon UNISYS Explore 2) and host-computer (Nihon UNISYS series 2200) are connected online so that this system can infer the cost of pile based on the results of analysis by the host-computer, thereby, improving the accuracy of construction cost evaluation.

This paper describes the outline and feature of this expert system and discusses an application example of actual design plan for comparison.

2. APPLICATION EXAMPLE AND ITS CONSIDERATIONS

Fig. 1 shows the type selection procedure for this system. Fig. 2 shows the input data with results obtained from this system. In which, span arrangement and bridge type combination obtained from the actual design plan under the same design condition are shown for comparison. In this case, the system provided 30 combinations (12 PC bridges, 18 steel bridges).

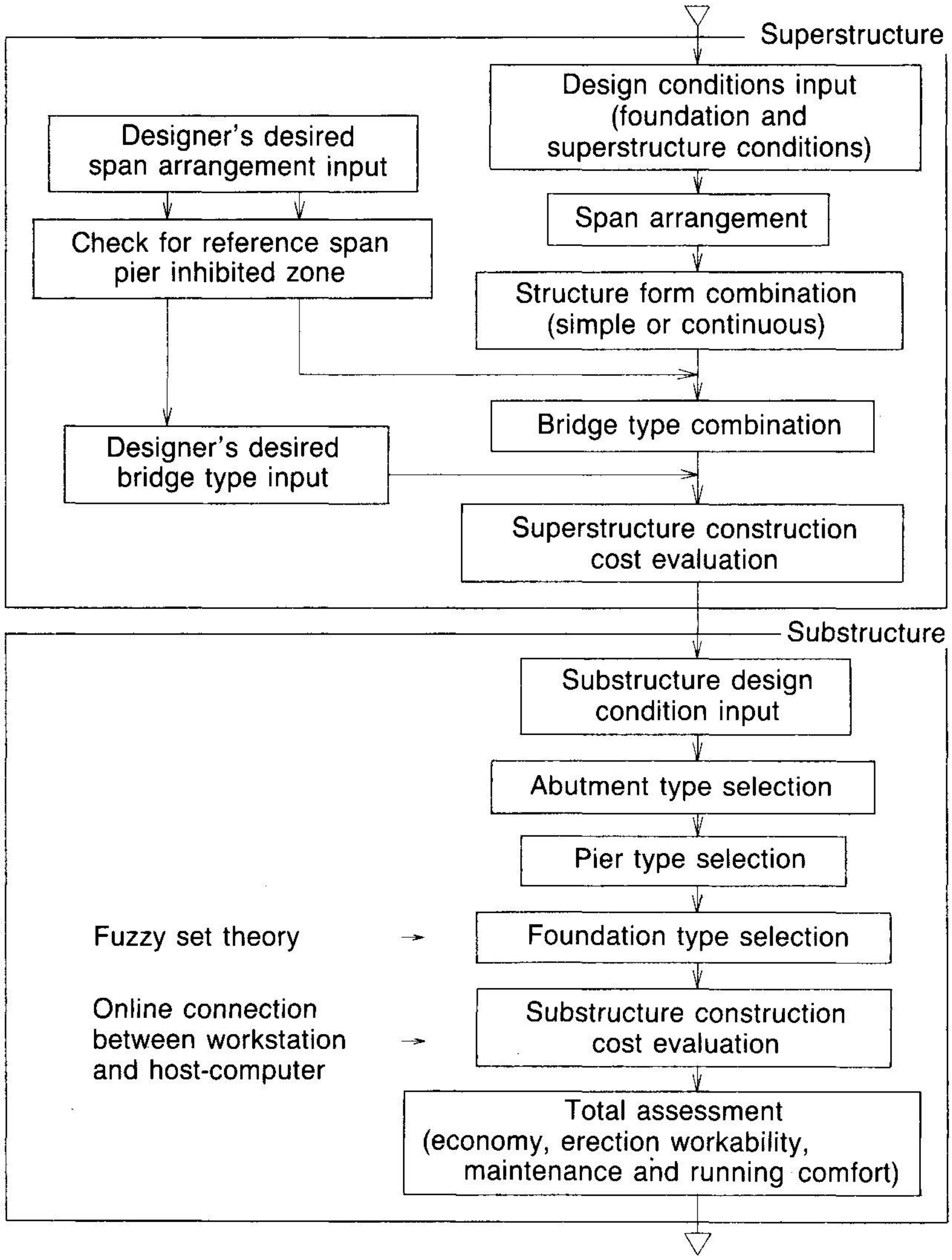


Fig. 1 Flow-chart of type selection procedure

Input data

BASIC DATA

Bridge length (m) : 307
River width (m) : 300
Skew angle (deg) : 80
Overall bridge width (m) : 10.75
Planned high water level (m) : 2
Hindrance to river improvement : No
River discahrge (m^3/s) : 4350
Storm tide area : No
Backwater area : No
. . . .
. . . .

SUPERSTRUCTURE DATA

Effective bridge width (m) : 9.75
Restruction on girder hight : No consider
Steel composite girder : Not used
. . . .
. . . .

SUBSTRUCTURE DATA

Horizontal seismic coefficient : 0.2
N-value of high river bed (left) : approx. 30
N-value of low river bed : approx. 30
N-value of high river bed (right) : approx. 30
Stage of gravel (mm) : from 5 to 10
Water depth at foundation work depth (m) : approx. 4
Foundation work depth (m) : approx. 15
Vertical Load (span length (m)) : approx. 60
. . . .
. . . .

An example of results

① Actual design plan

Number of span	2	3	2
Span length (m)	2 @ 30.6 = 61.2	3 @ 60 = 180	2 @ 32.9 = 65.8
Bridge type	Continuous noncomposite steel I-girder	Continuous noncomposite steel box-girder	Continuous noncomposite steel I-girder

② This system

Number of span	2	3	2
Span length (m)	2 @ 30.4 = 60.8	3 @ 60.333 = 181	2 @ 32.6 = 65.2
Bridge type	Continuous noncomposite steel I-girder	Continuous noncomposite steel box-girder	Continuous noncomposite steel I-girder

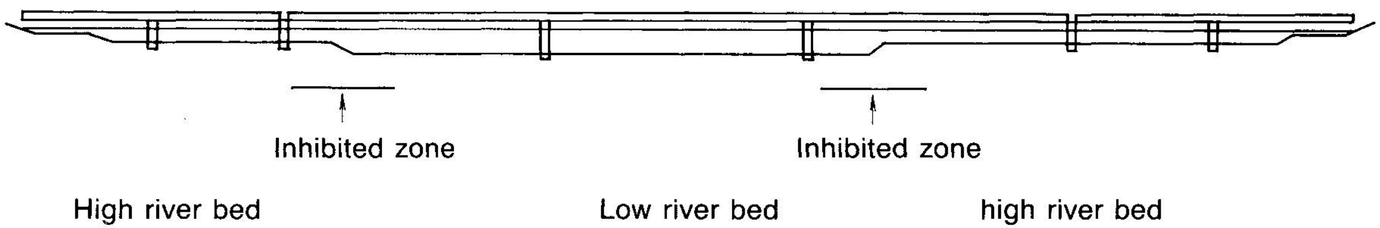


Fig. 2 Design conditions of actual plan and an example of results by this sytem

2.1 Type Selection of Superstructure

2.1.1 Span Arrangement

On completion of inputting data needed for type selection of superstructure, for example, topography, river width, river discharge, overall bridge width, bridge length, etc., the regulation of the River-Crossing Structure Law [1] (reference span length, pier inhibited zone, 5m relaxation regulation, exception to high river bed) are applied, and span arrangement is determined. These regulations are defined below and are used in the form of production rules in this system.

① Reference span length (minimum span length)
   This is to prevent any disturbance to the river flow caused by drifting objects such as trees due to floods.

② Pier inhibited zone
   This is to protect from anomalous scour around pier and obstruction to cross-selectional area of the river. So, pier inhibited zone is defined.

③ 5m relaxation regulation
   This is to relieve the condition that span length is much longer than reference span length. If the mean value of each length of all spans exceeds reference span length plus 5m, each span length can be made equal to or longer than reference span length minus 5m (in the case, the minimum reference span length is 30m).

④ Exception to high river bed
   When side span is located in the high river bed, if span length in the low river bed can be made longer, and if side span length can be made shorter, girder heights can be made lower. Thereby, raising of access road can be decreased.

Fig. 3 shows the rule-base of 5m relaxation regulation written in Japanese language.

All span arrangements obtained by the process above are displayed on the monitor screen. And the designer can remove undesirable span arrangements on his own judgement. The structural form (simple or continuous beam) with respect to span arrangements is determined. If the designer is not satisfied with any of the span arrangements proposed, this system allows the designer to input his desired span arrangement.

As shown in Fig. 2, these results from this system are approximately same span arrangements as the actual design plan. In actual design plan, expert designer moves pier minutely on his own judgement, taking into account appearance and other factors, for example, 1m or 1.5m off the boundary of pier inhibited zone. This problem can be solved by referring to the span arrangement results from this system and reinputting the span arrangement prefer by the designer.

基準径間長	: Reference span length	基本条件	: Basic conditions
上部工定数	: Superstructure constants	最大径間数	: Maximum number of spans
橋長	: Bridge length		

```
(IF (AND (AND (THE 基準径間長 OF 上部工定数 IS ? 基準径間長)
              (THE 橋長 OF 基本条件 IS ? 橋長)
              (THE 最大径間数 OF 上部工定数 IS ? 最大径間数))
         (AND (>= (/ ? 橋長  ? 最大径間数)  (+ ? 基準径間長 5 ))
              (OR (AND (>= (- ? 基準径間長 5 ) 30)
                       (>= (/  ? 橋長 (+ ? 最大径間数 I ))  (-基準径間長 5 )))
                  (AND (< (- ? 基準径間長 5 ) 30)
                       (>= (/  ? 橋長 (+ ? 最大径間数 I ) 30.0))))))
DO (PUT. VALUE  '上部工定数 '最大径間数 (+ ? 最大径間数 I )))
```

Fig. 3 Rule-base of 5m relaxation regulation written in Japanese language

2.1.2 Combination of Bridge Type

A bridge type associated with each span arrangement is selected. Fig. 4 shows bridge types used in this system. First, applicable bridge types are assigned to a given span arrangement using design manuals [2], [3]. Combinations of bridge types are decreased with structural restrictions and heuristic rules of expert designers.

This system uses about 50 rules (about 150 rules in the entire system) to decrease the number of combinations of bridge types to about one-fifth. Some of these rules for bridge type selection are given as follows.

Structural restrictions
① Different bridge types (Steel bridge, PC bridge, Preflection beam bridge) cannot be used together.
② Simple composite box-girder and simple noncomposite box-girder cannot be combined.
③ Simple steel deck I-girder and simple steel deck box-girder cannot be combined.

Heuristic rules of expert designers
① Combining four or more bridge types is not feasible.
② When spans are arranged symmetrically, bridge types must also be arranged symmetrically.
③ Span length of box-girder bridge type must be longer than span length of I-girder bridge type.

As in the case of span arrangement, there is a routine that displays on the monitor all selected bridge types combinations, to allow the designer to further decrease them by eliminating inappropriate bridge types. Another routine allows the designer to select the desired bridge type by using menu.

As Fig. 2 shown, the combination of bridge types in this system is identical with the actual design plan.

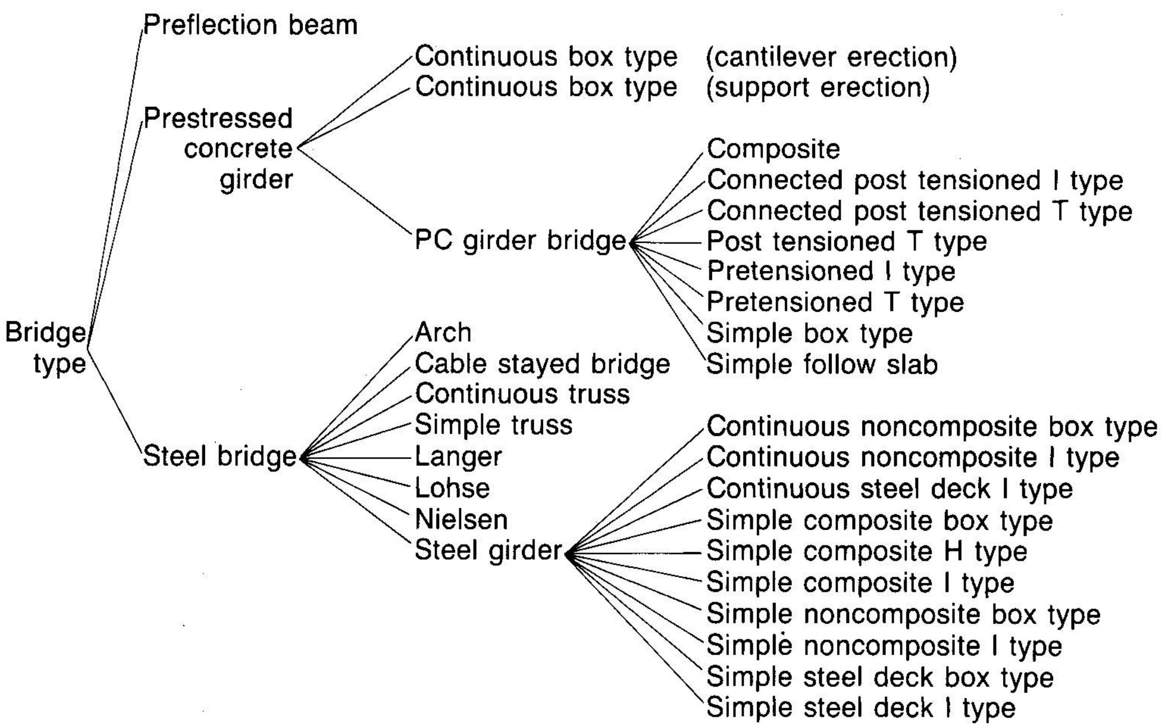


Fig. 4 Bridge types used in this system

2.1.3 Evaluation of Construction Cost

After selecting span arrangements and bridge types, the construction cost of superstructure is calculated for individual combination. The construction cost is calculated using charts information of design manuals [2], [3], [4].

Updating the knowledge base data to calculate construction cost of superstructure can be easily done by the designer in two ways as follows.
① Inputting the rise rate of construction cost for each bridge type.
② Reinputting the data for computerized charts of construction cost which are usually straight lines.

The evaluation is discussed in 2.2.3 with the construction cost of substructure.

2.2 Type Selection of Substructure

2.2.1 Type of Abutment and Pier

After the designer inputs data of the abutment shape, the system selects the type of abutment and pier according to table 1 that refers to the Design Data Book [2].

Using this table, the type of abutment and pier in this system is identical with the actual design plan.

2.2.2 Type of Foundation and Pile

Fig. 5 shows foundation types used in this system. The foundation type is usually selected by expert designer according to the foundation type selection table of the Specification for Highway Bridges [5].

However, the results of soil test are mere mean values of foundation work place, and these values have ambiguity. To give an example of "foundation work depth" in this table, when "foundation work depth" is 15 ~ 25m, sinso pile type is used frequently. On the other hand, when "foundation work depth" is 25 ~ 40m, sinso pile type is rarely used. If the result of soil test shows that "foundation work depth" is 25m without considering this ambiguity, selected pile type may be different from what expert designer will select.

Therefore, in this system, the authors applied fuzzy sets for pile type selection. As a result, a 1,200mm reverse pile is selected in this system. And this agrees with the pile type selected in the actual design plan. Thus, it seems that using fuzzy sets for pile selection is sufficiently practical.

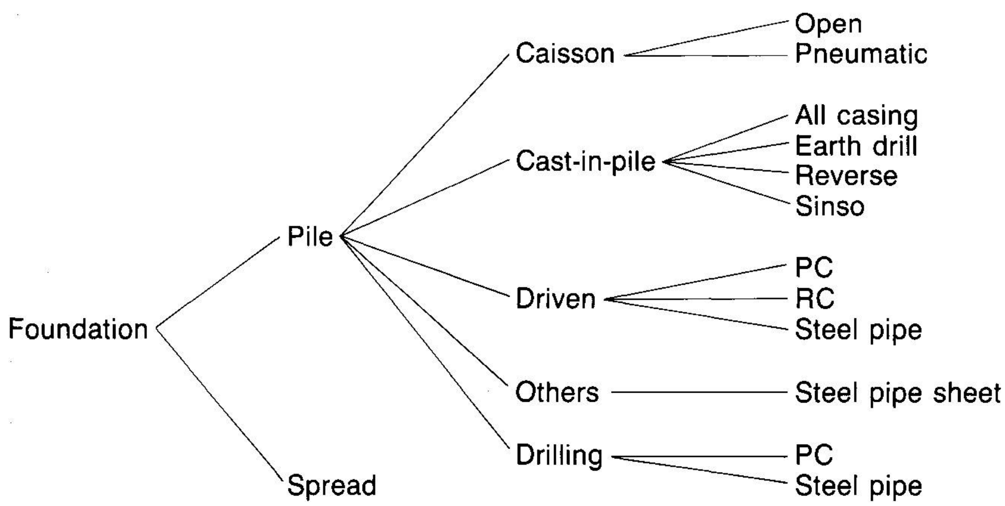


Fig. 5 Foundation types used in this system

2.2.3 Evaluation of Construction Cost

On completing type selection for abutment, pier and pile, the construction cost of substructure is calculated according to charts information of the Steel Bridge Design Planning Manual [3].

In calculating pile construction cost, the use of the charts in the Steel Bridge Design Planning Manual may cause some calculation error. In this system, therefore, the workstation and host-computer are connected online, and the pile construction cost is calculated based on the results of structural analysis by the host-computer.

Based on the result of pile type selection, the top three pile types are selected, and these pile diameters are changed. Table 2 shows all the possible pile types with the respect diameters. The stability and footing dimensions for abutments and piers and the required number of piles are calculated for each combination of pile type and diameter, using an existing structural analysis program at the host-computer. The input data for structural analysis are automatically created in the workstation and sent to the host-computer, then using the results of the analysis, each combination of pile construction costs is calculated as follows.

Pile construction cost =

the number of piles × pile length × pile construction unit cost

Finally, the pile type is selected as the most economical one. Similarly, the excavation and cofferdam cost can be calculated.

Updating the knowledge base data to calculate construction cost of substructure can be easily done by the designer in two ways as follows.

① Inputting the rise rate of construction cost for each abutment, pier and pile

② Reinputting the data for computerized charts of construction cost which are usually cubic curved lines.

Table 3 shows a comparison between the construction cost of superstructure and substructure and total construction cost obtained from this system and the actual design plan (Fig. 2). From these results, construction cost of superstructure by using charts information of manuals, and construction cost of substructure by using the host-computer are sufficiently accurate.

Table 1 Abutment and pier type and economical height

Abutment type	Economical height (m)
Gravity type	$h \leqq 4m$
Inverted T-type	$4m < h \leqq 12m$
Buttressed type	$12m < h$

Pier type	Economical height (m)
Wall type	$h \leqq 8m$
Inverted T-type	$8m < h \leqq 15m$
Column type	$15m < h$

Table 2 Pile diameters used in calculation by host-computer

Pile type	Diameter (mm)
Driven PC Driven RC Driven steel pipe Drilling PC Drilling steel pipe	600,800
Reverse All casing Earth drill Sinso	1000 1200 1500

Table 3 Comparison of construction cost between this system and actual design plan

Calculation method \ Details of construction cost	Superstructure (million yen)	Substructure (million yen)	Total construction cost (million yen)
① Actual design plan	597.0	264.0	861.0
② This system	628.5	270.0	898.6
(|① − ②| / ①) × 100	5.3%	2.3%	4.4%

(1 U.S. dollar :approx. 130 yen)

2.3 Total Assessment of type Selection Design Work Time

The total construction cost of the bridge can be obtained by summing up the costs of superstructure and substructure. The case with a lowest total construction cost is evaluated as "first rank economy". All cases are ranked according to total construction cost.

The system also displays the following message on erection workability, maintenance and running comfort.

① Erection workability: "good", "medium" and "bad"
This is evaluated according to the type of superstructure and substructure geological condition, area available for construction and the circumstances (urban or suburb).

② Maintenance: "repainting needed" for steel bridges and "pay attention to anticorrosion" for PC bridges.
In recent years, PC bridges are also affected by salt damage in Japan. So, this is evaluated according to whether construction place is near the sea.

③ Running comfort: "excellent", "medium" and "poor"
This is evaluated based on psychological effects and vibration that driver feels according to the number of joints, construction place and the stiffness of bridge type.

The most suitable selection (Fig. 2) based on total assessment is identical with the actual design plan. And design work time can be reduced to about half.

3. CONCLUSION

(1) The system has made it possible to obtain results that are almost equivalent to that of expert designers due to the integrated knowledge of the expert system, the Specification for Highway Bridges and the River-Crossing Structure Law.
(2) The overall result based on economy, erection workability and running comfort are sufficiently reliable. Consequently, the use of this system can greatly reduce the working time and labor required for the comparative design.
(3) Since a routine is added that allows a designer to select his desired span arrangements and bridge type combination, the system does not make automatic type selections, but can reflect the designer's ideas about type selection. As a result, the system is more practical.
(4) By using fuzzy sets which expresses the ambiguity of design data, the pile type can be selected by the process based on theoretical grounds. The result obtained through this process is sufficiently practical.
(5) By online connection between the host-computer and the workstation, the reliability of pile type selection and the accuracy of substructure construction cost evaluation can be improved.

ACKNOWLEDGEMENT

The authors would like to express the deepest appreciation to Prof. Shizuo Shimada of Nagoya University for his helpful advice. Also, thanks are expressed to Mr. Kazuhiro Hashimoto, Nihon UNISYS, Ltd. for his technical advice and suggestions through developing this system.

REFERENCES

1. Japan River Association: The River-Crossing Structure Law, pp.275~326, Apr. 1974 (in Japanese).
2. Japan Association of Steel Bridge Construction: The Design Data Book, July 1987 (in Japanese).
3. Japan Society of Steel Construction: The Steel Bridge Design Planning Manual, 1985 (in Japanese).
4. Japan Prestressed Concrete Contractors Association: The PC Road Bridge Construction Cost, March 1988 (in Japanese).
5. Japan road Association: The Specification for Highway Bridges — Substructure, pp.166~167, 1980 (in Japanese).

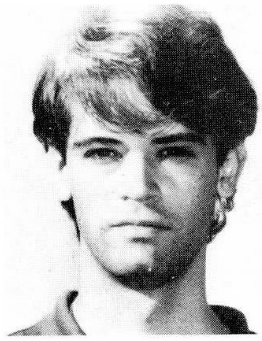
Leere Seite
Blank page
Page vide

Mixing Structural Simulation Models with Expert Systems

Utilisation conjointe des modèles de simulation structurale et des systèmes experts

Verknüpfung von strukturierten Simulationsmodellen und Expertensystemen

JP. LABAT
ILOG
Gentilly - France

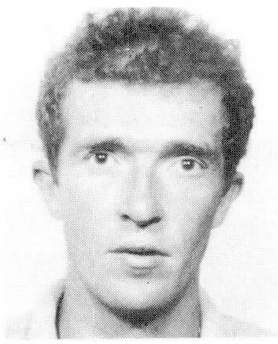


J.P. Labat, born in 1962, received his Civil Engineering Degree at the Ecole Nationale de Physique in 1986 and his MS in Computer Sciences at the Ecole Nationale Superieure de l'Aéronautique et de l'Espace in 1987. Since 1987, he is working on the development of ERASME.

O. CORBY
INRIA,
SOPHIA ANTIPOLIS
Valbonne - France

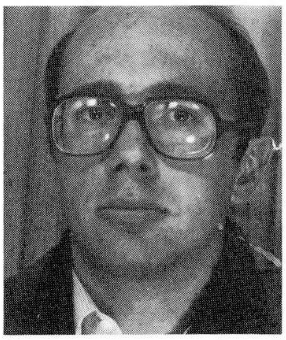
O. Corby, born in 1962, received his MS and PHD in Computer Sciences at the University of NICE. He is now working is a research Assistant at INRIA in SOPHIA-ANTIPOLIS. He is co-author of SMECI, a frame based expert system SHELL.

F. ALLEZ
CETE Méditerranée
Les Milles - France



F. Allez, born in 1956, Polytechnique, Ecole Nationale des Ponts et Chaussées, was involved from 1983 to 1985 in a Software Engineering Project (CONCERTO), Since 1985, he is project chief of ERASME.

P. HAREN
ILOG
Gentilly - France



P. Haren, born in 1953, Polytechnique, Ecole Nationale des Ponts et Chaussées, received his MS and PHD at the Massachusetts Institute of Technology (1980). From 1983 to 1987, he was project chief of SMECI at INRIA in SOPHIA ANTIPOLIS. He is now General Manager of ILOG, an Artificial Intelligence Company.

ABSTRACT
ERASME is a multi expert system for pavement defect diagnosis and rehabilitation, which is interfaced with four structural simulation models. At first, we present the project objectives. The concepts we use to encode knowledge are shown in the second section: blackboard-like architecture, multiple reasoning, multi expert systems. The third section concerns the knowledge representation: associational and causal knowledge, Generate & Test & Debug Paradigm. At last, an example of solving process is proposed in the fourth section.

RESUME
ERASME est un système expert multiple, pour le diagnostique des défauts des chaussées et leurs réparations, qui est interfacé avec quatre modèles de simulation structurale. Nous présentons d'abord les objectifs du projet. Les concepts que nous utilisons pour le codage de la connaissance sont montrés dans la deuxième partie: architecture type «blackboard», raisonnement multiple, systèmes experts multiples. La troisième partie concerne la représentation de la connaissance: connaissance associative et causale, génération & test & correction. Enfin, un exemple d'un processus de résolution est proposé dans la quatrième partie.

ZUSAMMENFASSUNG
ERASME ist ein vielfältiges Expertensystem für die Diagnose von Defekten bei Strassenbelägen und deren Reparaturen. Das Expertensystem ist mit vier Simulationsmodellen verknüpft. Zuerst wird das Projekt beschrieben. Das Konzept für die Kodierung der Grundlagen ist im zweiten Teil aufgezeigt: Aufbautyp «black board», vielschichtige Grundlagen, vielfältige Expertensysteme. Der dritte Teil behandelt die Darstellung des Wissens: verbundenes und ursächliches Wissen, Entstehen & Testen & Korrigieren. Im vierten Teil wird ein Beispiel einer Lösungsfindung beschrieben.

1 ERASME OBJECTIVES

ERASME is a three year old project for building a multi expert system for highway rehabilitation.

1.1 Solving Process

Following HALL [3], evaluation of a pavement and development of feasible rehabilitation alternatives is perfomed according to the following steps :

1. Evaluation of present condition,
2. Construction of different pavement assessments,
3. Prediction of future condition without rehabilitation,
4. Selection of rehabilitation approach,
5. Prediction of each rehabilitation approach,
6. Cost analysis of each rehabilitation approach,
7. Physical testings as needed.

Those steps are performed along diagnosis, prediction and design stages.

1.2 A user assessment

Before or while developing an expert system, it is important to pay attention to the expected user ! ERASME should be available to decision makers in the field of pavement rehabilitation at the regional services level. Our average user manages 3000 kilometers of minor roads. He analyses 300 kilometers each year. That leads to about 30 worksites. He spends a 50 million francs budget (that is approximately 7,5 million dollars). Using ERASME he should save at least 2.5 % on his budget. In FRANCE, ERASME should have about one hundred such users.

1.3 Diagnoses services

The user must be able to get diagnosis information about a particular section that worries him. He can either submit his case of interest to the generalist or make use of the skills of the specialists.

In the first case, the generalist will take care of the problem. He will call on adequate specialists to treat the problem.

In the second case, only the selected specialist skills will be called for. The expert system will focus its attention on the user's particular point of interest.

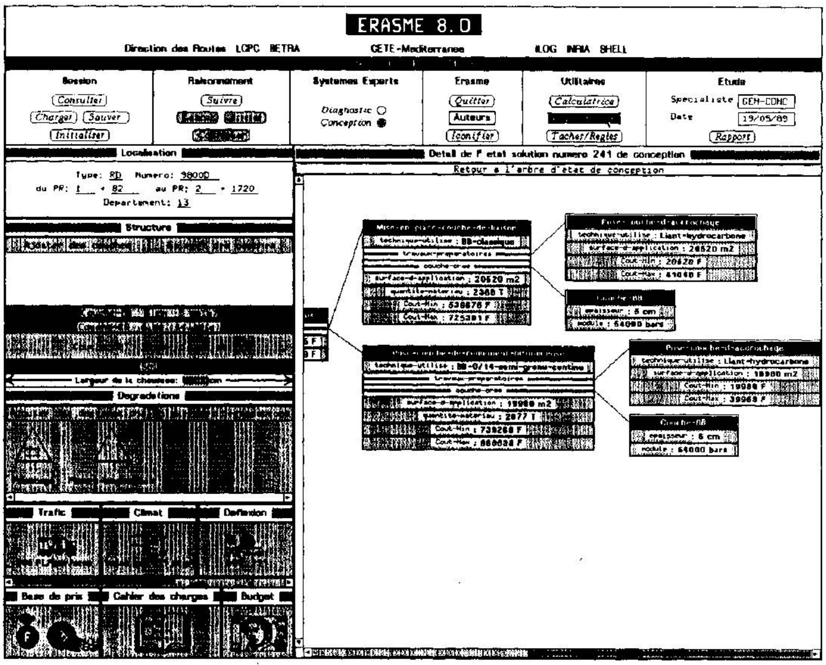
The specialist " pavement frost resistance" can either calculate a roadblock due to icy roads or evaluate past frost damages on the pavement.

1.4 Design services

Design follows a wholescale analysis, that is a diagnosis undertaken by the generalist expert for diagnosis.

Before actual design, the main specifications are drawn up by the user. These specifcations are expressed in terms of life service, surface course adherence, life time ...They constitue the requirement.

Generally, ERASME will propose several successful rehabilitaion techniques. For every proposed solution, a life-cost analysis will be performed.



1.5 Prediction services

Following HALL [3], we think that a pavement evaluation system which can only identify current rehabilitation needs has limited usefulness as a pavement management tool. In order to assist decision makers, the expert system must be able to predict pavement evolution in case of no rehabilitation.

This facility should enable pavement managers to assess the consequences of a work report.

1.6 Incomplete Data

Available information is sometimes scarce, in particular for low traffic roads (laboratory tests such as deflection or in-situ material tests). The pavement manager would like to know which laboratory tests he should require in order to assess pavement state and choose a reliable and cost-effective rehabilitation technique. When information is lacking, the system will propose several concurrent diagnosis and associated rehabilitation techniques. In a second stage, it will indicate the laboratory tests that would reduce the number of these concurrent diagnoses.

1.7 User Interface

We made much effort to create a user friendly interface featuring icons, mouse, windows and various editors.

2. THE SYSTEM ARCHITECTURE

2.1 The Multi Specialist Kernel

As the number of human experts involved in the projet is about twenty, a multi specialist architecture has been selected in order to produce a modular software. ERASME is in fact built on the model of blackboards [4]. It is a collection of simple cooperating knowledge bases, called specialists, where each one embodies specialized knowledge such as : frost resistance, asphalt concrete, struture adequation toward traffic, etc... It enables modular knowledge formalization and modular encoding.

As the system is developed by several persons (currently four), the software engineering modularity concept is of great interest. It enables easy internal modifications and greatly facilitates debugging. Furthermore, an incomplete system can be tested.

Whereas operational competence is distributed among specialists, structural knowledge is global and shared by all of them.

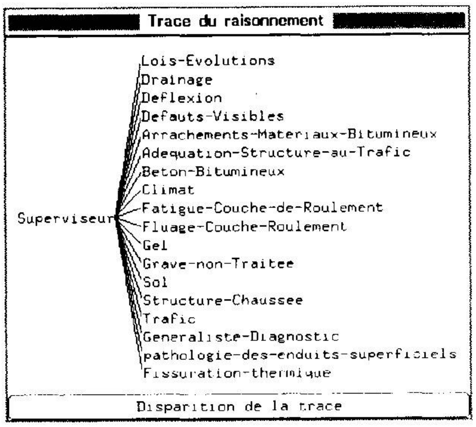


Figure 1 The Supervisor and its specialists

2.2 Structural Knowledge

Concepts involved in pavement diagnosis and rehabilitation are represented by SMECI [6] frames, including classes and instances. The global data base is a collection of such instances calleds *objects*.

The value of an object's slot may be either an integer, a real, a string, an object or a list of such values. Slot value may also be constrained by an interval or a list of possible values. Slot value may also be constrained by interval or a list of possible values. As the value of a slot may be another object, instances may be connected through slots values and form a net.

A class defines the structure of a family of objects in terms of slots. Classes are refined by standard subclass trees which specify default values, range constraints and specific methods. A class inherits methods, values and contraints from its ancestor, unless it redefines them.

Structural knowledge includes such classes as : *Pavement, degradation, traffic etc...* The following figures show the icons associated to the Degradation classes.

Deductions are carried out by production rules whose premises and actions operate on objects. The system records its deductions within slot values and new objects.

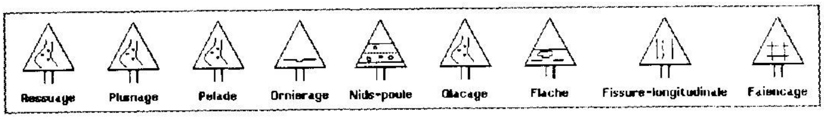


Figure 2 : Icons associated to Degradation classes

2.3 Multiple Reasoning

The SMECI shell provides multiple states that are similar to ART viewpoints [1] and KEE multiple worlds [2]. However, SMECI states cannot be merged.

An expert system programmed in SMECI states starts its reasoning from an initial state. Rules of current rule base generate states that are *sons* of the current state. If a rule has several instantiations, it produces one state per instantiation. If several rules fire, each one produces its own state. In order to prevent combinatorial explosion, it is possible to specify, for each rule base, the maximum number of applicable rules. It is also possible to prune the tree by mean of contradiction rules.

The next figure shows a state tree produced by ERASME :

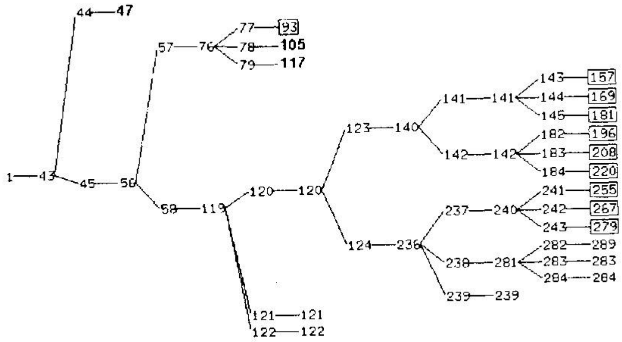


Figure 3: A state tree

2.4 Reasoning upon Reasoning

It is possible to have several expert systems in the same SMECI environment and make them work together.

In SMECI, an expert system is an object, instance of a system class called *Expert System*. Each expert system has its own knoledge base (classes, rules, methods) and data base and derives its own reasoning tree.

In order to construct its own reasoning tree, one expert system can look over the resuls previously attained by its colleagues. That feature is called *Reasoning upon reasoning.*

At present time, ERASME is a collection of two expert systems. The first one is in charge of pavement assessment or diagnosis. The second one is able to design rehabilitation techniques associated to previously attained diagnoses.

3. KNOWLEDGE REPRESENTATION

The following concerns the diagnosis expert system of ERASME.

3.1 Structural Simulation Models

In the field of pavement Engineering, knowledge is generally associated to causal models[1].

Often, causal models are implemented in terms of **Structural Simulation Models**.

Causal models can be used to predict the behaviour of a *known object*.

ERASME is interfaced with four structural Simulation Models : Gel (frost resistance), Alize (structural analysis), Ornier (asphalt concrete fow), Fistherm (asphalt concrete thermic cracking)[2].

3.2 Associational and Causal Knowledge

When causal models are available, pavement engineers use two types of knowledge :

1. **Causal knowledge** which map causes to effects,

2. **Associational Knowledge** which map effects to causes.

Associational rules could be automatically derived from causal rules by simply reindexing the later [5]. This would lead to the setting up of a huge number of associational rules.

In fact, experts use only a few associational rules which derives from their own experience. Associational rules encode two important abstractions of the causal domain models [5] :

. encapsulation of interactions,

. encoding of problem solving knowledge.

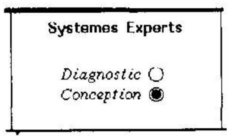


Figure 4: Expert System Selector

[1] However, some of pavement behaviors remain unclear or unknown : *unbound materials flow,soil behaviour,etc.*

[2] **Gel, Alize and Fistherm** belong to the Laboratoire Central des Ponts et chaussées, PARIS, FRANCE and **Ornier** to the Shell Compagny

3.3 Cooperation betwen Associational and Causal Knowledge

The Generate & Test & Debug paradign was published by MR. SIMMONS [5].

3.3.1 Generate, Test and Debug

The diagnosis expert system of ERASME has three main stages in its solving process :

1. **Generate**, it builds a model of the pavement according to some reasonable hypotheses set up by itself,

2. **Test**, it simulates the behaviour of the pavement given the proposed model in order to determine the validity of the hypothesis. If the test is successful, the hypotheses are accepted. Otherwise, the last stage is undertaken.

3. **Debug**, given the results of the testing stage, it emits suspicions in order to modify some of the previously defined hypotheses.

The system use **associational rules** to set up reasonable hypotheses and emits suspicions upon previously defined hypotheses.

```
DEBRULE      hypothesize-binder-class

LET          pavement a Pavement
             surface -course a Layer AMONG 1^Layers^pavement

IF           class^binder^asphalt^surface-course = () AND
             geographic-area^pavement = south-of-France

THEN ACTION
                 $ (hypothesize class^binder^asphalt^surface-course '40/50)

ENDRULE
```

Figure 5 : An hypothesizing rule

ERASME uses causal knowledge in the testing stage, including the **four Structural Simulation Models** it is interfaced with.

Specialists declare at the beginning of the reasoning process which hypotheses they are concerned with, in such a way that the supervisor may trigger them again when hypotheses are updated.

Such a triggering leads the specialist to carry out again its reasoning process, according to new hypotheses values.

3.3.2 Suspicions

In the *Debug* phase, the system undertakes a reasoning process with the following steps :

1. some of its specialists express *suspicions* on certain hypotheses according to some *suspecting rule*,

2. it generates one line of reasoning per suspicion,

3. each suspicion is sent to the competent specialist which

   . modifies some hypotheses,

   . or generates another suspicion,

given the current suspicion and according to some *debugging rule*,

4. it reprocesses some of its reasoning process.

```
DEBRULE      suspect-granulate-high-dosage

LET          pavement a Pavement
                 tear-out a Degradations of prototype tear-out
                     among degradations^pavement
                 coat a Material of prototype Surface-coat
                     among material^1^layers^pavement
                 granulate a Granulate among granulates^coat

IF           appearance^tear-out = first-winter AND
                 spot^granulate = no AND
                 modality^tear-out = generalised

THEN ACTION
                 $ (suspect granulate 'dosage 'high)

ENDRULE
```

Figure 6 : A suspecting rule

Suspicions leads to the construction of alternative worlds. Each leaf node of the diagnosis state tree describes a model that accounts for the real pavement.

4. EXAMPLE OF ERASME

This section shows an example of ERASME diagnosis expert system utilization. Let's suppose that the user consults ERASME by means of the supervisor which will act as a generalist expert calling specialists.

4.1 Request on an Example

The supervisor emits the first request :

R1 : visible-defects of pavement

It is routed by the supervisor to the Visible Defects Analysis specialist (VDA) which defines the surface state of the pavement. VDA emits a request :

R2 : definition of structure

The VDA specialist is then interrupted to let the Structure specialist (ST) answer *R2*. ST initializes the pavement structure.

After the two specialists reasoning, the supervisor is in possession of general data allowing it to carry out a diagnosis.

Suppose that the surface state presents a significant rut. The supervisor decides to consult three specialists : Structure Adequation to Traffic (SAT), Wearing Course Fatigue (WCF) and Frost Resistance (FR).

It emits three requests in order to trigger the specialists:

R3 : adequation of structure,

R4 : degradation of wearing course,

R5 : frost damage of structure.

After specialists consultation and some Structural Simulation Programs execution, three situations may happen :

1. The supervisor decides to stop reasoning. It evaluates the current state as a diagnosis because data are coherent. Every symptom has an identified cause, the diagnosis is archieved.

2. The supervisor detects incoherence. It may emit a suspicion on the value of an object slot and reprocess a part of previous reasoning.

3. The supervisor detects contradictions like in 2), but it is not able to emit any suspicion. The current state is tagged as contradictory and is abandoned.

4.2 Suspicions on an Example

We pursue the preceding example and let's suppose that the expert system produced only one line of reasoning. The last state contains the following important facts :

> *the structure is adequate with respect to traffic, there is no fatigue of wearing course, there is no frost damage of structure.*

Anyway, the current state is considered incoherent because the importance of the rut is high. Then the supervisor suspects the traffic is under evaluated.

The suspicion is transmitted to the Traffic specialist. It reprocesses its reasoning according to new data about traffic evaluation.

Some tasks contain rules that refer a suspicion in their premises as shown in figure 7.

```
LET     suspicion a Suspicion
If      slot^suspicion = evaluation AND
        object^suspicion = traffic AND
        value^suspicion = under-evaluated
THEN
        number-trucks^traffic = 3/2 * number-trucks^traffic
```

Figure 7 : A debugging rule

The new value of the slot *number-trucks* of traffic leads the supervisor to fire again SAT, WCF and FR because all of them declared that this slot was an hypothesis they were sensitive to.

Emitting a suspicion produces non monotonic reasoning by the mean of hypothesis dependency declaration.

5 CONCLUSIONS

Heterogenous Knownledge Pavement Engineering is a complex task which involves very different knowledge. One has to use several different schemes to encode Pavement Engineering knowledge.

Causal models Civil engineering is a domain where causal models represent a large part of existing knowledge. Causal models are sometimes available as structural simulation programs.

A lot of calculations For solving a particular problem, ERASME makes a lot of calculations (up to 150 executions).

REFERENCES

[1.] ART ART Programming Manual, tome 3 Inference Corp. LOS ANGELES.

[2.] FILMAN R. (1988) Reasoning with worlds and truth maintenance in a knowledge-based programming environnement. Communications of the ACM, April 88 : 31 : 4 : 382-401.

[3.] T. HALL, J. CONNOR, M. DARTER, S. CARPENTER DEVELOPPEMENT of an expert system for concrete pavement evaluation and rehabilitation. Second North american Conference on Managing Pavements, TORONTO 87.

[4.] NII. P. (1986) Blackboard systems : the blackboard model of problem solving and the evolution of blackboard architectures. AI magazine, Summer 1986 : 38-53. Blackboard system, blackboard application systems, blackboard systems from a knowledge engineering perpective. AI magazine, August 1986 : 82-106.

[5.] R. SIMMONS, R. DAVIS. Generate, Test and Debug : Combining Associational rules and Causal Models. IJICAI 1987 : 2 : 1075-1078.

[6.] SMECI Version 1.4 Reference Manual INRIA - ILOG, PARIS, FRANCE.