US Army Corps of Engineers
Construction Engineering Research Laboratory

American Society of Civil Engineers

USACERL CONFERENCE PROCEEDINGS P-89/10
January 1989

Proceedings: USACERL/ASCE First Joint Conference on Expert Systems, June 29-30, 1988

This conference was jointly organized by the U.S. Army Construction Engineering Research Laboratory (USACERL) and the American Society of Civil Engineers (ASCE). It was hosted by USACERL during June 29-30, 1988 in Champaign, Illinois.

The purpose of the conference was to exchange information on current research trends and developments in the area of expert systems. Some of the papers dealt with expert systems in general; others focused on how these systems relate to various aspects of civil engineering.

Due to the brevity of the conference, several worthy papers could not be included in the conference program. A section has been added to these Proceedings, containing extended abstracts that were not selected for presentation but recommended for publication by the reviewing committee.

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704 0188
Exp Date Jun 30 1986

1a REPORT SECURITY CLASSIFICATION: Unclassified

1b RESTRICTIVE MARKINGS

2a SECURITY CLASSIFICATION AUTHORITY

2b DECLASSIFICATION DOWNGRADING SCHEDULE

3 DISTRIBUTION/AVAILABILITY OF REPORT: Approved for public release; distribution is unlimited.

4 PERFORMING ORGANIZATION REPORT NUMBER(S): USACERL CONFERENCE PROCEEDINGS P-89/10

5 MONITORING ORGANIZATION REPORT NUMBER(S)

6a NAME OF PERFORMING ORGANIZATION: U.S. Army Construction Engr Research Laboratory

6b OFFICE SYMBOL (If applicable)

7a NAME OF MONITORING ORGANIZATION

6c ADDRESS (City, State, and ZIP Code): P.O. Box 4005 Champaign, IL 61824-4005

7b ADDRESS (City, State, and ZIP Code)

8a NAME OF FUNDING/SPONSORING ORGANIZATION: U.S. Army Constr Engr Research Laboratory

8b OFFICE SYMBOL (If applicable)

9 PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER

8c ADDRESS (City, State, and ZIP Code): P.O. Box 4005 Champaign, IL 61824-4005

10 SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO	PROJECT NO	TASK NO	WORK UNIT ACCESSION NO
4A161102	AT23		EN9

11 TITLE (Include Security Classification): Proceedings: USACERL/ASCE First Joint Conference on Expert Systems (U)

12 PERSONAL AUTHOR(S)

13a TYPE OF REPORT	13b TIME COVERED	14 DATE OF REPORT (Year, Month, Day)	15 PAGE COUNT
final	FROM ___ TO ___	1989, January	169

16 SUPPLEMENTARY NOTATION: Copies may be obtained from the National Technical Information Service Springfield, VA 22161

17 COSATI CODES

FIELD	GROUP	SUB GROUP
12	09	

18 SUBJECT TERMS (Continue on reverse if necessary and identify by block number): expert systems; symposia

19 ABSTRACT (Continue on reverse if necessary and identify by block number)

This conference was jointly organized by the U.S. Army Construction Engineering Research Laboratory (USACERL) and the American Society of Civil Engineers (ASCE). It was hosted by USACERL during June 29-30, 1988 in Champaign, Illinois.

The purpose of the conference was to exchange information on current research trends and developments in the area of expert systems. Some of the papers dealt with expert systems in general; others focused on how these systems relate to various aspects of civil engineering.

Due to the brevity of the conference, several worthy papers could not be included in the conference program. A section has been added to these Proceedings, containing extended abstracts that were not selected for presentation but recommended for publication by the reviewing committee.

20 DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☐ UNCLASSIFIED/UNLIMITED ☒ SAME AS RPT ☐ DTIC USERS

21 ABSTRACT SECURITY CLASSIFICATION: Unclassified

22a NAME OF RESPONSIBLE INDIVIDUAL: D. P. Mann

22b TELEPHONE (Include Area Code): (217) 373-7223

22c OFFICE SYMBOL: CECER-IMT

DD FORM 1473, 84 MAR — 83 APR edition may be used until exhausted. All other editions are obsolete.

SECURITY CLASSIFICATION OF THIS PAGE
UNCLASSIFIED

FOREWORD

These proceedings were prepared by the Facility Systems Division (FS) of the U.S. Army Construction Engineering Research Laboratory (USACERL). The work was performed under Project 4A161102AT23, "Basic Research in Military Construction"; Task A; Work Unit EN9, "A Physical Process Visualization Technique for Generating Networks."

The Conference was jointly organized by Dr. Simon Kim of USACERL-FS, Mr. Frank Kearney of the Engineering and Materials Divison (EM) of USACERL, and Dr. Lou Cohn, Chairman of the Expert Systems Committee of the American Society of Civil Engineers (ASCE). The Conference was hosted and the publication of the proceedings was coordinated by Dr. Simon Kim and Mr. Frank Kearney.

The assistance of Mr. Diego Echeverry, Mr. Thomas Gatton, and Dr. Francois Grobler of USACERL is gratefully acknowledged.

Dr. Michael O'Connor is Chief of USACERL-FS, and Dr. Robert Quattrone is Chief of USACERL-EM. COL Carl O. Magnell is Commander and Director of USACERL, and Dr. L. R. Shaffer is Technical Director.

CONTENTS

CONTENTS (Cont'd)

INTRODUCTION

The first joint USACERL/ASCE Conference on Expert Systems was conceived to provide a forum to communicate and discuss current research trends and developments in this area of intense activity. Many researchers participated, representing several research institutions all over the country, making this effort a very successful one.

The main objective of the conference was to provide for communication and exchange of ideas among participating researchers. Publication of these Proceedings is intended to make the conference materials accessible to the entire research community.

The variety of topics covered in the conference is a reflection of the flexibility and potential of knowledge based tools. Research work presented ranged from the support of facilities design to support of construction, operation, and maintenance.

The field of artificial intelligence and expert systems has been fertile ground for research activity at USACERL. For this reason, and because the conference was hosted at the USACERL facilities, many papers were submitted by USACERL researchers. However, the conference agenda allowed only one day of presentations, which restricted the number of papers accepted for presentation. In order to offer the reader a more comprehensive overview of the work being performed at USACERL, it was decided to add to these Proceedings a section containing those extended abstracts submitted by USACERL researchers and recommended for publication by the reviewing committee.

Finally, it is important to highlight the participation of the Expert Systems Committee of the ASCE in organizing this conference. It is also necessary to acknowledge the contribution of all the researchers who submitted papers. The quality of their work was the main reason for making this First Joint USACERL/ASCE Conference a truly successful one.

INDUCTIVE LEARNING IN ENGINEERING

T. Arciszewski

Intelligent Computers Center
Civil Engineering Department
Wayne State University
Detroit, Michigan 48202
(313) 577-3766

Extended Abstract

1. Introduction

This presentation discusses a new research area: inductive learning in engineering. Several engineering applications of inductive learning systems are described. In all these applications a new class of inductive tools is used, based on the rough sets theory. These systems were developed by W. Ziarko of the Computer Science Department, University of Regina, Canada. Our results are of a general character and provide a good insight into the potential of inductive systems in engineering applications. All these results were obtained in the Intelligent Computers Center, Civil Engineering Department, Wayne State University, as the result of cooperation between these two departments initiated in 1984.

An inductive system is understood here as a computer program using learning from examples as a basic component of engineering knowledge acquisition. It can be used in engineering for several different purposes. The best-known engineering application is as an expert systems building tool for the generation of decision rules from examples. A related application is inductive problem-solving: an extraction of decision rules from examples to find an unexpected rule or missing link which is necessary to solve a given complex engineering problem.

Inductive systems can also be used for the purposes of shallow modeling in engineering, where several different applications are possible when dealing with a group of examples described by a number of attributes. An inductive system can be used for the analysis of dependencies between individual attributes or groups of attributes, or in the analysis of the significance of individual attributes with respect to their contribution to the dependency between selected groups of attributes. It can be also used to reduce attributes to their minimal independent subset.

This presentation discusses individual applications of inductive systems using original examples from the area of structural engineering, and from a the diagnosis of the Space Shuttle Main Engine. Directions for future research are also discussed.

All inductive systems discussed here were built using a new mathematical concept referred to as the "theory of rough sets," recently proposed by Pawlak. This concept, considered as an alternative to the theory of fuzzy sets, provides solid data analysis and reasoning from such data. It has been used in applications ranging from medical diagnosis to industrial process control. The most impressive results have been obtained in applications requiring extensive data analysis, when traditional, statistical analytical methods could not be used. The theory of rough sets has been also utilized in the development of several learning algorithms.

2. Methodology of Inductive Learning

Inductive systems can be considered as new engineering tools which can be used for different purposes. The internal workings of these systems are a subject of interest to computer scientists. Engineers are interested in the rational use of such systems, and the methodology of their use in engineering is becoming very important.

Methodology is a subarea of inductive learning. Its subject is the process of generating decision rules from examples, and methods of control and optimization of this process in order to minimize the time required to extract set of decision rules from a given body of examples. The methodology of inductive learning considers the process of inductive learning from the system user's point of view. The user applies an inductive system as a kind of black box, and his understanding of the mathematical or computer aspects of learning is usually very shallow. So defined, the methodology of inductive learning will provide detailed methodological knowledge for a large class of potential inductive system users in engineering. Its development should also stimulate further applications of inductive systems, particularly in engineering.

The Author initiated research on the methodology of inductive learning several years ago. It had led to an initial understanding of this process, described in two papers. This work resulted in several different models of the inductive learning process in engineering. The selection of examples for individual learning process stages was analyzed and recommendations made. A system of control criteria was also proposed; these criteria can be used to monitor and control the learning process.

3. Extraction of Decision Rules from Examples

The author used inductive learning in engineering for the first time in 1985. The results obtained convinced him that an inductive system can be used as an effective tool for the generation of complex decision rules dealing with engineering problems. The first application was in the area of structural engineering. An inductive system was used to generate of decision rules to determinate the feasibility of different types of wind bracings in tall buildings, considered from the viewpoint of conceptual design.

In this section one engineering application of inductive systems is discussed. An inductive system was used to extract decision rules from examples for later use in diagnosis. This application concerns test data analysis and diagnosis of the High Pressure Oxidizer Turbo Pump (HPOT) in the Space Shuttle Main Engine.

The history of inductive learning in the Rocketdyne Division of Rockwell International is characteristic of artificial intelligence in industrial applications. It explains why inductive learning is so attractive in industrial applications and when its use is justified.

After each test firing of a Space Shuttle Main Engine (SSME), a large body of test data is generated and must be analyzed. The SSME must be diagnosed, and recommendations prepared for the next engine test. The SSME is the world's most complex liquid-fuel rocket engine. Its performance is crucial for the safety of the Space Shuttle and, particularly after the Challenger tragedy, both Rocketdyne management and NASA insist that a thorough investigation of each test firing be performed by Rocketdyne's most experienced and highly trained engineering staff. This staff has unique accumulated experience, gained during the last 13 years and covering more than 1400 SSME firings, as well as experience gained from earlier tests of the Apollo F-1 and the Atlas J-2 engines. There is, however, another important factor: a significant gap between staff with 20 to 30 years of experience and a younger generation of engineers with only 5 to 10 years of experience. All these most experienced engineers are approaching or retirement age, and their replacements have, at least for now, considerably less rocket engine experience.

In this situation, Rocketdyne was confronted with a difficult problem: how to improve the quality of SSME test analysis in the face of a diminishing senior staff. It has finally been decided to use a combination of staff, results from previous SSME tests, and automated tools, including inductive tools, to address the problem and to build a prototype for automated corporate expertise.

In 1984 Dr. K. L. Modesitt was hired to support the construction of an automated tool for SSME test analysis. The first project was to develop a proof-of-concept expert system for the analysis and diagnosis of test data for a High Pressure Oxidizer Turbopump (HPOTP) using an inductive tool for the generation of decision rules. Expert Ease, an inductive system developed by Intelligent Terminals, Ltd., was used in the first stage of this project. Later, a more advanced inductive system, ExTran 7, also by Intelligent Terminals, Ltd., was used. The initial expert system underwent subsequent several extensions and modifications, and it became an extensive software system, an Automated Test Data Expert, called Scotty. It is now at a stage between a research prototype and a production model; it is not in production yet, but is expected to be soon.

A mechanical engineer working at Rocketdyne identified nine attributes, including the conclusion, as sufficient to describe the performance of the HPOTP. A body of 42 examples was used to generate decision rules for the analysis and diagnosis of the HPOTP.

The same body of examples was also used for tests conducted by the author in the Intelligent Computers Center using ANLYST, an inductive system developed by Ziarko. The objective of these tests was to verify the attributes and decision rules generated by Rocketdyne. The tests proved that the attributes were selected properly. Also, the decision tree generated by authors is identical to that obtained by Rocketdyne. It was found that there is a very strong, functional relationship between the group of independent attributes and the dependent variable, conclusion. This relationship between two groups of attributes is measured in the theory of rough sets by the strength of the connection. In this case this parameter was equal to one: a functional relationship. Separate sets of decision rules were generated to support individual conclusions.

Individual independent attributes were considered from the point of view of their affect on the relationship between a group of independent attributes and a given conclusion. These effects are measured in the theory of rough sets by the significance factor. In the case of the first conclusion it was found that significance factors are strongly differentiated for individual independent attributes, and in three cases, for attributes one, three and seven, are equal to zero. These independent attributes can then be eliminated without affecting the accuracy of the decision rules related to the first conclusion. All eight decision rules generated were then obtained using only the reduced set of independent attributes.

Results obtained using ExTran-7 at Rocketdyne and ANLYST in the Intelligent Computers Center are identical from the user's point of view. It should be observed, however, that ANLYST produced decision rules using only the reduced set of attribute and that its analysis was conclusion-oriented. These two distinctive and advantageous features of ANALYST may be very important in industrial applications, particularly when the body of examples involves many attributes and a large number of examples, and when the available computer has a limited working memory.

4. Inductive Problem-Solving

A very interesting and potentially important application of an inductive system is as a problem-solving tool. In many cases engineering problems are too complex to solve. This may be caused by a very complicated or confusing theory governing a given problem. Very often, engineering problems are described by such a large number of attributes that their analysis, particularly the analysis of their combinations, exceeds the ability of human experts. Also, when a large body of confusing examples is given, human induction abilities are usually insufficient. Human inability to cope with a large number of attributes, examples, or general pieces of information can be explained by the very limited human working memory, which usually does not exceed seven, with a maximum of nine, pieces of information. In these cases an inductive system can be used to extract decision rules from examples to find an unexpected rule, a missing link, which cannot be identified using traditional engineering analytical methods. An experimental application of an inductive tool in problem-solving is presented here. An inductive system was used in the area of steel structures. This application was developed only for experimental and educational purposes, but it demonstrates the character of inductive problem-solving and may inspire other, product-oriented applications of inductive tools.

Dr. R. Sillen of Novacast A.B. of Sweden is a pioneer in inductive problem-solving in engineering. The example presented here is similar to his now-classical inductive solution of the problem of welds in heavy steel trusses composed of large-diameter pipes.

In our example, a quality control problem in the manufacture of structural components is used to show how inductive problem-solving can improve quality in manufacturing. It has been assumed that a series of 22 steel plate girders was produced with two different types of stiffeners: standard stiffeners with cut corners, and stiffeners of a new experimental type with smooth corners. Three different types of weld web-stiffeners, specifically, fillet, double fillet, and direct weld, were used. Experimental girders were prepared by welders who can be classified as having low, average, and high experience. These girders were prepared during periods of low, normal, and high humidity and when the temperature was below average for the time of year, average, and above average. Three girders were faulty and excessive deformations of stiffeners were observed. Why were these girders faulty?

A group of technicians was assigned to solve this problem, but unfortunately their analysis did not produced the desired results and an inductive problem-solving process was proposed. In the first stage of this process a set of attributes and their values was identified, containing five independent and one dependent attribute, this last representing the conclusion: product quality.

All 22 experimental girders were analyzed and a set of 22 examples was prepared. All examples were entered into an inductive system in a single batch. In this case a point model of the learning process was used. This model was chosen because of the relatively small number of examples.

It was assumed that the objective of the process of learning in our case is to obtain all decision rules whose implementation leads to a faulty product. It was hoped that one of these rules would explain why some products are faulty. For this reason the analysis was reduced to the extraction of decision rules supporting the conclusion product quality = bad.

In the first step of the analysis the significance of individual attributes was analyzed in order to eliminate redundant attributes. This analysis showed that the significance factor for the attribute A2, type of weld, is equal to zero; therefore this attribute is redundant and can be eliminated without affecting the accuracy of the final results. Thus the process of generation of decision rules was performed using only five of the six attributes.

This process produced a single rule using four independent attributes. The rule is very strong and is satisfied in all cases where faulty girders were manufactured. Its complexity and the unusual combination of several factors made it difficult to identify using traditional human and biased analysis, but it was immediately discovered by the inductive system.

5. Shallow Modeling

Modern engineering systems are usually enormously complex. Very often their traditional, theoretical, or deep models are too complicated or inaccurate to be used for practical purposes. For these reasons there is growing interest in shallow modeling. A shallow model of an engineering system is understood here as a relational model based on the observed behavior of the system in a specific experimental domain, and is valid only in that domain. This model is based only on the observed system's behavior, and thus differs from a deep model, which is based on the underlying theory behind the system's behavior.

Inductive tools can be used for the purposes of shallow modeling. An experimental use of ANLYST for this purpose in the Intelligent Computers Center was quite successful, and the methodology of inductive shallow modeling was developed. Inductive systems may become powerful shallow modeling tools applicable to a large class of engineering problems.

In its first experimental application, ANLYST was used to identify relationships among attributes describing cold-rolled steel beams under bending, and attributes representing deformations, whose magnitudes were obtained experimentally.

In the experiment described here, ten attributes were analyzed. Four attributes are dependent; they represent deformations of the steel beams. The remaining six attributes are independent; they represent the dimensions of the beams and the locations of strain gauges.

The global dependency between two groups of independent and dependent attributes is measured by the strength of connection in the theory of rough sets. The influence of individual attributes from a given group of independent attributes on the global dependency between this group and the assumed group of dependent attributes is measured by the significance factor. In case, this factor was monitored during the process of inductive learning, and the results confirmed the well-known relations of the theory of elastic bending. In our analysis three independent attributes had significance factors equal to zero and therefore could be eliminated from the shallow model without affecting its accuracy.

The results mentioned here were obtained from the analysis of only 15 experiments. This number is very small considering the number of attributes involved and the complexity of the problem being analyzed. However, even in this case an inductive shallow modeling tool produced meaningful results.

6. Future Research and Conclusion

Current research in the area of inductive learning in the Intelligent Computers Center concentrated on exploring the possibilities of various applications of inductive systems using the rough sets approach. Applications under consideration include knowledge acquisition, conceptual design, shallow modeling, and inductive problem-solving.

From the engineering point of view, the most important obstacle to the use of inductive systems is the lack of any methodology of their use in different applications. In 1986, the author initiated research on the engineering methodology of inductive learning. This work is of rather limited scope, and more research is needed.

The results of the initial applications of rough sets-based inductive systems are very promising. The author is convinced that these systems can become useful engineering tools for different applications, definitely including knowledge acquisition, inductive problem-solving, and inductive shallow modeling. The author hopes that this presentation will inspire creative readers and that many other interesting and innovative applications of rough sets-based inductive systems will be proposed.

This presentation is based on the following papers:

1. Arciszewski T., Ziarko W., Engineering Applications of Inductive Systems Based on the Rough Sets Approach", prepared for the Journal AI-EDAM.
2. Arciszewski, T., Ziarko W., Adaptive Expert System for Preliminary Engineering Design, Proceedings of the 6th International Workshop on Expert Systems and their Applications, Avignon, France, (1986).
3. Arciszewski, T., Ziarko W., Learning Expert System for Preliminary Design of Wind Bracings, Second Century of Skysrapers, Van Nostrand Publishing House, (1988).
4. Arciszewski, T., Mustafa, M., Inductive Learning Process: User's Perspective, chapter in Machine Learning, edited by R. Forsyth, Chupman and Hall, to appear.
5. Hajdo, P., Arciszewski, T., Ziarko, W., Aktan, H., Inductive Shallow Approach for Generation of Engineering Models, Proceedings of the Ninth European Meeting on Cybernetics and System Research, Vienna, Austria,(1988).
6. Arciszewski T., Mustafa M., Ziarko W., "A Methodology of Design Knowledge Acquisition for Use in Learning Expert Systems," International journal of Man-Machine Studies, Vol.27, No.1,(1987)

MODEL-BASED EXPERT SYSTEM FOR ENGINEERING DESIGN

T. Arciszewski

Intelligent Computers Center
Civil Engineering Department
Wayne State University
Detroit, Michigan 48202
(313) 577-3766

Extended Abstract

1. Introduction

This presentation describes a new concept of model-based expert system for engineering design, with quantitative knowledge representation in the form of generalized characteristics. This system will be applicable to a large class of engineering design problems, including detailing. The presentation gives roots of the proposed concept, its description, the developed methodology of model building and two examples of engineering applications of the proposed concept, one in the area of steel beams under bending, and one in the area of connections between steel columns and roof trusses in industrial buildings.

The proposed concept of a model of an engineering system has its roots in cybernetics and in the general systems theory of control. The author was particularly influenced by the theory of generalized characteristics proposed in the late sixties for adaptive control purposes. It was developed by the author in the early eighties into a cybernetics of design systems. The theory of generalized characteristics was used by the author in his Ph.D. dissertation for the description of the characteristics of wind bracings in skeleton structures in tall buildings. In the early seventies, when this application was developed, the available computer technology was insufficient to support the preparation of computer programs for engineering design with built-in quantitative models for the problem being solved. Only recently has the development of expert systems technology made the implementation of this old concept feasible and attractive. Expert systems with a knowledge base in the form of if-then rules are adequate for dealing with all classification problems, but they are insufficient for design purposes, where decisions require both qualitative and quantitative input. Quantitative input may be obtained from a traditional algorithmic computer program, which is integrated with a given expert system, or through user-defined functions which may be used to conduct the necessary calculations. The first approach requires a very time-consuming integration of expert systems and traditional programs, while the second one requires the development of number-crunching programs in symbolic languages, what makes them very slow and and often exceeds the limits of available expert system development shells. Both approaches are very inefficient, particularly when dealing with complex engineering programs, such as the design of bridges, wind bracings in tall buildings, etc. These are additional reasons why the author found the concept of quantitative knowledge representation in the form of generalized characteristics so attractive.

Research on this concept was initiated in 1985 by Arciszewski, in cooperation with Reynolds of Wayne's Computer Science Department. It has been supported by WSU's Institute for Manufacturing Research. A Ph.D. student, Kamal Shenaq, is involved in this project and his Ph.D. dissertation will result from it. The research led to the development of the proposed model-based expert system and the methodology of its development. Initial implementation and tests were conducted on the problem of designing steel beams under bending. The results obtained were quite promising and have caught the attention of local industry. Working cooperation was established with Albert Kahn Associates of Detroit; the objective is to develop a model-based expert system for design and detailing of connections between roof trusses and columns in industrial buildings, and eventually to develop a system for roof truss design. Albert Kahn provides all necessary technical data and engineering experience regarding these connections, while all necessary LISP programming and computations using the methodology developed is being done in the Intelligent Computers Center at WSU.

2. Concept of a Model-Based Expert System for Engineering Design

An engineering system can be identified for design purposes using a systems approach. In this cas^ its individual subsystems and their relationships can be described using a number of attributes of a qualitative and quantitative character. Qualitative attributes identify the internal structure of the system and are related to such incommensurable system properties as the kind of material used or the shape of individual structural members, but they may also include some measurable properties, which are important from the qualitative point of view. Quantitative attributes are related to the system's measurable properties, such as dimensions, weights, etc. An engineering system can be also described by an equivalent set of attributes, composed of two subsets of independent and dependent attributes. Independent attributes, or control attributes, are under the direct control of the designer. Dependent attributes, or design attributes, are controlled only indirectly, through independent attributes, and their final values can be considered as the result of decisions made regarding independent attributes. Therefore the identification of an engineering system, or the construction of its model for the purposes of design, requires the determination of all attributes and the relationships between independent and dependent attributes. Also, all existing relationships among independent attributes must be known, but these relationships are usually given and may be considered as part of the design constraints.

In this presentation a model of an engineering system is understood as a relational model. It contains a set X of n attributes x_n and all relationships between them necessary for a given model application. In this case, we assume that the same engineering system can be described by several related models developed for different applications and covering different aspects of the system's behavior.

It can be assumed that the relationships between independent and dependent attributes have a functional character and can be described by continuous or discrete functions. Both types of functions can be used in a relational model.

The proposed relational model can be used in expert systems for detailed engineering design. Experience strongly indicates that the qualitative aspects of engineering design can be properly handled using qualitative if-then decision rules. However, there is usually a serious problem with quantitative, or numerical, aspects of design. This problem can be easily solved when dealing with relatively simple design problems requiring only simple calculations which can be performed by user-defined functions combined with an expert system. This solution is not feasible, however, in complex design problems, when numerical input requires significant numerical processing which cannot be handled by user-defined functions. Coupled expert systems which couple symbolic and numerical processing can be used, but this solution requires the development of complex, unhomogenious computer systems. An attractive alternative to a coupled expert system is one with a knowledge base containing qualitative knowledge in the form of if-then rules and quantitative knowledge in the form of a relational model.

The subject of our interest is the methodology of development of a relational model and combining this model with an expert system to develop a computer tool supporting detailed or quantitative engineering design. In engineering terms, we are looking for a model-based expert system and the methodology of its development. Such a system will produce the values of all necessary design variables, or dependent attributes, for a given set of assumptions represented by the combination of values of independent variables, or control variables.

3. Methodology of Model-Building

In engineering, theoretical or deep models of systems are normally used; these models provide all required relationships among independent and dependent attributes. In the case of the proposed model, these relationships have to be identified through an appropriate modeling process.

Our basic assumption is that engineering design knowledge can be represented in different equivalent forms. In our case, we consider general design knowledge in the form of textbooks, design manuals etc. as the available knowledge, which can be used as an initial input in the multistage process of knowledge transformation whose final product, or output, is the relational model. In the proposed process five equivalent but different forms of design knowledge are used and four knowledge transformations performed.

In the proposed model-building methodology the first form of design knowledge is general. This knowledge can be transformed into well-structured procedural and factual knowledge through the traditional analysis of all sources of information, understood as dispersed general design knowledge. Structural procedural and factual knowledge, the second form of knowledge, can be considered together in a computer program, written in a symbolic programming language, for the design purposes of the specific engineering system under consideration. The symbolic programming in LISP can be considered here as the second knowledge transformation. Its result, a computer program, is a third form of the design knowledge, and can be used to produce a collection of design examples, which represents the next equivalent form of this knowledge. The generation of examples is the third transformation of the design knowledge. The collection of examples can undergo the next transformation: the extraction of generalized characteristics, which can then be used to build the desired relational model.

The proposed model-building methodology was found very useful for practical engineering purposes, and it was used in two applications described in the next section.

4. Examples of Structural Applications

4.1 Design of Steel Beams Under Bending

The first experimental application of the developed concept of modeling of an engineering system and the methodology of its preparation was in the area of the design of simply-supported steel beams under bending.

All available sources of information regarding this domain were analyzed, and basic design assumptions as well as analytical and design procedures were identified. It was assumed that the working stress method should be used, and only uniform loads and the deflection related to live load considered. Cross-sections of beams were limited to W-sections, assumed in accordance to the AISC steel manual.

All control attributes and their assumed ranges of variation were shown in a table identifying the assumptions space for the model and the future applications of an expert system containing this model.

The traditional analysis and design process was used to prepare a symbolic computer program written in GCLISP for both the Texas Instruments Business Professional Computer and the Explorer LX. This program contains 187 W-sections taken from the AISC steel manual.

The examples generated were analyzed and divided into classes. Each class contains all examples with the same value of the dependent attribute, that is, the size of the cross-section. The values of individual independent attributes associated with a given class of examples obviously determine the assumptions envelope for a given cross-section. Each class of examples was additionally divided into subclasses according the grade of steel used and the compact coefficient. It was also assumed that two independent attributes, dead load (DL), and live load (LL), can be combined, since in the working stress method they are treated identically. The examples so prepared were used to determine, for individual sizes of cross-section, the relationships between the attribute which represented the sum of the dead and live loads and the length of the beam. This analysis was conducted using a statistical package, Trajectories. Four different types of relationships were analyzed: linear, logarithmic, power, and exponential. The power model provided the best fit.

In this case the generalized characteristic, the relationship cross-section versus dead plus live load and beam length is in the form of a strip, including all points between two parallel lines described by the power function. Since there are two steel grades and two compact coefficients are considered, four classes of generalized characteristics cover the entire assumptions space.

The strip character of the generalized characteristics enabled us to store these relationships in the database program prepared using DBASE III. The database program contains all selected W-sections with their corresponding assumptions envelopes.

The results obtained were used to prepare an expert system, based on the TI Consultant Plus expert system development shell. The prepared expert system contains only six if-then type rules plus the knowledge in terms of facts stored in the data base system.

4.2 Design of Connections: Column-roof Truss in Industrial Buildings Including Prying Action

The first proof-of-concept application of the proposed expert system was successful; the system performs as expected. This showed the feasibility of the proposed new type of expert system. The author decided to select the second application with commercial potential, which would be developed in cooperation with industrial experts. Working cooperation was established with a group of structural engineers from Albert Kahn Associates of Detroit, who are interested in expert system technology. A number of complex structural design problems was reviewed, looking for a problem whose complexity and engineering importance would justify the development of an expert system.

The analysis and design of joint column-roof trusses in industrial buildings was finally selected. When a prying force is considered, the analysis and design of such joints require significant experience and are time-consuming. This joint is also very important to the safety of an industrial building, and its failure leads to the collapse of the entire transverse system. For these reasons the design of the joint column-roof truss is considered as one of the most important structural design problems in the domain of industrial buildings, and the development of a computer system for its safe design is expected to be of significant engineering importance.

The design of such joints requires the design of connecting angles bolted to the column flange, the design of the weld to the gusset plate, and checking the prying action through the column itself. As a first step, the design of the connection angles is considered here.

In cooperation with industrial experts, the domain engineering knowledge was analyzed and the design process identified. Also, all recommended steel plate thicknesses and sizes of truss members were determined and used to prepare the assumptions space. This engineering knowledge was used to prepare a LISP program for the TI Explorer LX. This program contains all the design variables and their possible values. The developed program was used to generate approximately 123,000,000 examples, which represent the equivalent of the domain knowledge and were used to prepare generalized characteristics. These examples covered all feasible combinations of assumptions and therefore can be considered as the equivalent of the factual and procedural knowledge contained in the LISP program. The following independent attributes are considered: moment (M) which equals the product of reaction and eccentricity, web thickness of the top chord of the truss (Tw), angle size and gauge distance, bolt diameter (D), pitch or vertical spacing (P), and number of bolt lines (NL). The dependent attribute we are looking for is the required angle thickness (Tr).

It was assumed, following engineering practice, that the bending moment is decisive and that a designer is chiefly concerned with providing sufficient bending capacity for the joint being designed. Therefore the relationships bending moment-number of bolt lines-required angle thickness were determined for individual combinations of assumptions regarding bolt diameter, pitch (vertical spacing), web-thickness of top chord, and number of bolt lines. Sixty-three such relationships were identified, and covering the entire assumptions space. The relationships obtained were identified as step functions. These relationships are equivalent to all generated examples.

The discrete character of generalized characteristics was very advantageous here, because it enabled us to store these relationships in the database program prepared using DBASE III. This database was integrated with an expert system prepared using the Consultant Plus expert system development shell. The finished expert system contains only eight if-then type rules plus the knowledge in terms of facts stored in the data base system.

The system was prepared as proof of the concept, but it could be used for practical design purposes.

5. Conclusions

The proposed concept of a model-based expert system for engineering design is feasible. It can be used in the development of expert systems for design purposes to address a large and very important class of engineering problems, which require qualitative and quantitative data. The present research is still in progress, but results obtained up to now are conclusive as far as the feasibility of the proposed concept and the methodology of its development are concerned. The developed prototype for the design of connections in column-roof trusses in industrial buildings has been validated in the Intelligent Computers Center and by the industrial experts. Its commercialization is being considered.

This presentation was based on the following papers:

Shenaq K., Arciszewski T., Model-Based Expert System for Structural Design, paper submitted for Texas Instruments competition, 1988

Arciszewski T., Wind Bracings in the Form of Belt Truss Systems in Steel Skeleton Structures of Tall Buildings, Ph.D. Dissertation, Warsaw Technical University, Poland, 1975

Arciszewski T., Pancewicz Z., Analysis of Wind Bracings Characteristics in Typical Skeleton Structures, Proceedings of Wroclaw Technical University, 1976.

Staniszewski R., The Generalized Model of Parameters Optimization in Mechanical Engineering, Bulletin of the Polish Academy of Sciences, No.9, 1970

Staniszewski R., Cybernetic Model of an Adaptive Design System under Unstable conditions," Progresses of Cybernetics, No.1, Poland, 1978

Staniszewski R., Cybernetic Theory of Engineering Design, Ossolineum, Poland, 1986

USING CAD FOR KNOWLEDGE BASED SYSTEMS

Mike Case[1] and Lee Hian Quek[2]

U.S. Army Construction Engineering Research Laboratory
Champaign, Illinois

PROBLEM STATEMENT

The Energy Technology Alternatives team at USA-CERL is developing knowledge-based applications which will design alternative energy systems for commercial scale construction. The first module, for the design of solar thermal domestic hot water systems, required information during the inference process which was most readily available from drawings generated by a Computer-Aided Design (CAD) software package. Unfortunately, there was no way of foretelling exactly which CAD software, operating system, or processor architecture would have been used to create and store the drawing(s) to be accessed. Furthermore, a method to extract information from the drawing in a form usable by an inference engine was required.

OBJECTIVE

There is a need for a generic means of accessing graphically represented information. This program of research seeks to develop such a capability within the larger context of a Multiple Cooperating Knowledge Source (MCKS) framework being developed in cooperation with the Knowledge-based Engineering Systems Laboratory at the University of Illinois.

---

[1]Principal Investigator, US Army Construction Engineering Laboratory, Champaign, Illinois.

[2]Research Assistant, Department of Computer Science, University of Illinois at Urbana-Champaign, Champaign, Illinois.

APPROACH

Multiple Cooperating Knowledge Sources

The rationale for requiring multiple knowledge sources to work together is that there are many methods of obtaining information. On a computer network, we may currently access expert systems, databases, induction learning systems, dedicated engineering software (finite element, linear programming, etc.), and CAD systems. In many cases, the information supplied by these sources might conflict, yet still be correct from each individual viewpoint. A MCKS system provides a mechanism to allow these sources to communicate partial results, query other sources, identify conflicts, and resolve those conflicts. Each of the knowledge sources listed above may be accessed in a common way, even if it is on a different machine in the network. We have developed an approach for integrating CAD knowledge into the MCKS framework.

A Generic CAD Interface

A Knowledge-based system might reasonably be expected to ask questions about a drawing, reason about the information obtained, and then modify the drawing. The interface which we have developed works with one or more human designers to extract and manipulate drawings using the keyboard, display monitor, and digitizer input. This is a generic interface, in that it does not depend on the capabilities of any one commercial CAD software package. It does not translate a drawing from one format to another, but rather extracts information. The interface is able to:

1. Enter the CAD environment, obtain a reference point, and determine physical scale.
2. Copy all or relevant portions of the drawing for later manipulation.
3. Prompt the designer for significant information about the drawing.

4. Manipulate the drawing as needed, including an ability to exercise pan, zoom, and drawing functions.

5. Define graphic objects for manipulation. For example, define a pipe object and insert instantiations of different lengths and widths.

6. Return from the CAD environment with desired information in a form readable by the knowledge system.

There are four layers in our CAD interface. The top layer is the **CAD SOURCE**, which communicates directly with the MCKS environment. It is an inference engine based system with meta-knowledge of the type of information which may be accessed from a particular CAD system. This section communicates with both the MCKS blackboard and the next layer. It will frequently include domain specific information. The **CAD LANGUAGE** layer is a collection of high level functions which perform the six tasks listed above. This module contains a "Macro" language for CAD functions which are not dependent on the commercial CAD software implementation or operating system. Examples include drawing lines, copying sections, and prompting for information. The **CAD DRIVER** layer is completely implementation dependent, consisting of translator modules which generate command sequences for a particular CAD package from the higher level functions of the CAD LANGUAGE module. This driver starts the CAD program, transmits the command file, collects replies to queries, and returns control to the upper level modules. Ultimately, this level will handle differences in operating systems as well (MS-DOS, Unix, VMS, OS/2, etc.). The final layer is the **CAD PROGRAM** itself, which could be any commercial software with an internal programming language, such as AutoCAD, Intergraph, or VersaCAD.

CURRENT STATUS

The CAD interface has been implemented using Gold Hill Common LISP and

AutoCAD on a COMPAQ DeskPro 286. The solar design knowledge-based system is able to issue requests for information such as roof area, building orientation, and legal roof penetration points. It then uses this information to design a solar collector array with balanced reverse-return piping, which is automatically drawn onto a copy of the original drawing. Performance limitations (processing speed) are due mainly to the single-tasking nature of MS-DOS, in that AutoCAD must be loaded each time a set of queries is sent to the CAD interface. This problem is avoided by grouping the queries together and posing them as a set. In a multi-tasking environment, the CAD program would be kept running as a process and accessed when needed.

FUTURE RESEARCH

As the MCKS framework is developed further, we intend to implement the CAD interface as an independent knowledge source capable of working in a networked environment using separate processors. In addition, support will be developed for other CAD software used by designers on both micro-computers and engineering workstations.

BIBLIOGRAPHY

Charniak, E., Riesbeck, C.K., McDermott, D.V., and Meehan, J.R., *Artificial Intelligence Programming*, 2nd ed., Erlbaum, Hillsdale, New Jersey, 1987, pp. 149-275.

Cox, B.J., *Object Oriented Programming: An Evolutionary Approach*, Addison-Wesley, Reading, Mass. 1987.

Fikes, Richard and Kehler, Tom. "The role of Frame-based Representation in Reasoning," *Communications of the ACM*, Sept. 1985, V28, No. 9, special section.

Steeb, R.S., McArthur, S.J., Cammarata, S.J., Narain, S., and Giarla, W.D., "Distributed Problem Solving for Air Fleet Control: Framework and Implementation," *Expert Systems*, Klahr, P. and Waterman, D.A., ed, Addison-Wesley, Reading, MA., 1986, pp. 391-432.

Stefik, M. and Bobrow, D.G., "Object-Oriented Programming: Themes and Variations," *AI Magazine*, Vol. 6, No. 4, Winter 1986, pp. 40-62.

Stefik, M. Bobrow, D.G. Mittal, S. , and Conway, L. "Knowledge Programming in LOOPS: Report on an experimental course. *AI Magazine*, Vol. 4, No. 3, Fall 1983, pp. 3-14.

Thompson, J.B., Lu, S. C-Y., "Multiple, Cooperating Knowledge Sources for Integrated Engineering Design," Extended Abstract, Knowledge-Based Engineering Research Laboratory, University of Illinois at Urbana-Champaign, 1988.

Waterman, D.A., *A Guide to Expert Systems*, Reading, MA: Addison-Wesley, 1986.

The Construction Schedule Generator: AI Tools for the Generation of Initial Construction Schedules

By D.Echeverry[1], S.Kim[2] and C.W.Ibbs Jr.[3]

1. INTRODUCTION.

The task of generating the schedule for a construction project relies heavily on expertise, skills and knowledge of experienced project managers or construction schedulers.

The objective of the research effort described here is to produce a computerized tool that provides "intelligent" assistance in the generation of initial construction schedules. The goal is to have a system that receives as input project information, and that provides as output a reasonable schedule of the given construction project. The system is also conceived as being interactive in such a way that the user (construction scheduler) is able to provide his feedback to the scheduling process whenever desired.

As mentioned above, it is believed that scheduling involves a large body of knowledge. Therefore, the present research has been narrowed down to only a certain type of construction projects, namely mid-rise office/residential buildings. The prototype that will be generated as part of this effort is expected to handle only schedules for these construction

---

[1] Research Assistant, Civil Eng. Dept., Univ. of Illinois, Urbana, Illinois 61801.

[2] Team Leader, Construction Management Team, CERL, Champaign Il 61820.

[3] Associate Professor, Dept. of Civil Eng., Univ. of California, Berkeley Ca 94720.

projects.

2. BACKGROUND.

The work described here is another step in a continuing effort both at the University of Illinois and at CERL. This work is a follow up of research performed for acquiring and eliciting the knowledge used for criticizing construction schedules [O'Connor 86], [De La Garza 88]. The long term goal of this continuing effort is to generate an integrated computer environment for construction planning and control.

Important research efforts are being invested in other research institutions, pursuing similar goals, and following different approaches. It is necessary to mention that the present work has been enriched by communicating with other research teams, also involved in automated scheduling, at the University of Illinois [Hassanein 88], the University of California at Berkeley [Ibbs 88], Carnegie Mellon University [Hendrickson 87], Stanford University [Levitt 87] and MIT [Logcher 87]. Main contributions of the present work are expected to be the consideration of trade interaction for scheduling and an exploration of techniques for allowing the scheduling system to "learn" from executed scheduling tasks.

3. METHODOLOGY.

The initial phase of this work is an attempt to acquire, represent and utilize the knowledge used by expert schedulers when generating a schedule. The next phase, described in more detail in Section 5., consists of the inclusion of additional features that would allow the capture of the experience generated when solving a scheduling problem, for use in future scheduling problems.

The following tasks are the most relevant ones in achieving the goals of the first phase:

* **KNOWLEDGE ACQUISITION:** experienced schedulers of different construction firms will be contacted. The process of interviewing them for acquiring scheduling knowledge is being defined. Some of the important aspects to be explored are: (1)process of breaking down a project into activities and tasks for construction; (2)dominant criteria in generating a schedule (e.g. equipment, crew flow, imposed milestones, etc.); (3)process of determining activity durations and assignment of resources; (4)determining factors for dictating construction sequences (precedence relationships); and so on.

Construction literature represents another source of easily accessible knowledge. A survey of papers and textbooks is being performed for extracting applicable knowledge.

* **KNOWLEDGE REPRESENTATION:** different knowledge units or modules have been identified (see Section 4.). Different forms of representing the knowledge within these modules are being analyzed, in order to ensure clarity, expandability, and compatibility among the different modules. In general, a blackboard architecture will be used. The knowledge will be

represented using a combination of rules and frames (objects) with inheritance and message passing features.

* **CPM KERNEL:** a vital part of a schedule is the determination of starting and finishing times for all the activities involved in the project, as well as floats and criticality. A demon residing in the same environment as the knowledge base is used for performing the CPM calculations that yield all this information.

* **PROTOTYPE TESTING AND VALIDATION:** once an adequate body of knowledge has been included in the prototype, testing and validation of the system can begin.

4. STATUS OF THE WORK.

An initial version of the CPM kernel has been completed. It resides in the same environment (Goldworks) where the knowledge base is expected to reside, and is implemented using an object oriented approach.

The knowledge acquisition and knowledge representation tasks are under way. The status of these tasks can be summarized by describing the different knowledge modules or sources that are included in the knowledge base (see Figure 1.). Most of the work performed so far in knowledge representation has been conceptual. Implementation is expected to begin soon.

* **CONSTRUCTION KNOWLEDGE MODULE:** this module contains a compositional breakdown of a project (Foundation, Structure, Closure, etc.). For each compositional element of a project, there is a prediction of some of the more frequent

construction methods used to build it. This description consists of a breakdown into typical tasks. These tasks have attributes that describe constraints, ranges of required resources, and other characteristics.

* **SCHEDULING KNOWLEDGE MODULE:** this module contains the knowledge necessary to breakdown a project into activities, generate durations for these activities, and link them to produce constructive sequences.

* **ASSUMPTION HANDLER KNOWLEDGE MODULE:** the prototype resulting from this research effort is expected to deal with incomplete project description. Often, reasonable project schedules are required before all the project design phase is concluded. The function of this module is to contain knowledge to make reasonable assumptions to complement insufficient input information about the project.

5. FUTURE RESEARCH.

Construction schedulers gain expertise through experience. Each project schedule they generate enriches their knowledge. A computerized schedule generator therefore should learn from its own scheduling experiences too. A computerized schedule generator should be able to use previous solutions for new scheduling problems.

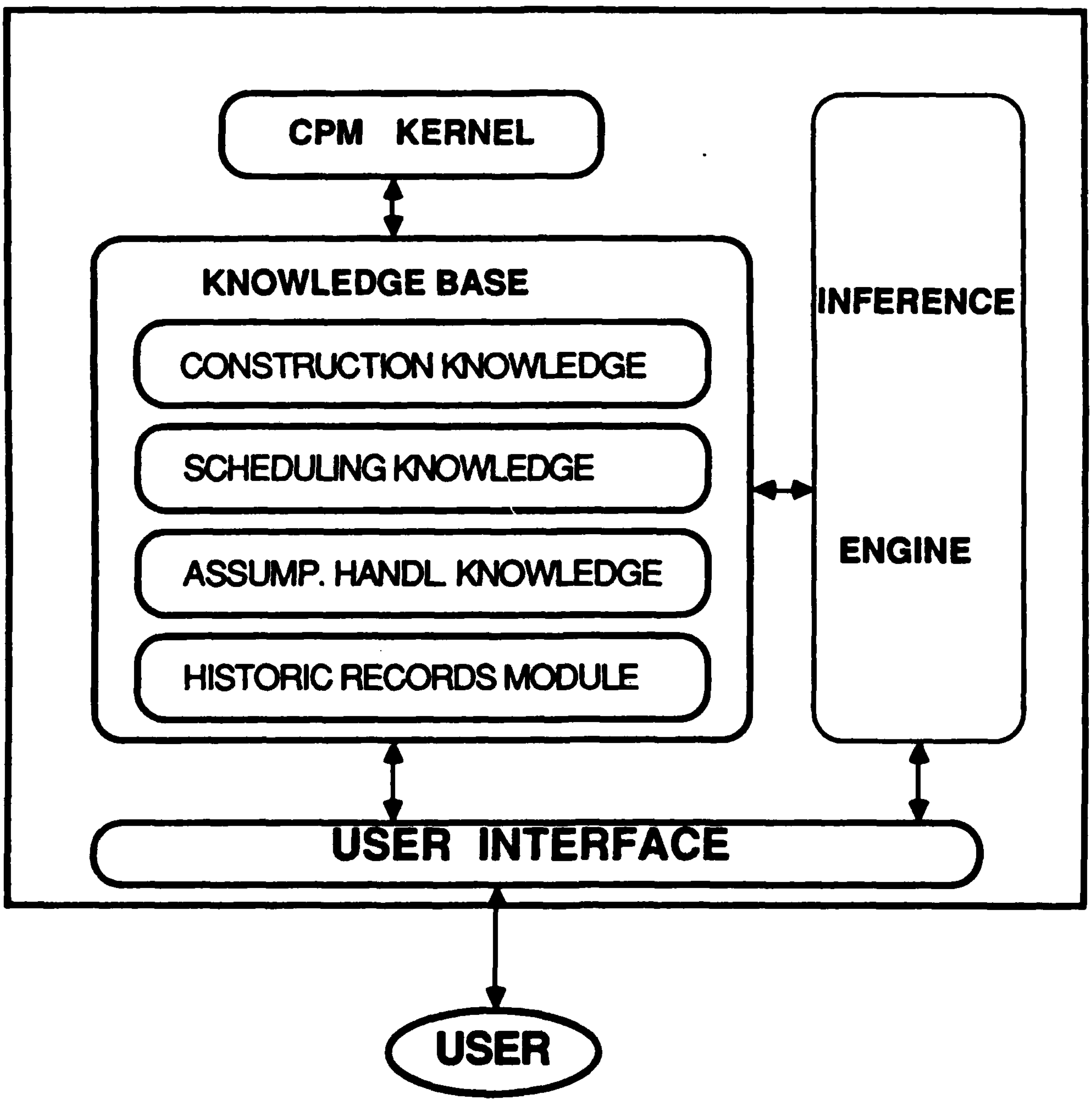
The area of machine learning presents a very interesting long term potential in this respect. It is however at its infancy. But some steps can be taken in the present work, in such a way that not everything is lost from each scheduling problem that is solved by the system.

The approach that is intended here for future research is to build an object-oriented database of previously scheduled projects. By comparing the current scheduling problem with previous ones, similarities should be matched, and parts of the existing solution processes could be transformed to satisfy the current problem constraints. This is an area of active research in the AI field called analogical reasoning. As described above, future research is expected to explore its feasibility for schedule generation.

Other research efforts predicted as a continuation of this project include its integration to other Construction Planning and Control systems and to design systems as well.

6. FIGURES.

THE CONSTRUCTION SCHEDULE GENERATOR



7. REFERENCES.

[Carbonell 82] Carbonell J.G., **Learning by Analogy: Formulating and Generalizing Plans from Past Experience**, Technical Report CMU-CS-82-126, Dept. of Computer Science, Carnegie Mellon Univ., June 1982.

[De La Garza 88] De La Garza J.M., **A Knowledge Engineering Approach to the Analysis and Evaluation of Construction Schedules for Vertical Construction**, Unpublished PhD thesis, Dept. of Civil Engineering, Univ. of Illinois, 1988.

[Gray 86] Gray C., **"Intelligent" Construction Time and Cost Analysis**, Construction Management and Economics Journal, 1986.

[Hendrickson 87] Hendrickson C., Zozaya-Gorostiza C., Rehak D., Baracco-Miller E. and Lim P., **An Expert System for Construction Planning**, Publication No. EDRC-12-16-87, Civil Engr. Dept., Carnegie Mellon Univ., May 1987.

[Ibbs 88] Ibbs C.W., Chang T.C. and Echeverry D., **Construction Network Generation with Approximate Reasoning**, Unpublished paper, April 1988.

[Hassanein 88] Hassanein A. and Melin J.W., **Development of Construction Contractor Resource Balancing Algorithms**, Unpublished Technical Report, US Army Constr. Eng. Res. Lab., January 1988.

[Levitt 87] Levitt R.E. and Kunz J.C., **Using Artificial Intelligence Techniques to Support Project Management**, Journal of Artificial Intelligence in Engineering Design, Analysis and Manufacturing, Vol 1, No. 1, 1987.

[Logcher 87] Logcher R.D., **Personal Communication**, MIT, October 1987.

[O'Connor 86] O'Connor M.J., De La Garza J.M. and Ibbs C.W., **An Expert System for Construction Schedule Analysis**, in Expert Systems in Civil Engineering, edited by Kostem C. and Maher M.L., ASCE, April 1986.

[Zhao 87] Zhao F., **Application of AI Techniques to Planning Structural Systems Configurations**, Master's thesis, Dept. of Civil Engineering, Carnegie Mellon Univ., March 1987.

An Object-Oriented Model for Building Design and Construction†

J. H. Garrett, Jr., J. Breslin and J. Basten ‡

Introduction

During the life-cycle of a building, many different descriptions of that building are developed and used by the agents designing, constructing and occupying it. Starting with the specification of desired functionality, to the description of the physical structure after many years of use and modification, the building description goes through many different phases and transformations. Gielingh describes five distinct phases of a building description: as-required, as-designed, as-planned, as-built and as-used [4]. The agents of the building design and construction process are constantly transforming information between and within these descriptive phases. For example,

> the architect transforms the functionality requirements of the owner (as-required) into a collection of interrelated living and work spaces (as-designed) that together provide the required functionality;
>
> the structural engineer transforms the form envisioned by the architect (as-designed architecturally) into a collection of structural systems, subsystems and components that transmit live and dead loads from their points of application into the foundation (as-designed structurally);
>
> the construction manager transforms the building drawings and specifications (as-designed) into a set of sequenced, staffed activities (as-planned) that, when completed, gives physical reality to the building;
>
> the contractor and subcontractors use the as-designed and as-planned descriptions of the building and physically transform them into a physical reality (as-built);
>
> the facility manager continues to transform the original physical structure (as-built) through maintenance, renovation, space reallocation, or replacement over time, thus requiring an updated description (as-used).

Traditionally, these agents communicated with each other only through very rigid channels. Hence, it is no surprise that the automation of this process has lead to "islands of automation", where sophisticated systems for addressing portions of the building design and construction

---

† Submitted on June 15, 1988 to CERL/ASCE Expert Systems Committee joint conference on Expert Systems in Civil Engineering, June 29-30, 1988

‡ Assistant Professor, Research Assistant, research Assistant, University of Illinois, Department of Civil Engineering, 205 North Mathews, Urbana, IL , 61801.

process exist, but do not effectively communicate with other agent systems. For example, the architect is able to design the spatial layout of a structure using the computer, but the results cannot be used by the structural design system unless translated.

Howard and Rehak proposed a method for integrating these systems by providing a conceptual global model of a building and then translating between the global model and the local models of the various applications [5]. Although this macro model of integration was clearly spelled out and tested in [5], the form and content of this conceptual global model were less developed. The objective of the research described in this paper is to apply object-oriented and feature-based solids modelling techniques to develop a building model that is capable of representing functional and spatial characteristics, both abstract and detailed, and can be used as global conceptual model to integrate the building design and construction processes.

A conceptual global model of a building must be able to represent the abstract functional and spatial descriptions developed in the early stages of the design as well as the detailed functional and spatial descriptions developed in later stages of design and used in construction and facility management. For example, at an early stage in the building design process, the functional and spatial description of a floor in a structure might look like that in Fig. 1., while at a later stage in the design process it might look like that in Fig. 2.

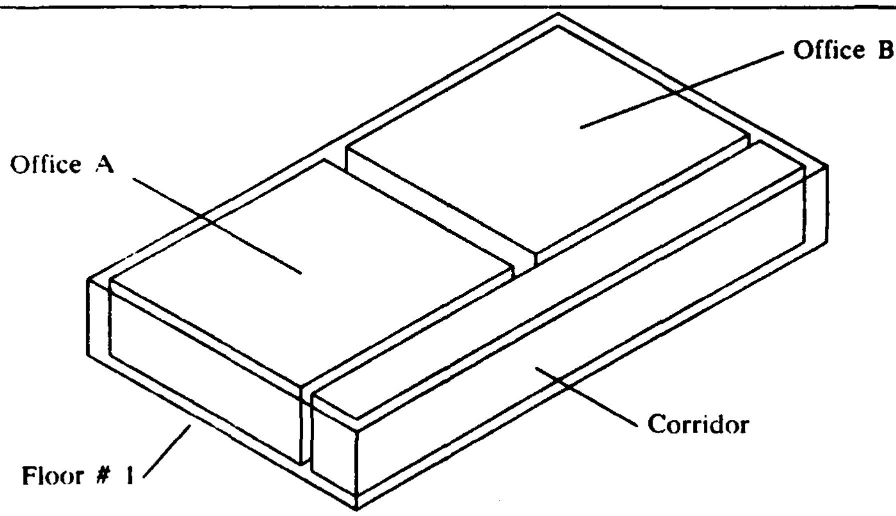


Fig. 1. Abstract Spatial and Functional Description

In addition to the ability to describe various functional and spatial abstractions, a mechanism is needed to ensure consistency between the abstract and detailed descriptions. When going from the abstract description in Fig. 1. to the more detailed description in Fig. 2., The walls in the detailed description in Fig. 2. are part of the rooms and corridor, which themselves are part of

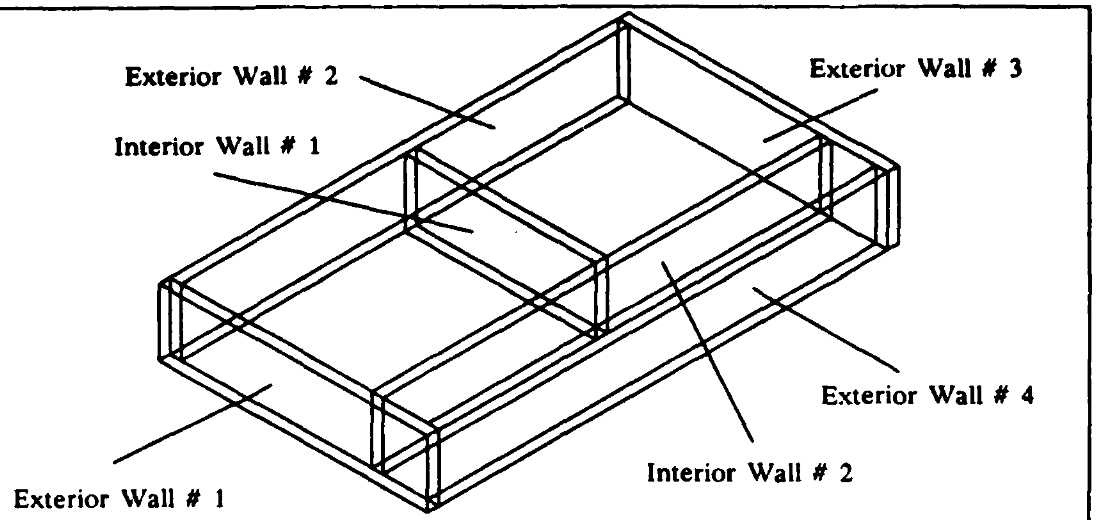


Fig. 2. Detailed Spatial and Functional Description

the floor in Fig. 1. It can also be said the the walls are spatially related to the spatial positions of the rooms and corridors. Hence, these relationships are the mechanism by which consistency between the abstract and detailed descriptions can be maintained.

The objective of the research described in this paper is to develop a building model that is capable of representing functional and spatial characteristics, both abstract and detailed, and can be used as global conceptual model to integrate the building design and construction processes.

Conceptual Global Building Model

Buildings are physical objects and composed of other physical objects. There are a finite number of physical objects from which all buildings are constructed, such as steel sections, concrete sections, bricks, duct, pipe, door, etc. Each such physical object has a geometric shape, a location and orientation within the building, and a collection of properties related to the type of material. More complex physical objects are built up from less-complex physical objects, such as a wall being composed of bricks. At some level of aggregation, a composition of the physical objects is capable of performing some function, such as structural support or air tempering. Thus, aggregations of physical objects form functional systems.

Buildings are composed of many functional systems that support all of the intended functions of the building, such as protection, comfort, lighting, performance of specific tasks, etc. These functional systems are composed of subsystems that are eventually composed of physical objects. Each of these functional system hierarchies represents a different perspective of the physical composition of the building. For example, the structural engineer looks at the superstructure, which may be comprised of frames that are at the lowest level made up of steel sections. This is his perspective of the building, but not that of the HVAC engineer or anyone else. The HVAC engineer views the building as a collections of ducts, compressors, AC loads, etc. Hence, to support all of the agents in the design and construction process, their functional perspective of the building must be representable within the conceptual global model. In addition to supporting different functional perspectives, different ways of building up these hierarchies must be supported.

1. In a bottom-up fashion, the physical objects can be aggregated into functional subsystems, which can then be aggregated into functional systems, and so on.
2. In a top-down fashion, the functional systems can be decomposed into functional subsystems, which can then be decomposed into physical objects.

For most of the agents, design is a top down process, where abstractions are made about the systems being designed and then refined over time. Construction on the other hand is a process of building the functional systems from physical objects. Because this model must support both design and construction, the hierarchies must be constructable using either one or both methods.

The primary function of all buildings is to provide protected space for a variety of functional uses. Hence, buildings are often described in terms of their component subspaces, each of which may have their own function, such as office, work area, etc. These subspaces are oriented in three-dimensional space with respect to some datum (locations), as well as with respect to each other (spatial relationships). In addition to the building spaces, the objects that are used to divide and enclose that space are physical and must be spatially oriented and interrelated as well. Much like the functional systems, which are eventually decomposed into low-level physical objects, the abstract building spaces, such as a floor, can eventually be decomposed into spaces associated with the physical objects comprising the building. The space associated within the walls of a room can be represented as a physical space containing air, much the same way the walls will be volumes made of brick and mortar. However, the space associated with an entire floor of the building is an abstract space composed of abstract room spaces, which are eventually subdivided into spaces made of either solid materials or air.

For both functional systems and spatial systems, subsystems are related to systems through aggregation (e.g., part-of/components for functional and within/surrounds for spatial). In addition to these aggregation relationships, there exist other relationships between these subsystems (e.g., *connected-to* for functional and *above* for spatial). In addition to being able to represent functional or spatial structure, a relationship can be used to represent ***design intent***. For example, when an architect passes on drawings of a product, the locations of the various rooms and halls are apparent from the drawing, but not his underlying intent. If someone later changes the layout, will the architect still be satisfied? His intentions are known only if the explicit spatial relationship that the kitchen be adjacent to, and west of, the dining room is recorded.

Once they are known, constraints can be added to automatically maintain these spatial and functional relationships.

In [1], Eastman defines a general organization for an integrated design database. According to Eastman, "The most convenient general conceptual organization of a project database is as a description of entities and their composition.[1]" He goes on to state that the representation of entities and their attributes alone is not sufficient for design; in addition to the entities, both functional and spatial compositions must also be represented because the individual parts may not represent the attributes of the composition, referred to a "emergent properties" by Eastman [1]. Hence, a more complete description consists of the entities, the functional composition and the spatial composition. The conceptual model that Eastman developed for structuring design information is termed an *abstraction hierarchy*. The research described in this paper is an attempt to extend the abstraction hierarchy approach described in [1], and apply it as the organizational structure for the conceptual global model of a building.

Law also describes the application of the abstraction hierarchy approach to the representation of building design information in [9]. In that work, he describes a three part model made up of the following three parts: a topological hierarchy, a structural hierarchy and an architectural hierarchy. While being a correct and complete set of hierarchies for architectural and structural design, they do not form a complete model that would support all agents in the building design and construction process.

Keirouz is currently developing an object-oriented building model for use in robotic construction planning. The model consists of functionally and spatially interrelated physical objects [7]. Each object maintains information about its geometry, non-geometric attributes, and its function, but it is unclear whether his functional system objects will be explicitly represented or only exist implicitly as interrelated objects. Keirouz places both functional and spatial relationships between two physical objects in what he calls a connection object. It is also unclear whether these connections are physical or purely relational objects. Because the model is being developed to support robotic construction planning, it is capable of representing the "as-designed" information, but it does not appear to be able to describe the abstract functional and spatial descriptions that occur at earlier stages of the building design process. Such a capability is required for a conceptual global model of a building.

The conceptual global model described in this paper is organized into three categories: physical objects, functional systems, and spatial systems. Note that these three categories are almost identical to those of Eastman, and much more broad than those of Law. An object-oriented approach, described and justified in a later section, is used to develop the conceptual global model. An object is a collection of slots that represent either attributes or methods that can be performed by the object. The model is constructed from many of objects, each being either a physical object, functional system, or spatial system.

This model is currently being implemented in KEE [6], and a CommonLISP-based solids modeller, Vantage[13]. KEE is being used as the object-oriented environment in which the physical objects, functional systems, spatial systems and relationships are defined. The lisp-based solids modeller, Vantage, is being used as the informationally complete solids model on top of which will be implemented Woodbury's feature-based geometric reasoning architecture.

Acknowledgements

This research is being supported by the Army Research Office as part of the program of the University of Illinois Advanced Construction Technology Center.

Bibliography

[1] Eastman, C. M., "The Representation of Design Problems and Maintenance of Their Structure", *Artificial Intelligence and Pattern Recognition in Computer-Aided Design*, Latombe, ed., North-Holland Publishing Company, 1978.

[2] Fenves, S. J. and N. Baker, "Spatial and Functional Representation Language for Structural Design", Engineering Design Research Center Report, Number EDRC-12-18-87, Carnegie Mellon University, Pittsburgh, PA, 1987.

[3] Garrett, Jr., J. H. and S. J. Fenves, "A Knowledge-Based Standard Processor for Structural Component Design", *Engineering with Computers*, Volume 2, Number 4, pages 219-238, 1987.

[4] Gielingh, W. F., "General Reference Model for AEC Product Definition Data - Version 3", Technical Report Number BI-87-73/68.5..4201, TNO Institute for Building Materials and Structures, Delft, Netherlands, 1988.

[5] Howard, H. C. and D. R. Rehak, "Interfacing Databases and Knowledge Based Systems for Structural Engineering Applications", Engineering Design Research Center Report, Number EDRC-12-06-87, Carnegie Mellon University, Pittsburgh, PA, 1986.

[6] *KEE Software Development System User's Manual*, Document No. 3.0-U-1, IntelliCorp, Mountain View, CA, 1986.

[7] Keirouz, W. T. , D. R. Rehak and I. J. Oppenheim, "Development of an Object-Oriented Domain Model for Constructed Facilities", Engineering Design Research Center Report, Number EDRC-12-10-87, Carnegie Mellon University, Pittsburgh, PA, 1987.

[8] Keirouz, W. T. , D. R. Rehak and I. J. Oppenheim, "Object-Oriented Programming for Computer-Aided Design", Engineering Design Research Center Report, Number EDRC-12-09-87, Carnegie Mellon University, Pittsburgh, PA, 1987.

[9] Law, K. H. and M. K. Jouaneh, "Data Modelling for Building Design", in Proceedings of the Fourth Conference on Computing in Civil Engineering, American Society of Civil Engineers, Boston, 1986.

[10] Law, K. H., M. K. Jouaneh and D. L. Spooner, "Abstraction Database Concept for Engineering Modelling", *Engineering with Computers*, No. 2, pp 79-94, 1987.

[11] Libardi, E.C., J. R. Dixon, M. K. Simmons, "Computer Environments for the Design of Mechanical Assemblies: A Research Review", *Engineering with Computers*, No. 3, pp 121-136, 1988.

[12] Stonebraker, M. and L. A. Rowe, "The Design of POSTGRESS", Memorandum No. UCB/ERL 85/95, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, 1985.

[13] *Vantage User's Manual*, Carnegie Mellon University, Pittsburgh, PA, 1987.

[14] Woodbury, R., "The Knowledge Based Representation and Manipulation of Geometry", Unpublished Ph.D. Dissertation, the Department of Architecture, Carnegie Mellon University, Pittsburgh, PA, 1987.

INTERACTIVE DOMAIN MODEL (IDM): A SYSTEM FOR AUTOMATED KNOWLEDGE ACQUISITION

Thomas M. Gatton[1], Bruce Ferguson[2], Peggy Johnson[1], Debbie J. Lawrence[1], Frank W. Kearney[3]

1. BACKGROUND

The Materials and Quality Assurance (MQA) team of the Engineering and Materials Division at the Construction Engineering Research Laboratory has been involved with artificial intelligence research and development in the areas of expert systems, machine learning, image processing, robotics, and intelligent control systems since 1980. We have developed expert systems in the following applications:

1) WELDER - Non-destructive (NDE) evaluation of welds,
2) NDE Condition Assessment of Concrete,
3) ESRAM - Maintenance of railway systems,
4) DRHVAC - Diagnosis of residential heating, ventilating, and air conditioning equipment,
5) ESROM - Repair and maintenance of built-up-roofs (BUR),
6) MITER - Diagnosis and repair recommendations for miter gates,
7) ESAP - Diagnosis problems and recommend inspection procedures in asphalt paving, and
8) ESROC - Inspection recommendations for quality assurance in the construction of built-up-roofs.

These applications have been developed with both commercially available shell systems and our own expert system shell, CRITIC.

---

1 Principal Investigators, US Army Construction Engineering Research Laboratory, Champaign, Illinois.

2 Research Assistant, Dept. of Computer Science, University of Illinois at Champaign/ Urbana, Illinois.

3 Team Leader, Materials and Quality Assurance Team, Engineering and Materials Division, US Construction Engineering Research Laboratory, Champaign, Illinois.

One of the prime areas of focus in our development of expert systems has been in the area of knowledge acquisition. The knowledge acquisition process is a major bottleneck in the development and application of expert systems [GAINES88]. Successful expert system development must deal with the problems involved in knowledge acquisition.

In the traditional role of the knowledge engineer, domain information is gathered from the expert(s) and structured into a knowledge base. This requires that the knowledge engineer become a "pseudo expert" in order to organize the domain knowledge into a form that can be converted into a knowledge base. The difficulty and time requirements that are involved in this process have sometimes been avoided by educating a domain expert in the development of an expert system shell rather than educate a knowledge engineer to a "pseudo expert" level in the domain.

Currently, the MQA team approach to expert system development involves the creation of a prototype system based on "book" knowledge. This prototype is then taken to the expert for critical review. This approach has several advantages. First, the expert is not faced with a knowledge engineer that has little, if any,knowledge about the domain. Secondly, the expert is much more inclined to criticize mistakes in an existing system than be able to organize the domain knowledge in a constructive way from scratch.

Although this approach has significantly improved the knowledge acquisition process, it still remains a major bottleneck in the development of expert systems. Research into the development of knowledge engineering tools must consider this problem and find solutions.

2. OBJECTIVES

The objective of this project is to automate the knowledge acquisition process and allow direct input from expert(s) into a system for the generation of expert system knowledge bases. This involves, first, the acquisition of knowledge from an individual expert and then, secondly, the integration of knowledge from multiple experts. This report will deal with the first problem, that of knowledge acquisition from a single expert. Furthermore, as indicated by the applications that we have presently developed, the automated knowledge acquisition system will only be required to handle the diagnosis of physical system for identification of faulty components and recommendations for correction procedures.

3. METHODOLOGY

The use of an abstract domain model has been selected as a suitable user interface. Through recent research done by the MQA team [LANGE86,GATTON87,BUCHNER88] two approaches have been developed in the application of these models. First is the concept of utilizing an Abstract Causal Domain Theory (ACDT) for a distributed knowledge acquisition (DKA) system. This work has been described in another paper at this conference [HARANDI&LANGE&KEARNEY88]. Second is the linkage of a Graphical Abstract Domain Model (GADM), to the production rule representation of the domain knowledge and its associated heuristics. This report deals with this, the second method.

Basically, a Graphical Abstract Domain Model (GADM) is a graphical representation of the elements in a physical system and its interconnections. For the purposes of this project, a GADM is limited to the representation of physical objects through elements at three hierarchical levels:

1) Assembly,
2) Sub-Assembly, and
3) Component.

This hierarchy is demonstrated in the breakdown of an automotive system shown here, in Figure 1, and is labelled to indicate assembly (A), sub-assembly (SA), and component (C) levels of the hierarchy:

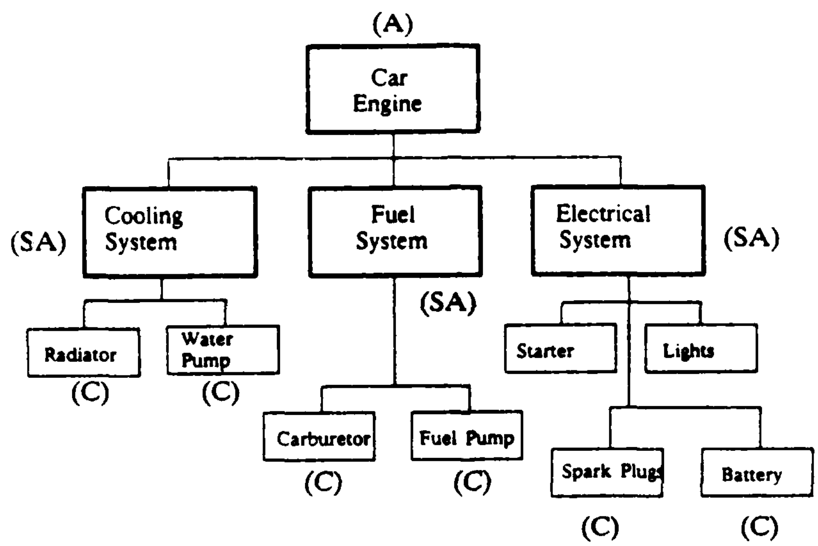


Figure 1

The elements are configured in a hierarchy that represents a simplified GADM of the levels of assembly, sub-assembly, and components that are present in an automobile motor. The element at the top level of the ADM is always an assembly, the elements at lowest level are always components, and those elements in between are always sub-assemblies.

The GADM shown in Figure 1 is related to the heuristics and knowledge that the expert mechanic has and is related to the logic or decision tree shown in Figure 2. This tree indicates the production rules that could be utilized to diagnose an engine that does not start.

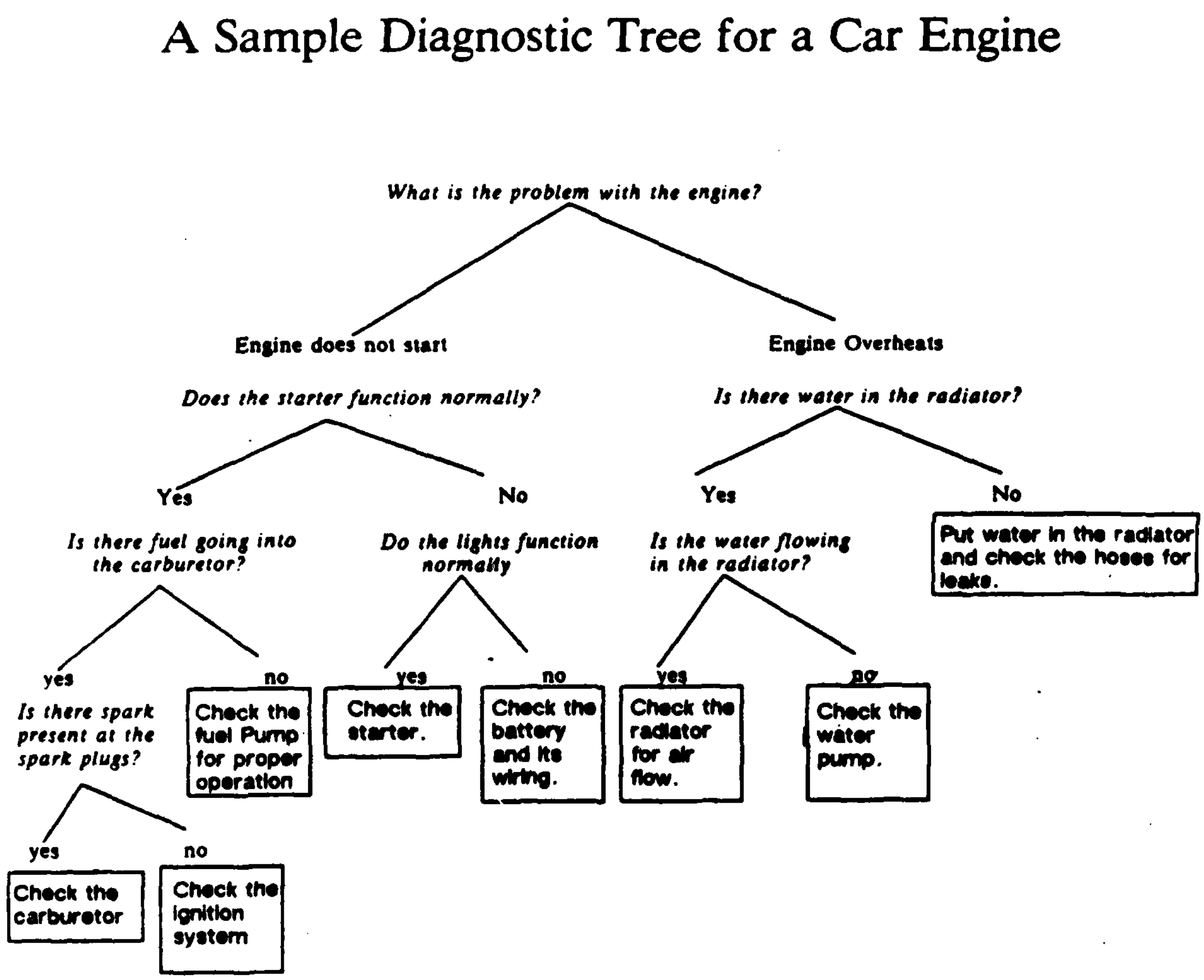


Figure 2.

Upon inpection of the structure of the GADM and comparison to the logic or decision tree, it is evident that following down the logic diagram and comparing them to the corresponding locations in the domain model seems to be a haphazard process. For example, consider a simple rule to determine faulty spark plugs:

```
IF the STARTER is NORMAL
   AND the GAS GAUGE indicates GAS
   AND the CARBURETOR has GAS
   AND the SPARK PLUGS have NO SPARK
THEN
   CHECK THE IGNITION SYSTEM
```

When comparing this rule to the GADM, the diagnostic procedures shown in the logic or decision tree have no direct relationship to the hierarchy of the GADM.

However, it is exactly this relationship that needs to be identified in order to acquire the diagnostic procedures from the expert. The process necessary to perform this task involves identification of individual rules and the ordering of those rules and their parameters to capture the expert's heuristics.

The first step is to identify the rules leading to each conclusion or diagnosis. This can be accomplished by prompting the expert for information about each component. This includes information about the possible conditions and related recommendations about component failures. Also, information about symptoms at the assembly and sub-assembly levels that are related to a faulty component would be gathered. This interaction will give information about the rules and recommendations. In order to determine the order of the rules, the expert will be asked, beginning at the assembly level, what the next check would be if certain symptoms were exhibited. This process is continued until the expert indicates that a check is being made on a component. Once this is done, the rule is known and the procedure is repeated until rules for all of the component failures are known. The order in which the information is gathered implicitly contain knowledge about what the heuristics of the expert are for diagnosis and allows subsequent ordering of rules and parameters in the knowledge base.

This information is gathered in a generic production rule representation to which shell specific generators can be interfaced for production of a knowledge base.

4. PROJECT STATUS

Currently, a prototype system has been developed to interactively input a domain model and subsequently interact with the expert for knowledge acquisition. The program will then utilize the domain knowledge and heuristics to generate a knowledge base for use with the CRITIC shell system. Presently, we are adding additional capabilities to modify an existing ADM and its associated information. This includes the following options:

1) Modification of the GADM,
2) Modification of diagnostic procedures,
3) Modification of explanations, and
4) Modification of recommendations.

Additional upgrades include an option to run the expert system for checking of logic, explanations, and recommendations as well as restoring previous versions of an GADM.

5. PRELIMINARY RESULTS

The prototype has indicated that the concept of linking a hierarchical model, or GADM, of a physical system with the production rules necessary to diagnose faults in that system is sound and offers strong potential for the automation of the knowledge acquisition process. The further extension of the prototype's capabilities and subsequent testing of the user interface with actual experts will be a critical step in making a user friendly system.

Work will continue in this area as well as research into the extension of the system to handle other types of knowledge representation besides production rules for diagnosis. Additionally, the application of this concept into the areas of design and construction, as well as the use of natural language interfacing, offers challenging opportunities for future research.

REFERENCES

Buchner, B.A., "A Domain Theory for Knowledge Acquisition," Master's Thesis, University of Illinois, Department of Computer Science, May, 1988.

Gaines, Brian R., "Knowledge Acquisition: Developments and Advances," in 'Expert Systems and Intelligent Manufacturing,' North-Holland, New York, 1988.

Gatton, T. M., "Decision Models for Expert System Development: Designing for Multi-Level User Domain Knowledge," Master of Computer Science Project, University of Illinois, 1987.

Harandi, M., Lange, R., & Kearney, F., "Distributed Knowledge Acquisition Using Abstract Domain Models," USA-CERL/ASCE Expert Systems Conference, June, 1988, Champaign, Illinois.

Lange, R., & Harandi, M.T., "The elements of a Distributed Knowledge Acquisition System," in the 'Proceedings of the 6th International Workshop on Expert Systems and Their Applications,' April, 1986, Avignon, France.

Distributed Knowledge Acquisition Using Abstract Domain Models

Mehdi T. Harandi, Rense Lange, Brian Buchner

Department of Computer Science

University of Illinois at Urbana-Champaign

and

Frank Kearney

ARMY Construction Engineering Research Lab., Champaign.

BACKGROUND

It has become clear that the success of expert systems is more a function of the power of the representation and the richness of their knowledge base than their choice of control schemes. An expert system depends for its "expertise" on large stores of domain-specific knowledge. Much time has to be spent in building and maintaining such large knowledge bases. As a result, the creation and management of knowledge bases have become an important research issue [Davis, 1978; Robinson, 1982; Lange & Harandi, 1986]. Of especial importance are the issues of knowledge acquisition and verification.

As a main component of an expert system building environment [Harandi 84], [Harandi 85], we have developed a rule base management facility which assists with the tasks of acquisition, manipulation and maintenance of knowledge [HaScCo 86]. This rule base manager organizes a global semantic information about the rule base. This semantic information is extracted from the texts included in rule structures. The effectiveness of this simple mechanism lead us to believe that it is possible to construct knowledge bases from scratch in a similar fashion.

OBJECTIVES

The objective is to construct a *Distributed Knowledge Acquisition System* that allows domain experts to assist in the construction/updating of knowledge bases. In this approach domain experts would communicate their know-how in an unstructured way and the system would discover the underlying structure of each knowledge unit, gradually building a knowledge base. The major purpose of the work is to reduce the involvement of knowledge engineers, thereby freeing some of the major scarce resources in the creation of expert systems.

Specifically, the method of *distributed knowledge acquisition* attempts to avoid some of the pitfalls of the traditional knowledge engineering methods. Our first concern is to diminish the role of the knowledge engineer, thereby gaining in overall efficiency. In addition, we feel that it is necessary for any realistic expert system to integrate different kinds of knowledge into a single knowledge base. To this end, the knowledge base should be accessible to different experts with various backgrounds and experiences. The method is called "distributed" because we envisage that experts can independently contribute small "knowledge units" to a **communal** knowledge base.

METHODOLOGY

Distributed knowledge acquisition basically creates a knowledge base by analyzing and integrating the information provided by several different experts. To start this process, the system is provided with a general knowledge representing an overall view of the domain. This might be done by inputing information obtained through traditional knowledge engineering methods. Alternatively, it might be possible to "borrow" relevant parts from already available knowledge bases in similar domains. Following this bootstrapping operation, experts interact directly with the knowledge base, using conventional network media and communication software.

Abstract Domain Models

It is one of the major functions of the acquisition system to elicit new information from the expert. Such information "makes sense" only to the extent that it can be understood in terms of the already available domain knowledge. Operationally, this means that the system has to construct a proof of the consistency of new information entered by the expert.

To provide this capability, the acquisition system uses qualitative knowledge of the domain, in the form of an **abstract domain model**. Similar to the approach followed in qualitative physics [DeKleer, 1975], abstract domain models allow the system to simulate the behavior of the domain. In the context of knowledge acquisition, this has the following advantages:

(1) Information can be stored according to its functionality and a consistent terminology can be enforced.

(2) The system can check the consistency of experts' inputs.

(3) The system can eliminate redundant solutions and identify alternative solutions.

(4) The system knows, to some extent, what it knows and does not know, therefore is capable of initiating queries which could lead to enhancement of the knowledge base.

STATUS OF THE WORK

As a first step towards this system we have developed a domain model representation scheme which we call *Abstract Causal Domain Theory.* Our domain theory models the functional behavior, in terms of causation between the components, or entities, of the domain. Each component has an associated knowledge structure which describes its functional behavior. Interactions between components are modeled with qualitatively valued causal flows. Causal equations describe the input–output relationships of a component in terms of these qualitative values.

An interactive domain editor has been developed which allows experts to define and maintain domain models in efficient and transparent fashion. The ICE system includes a graphic interface to provide the user with a natural visual representation of the knowledge structures he/she is editing. In this interface icons are used to represent domain components. Components can be selected and connected together to form an entire (sub–) model. The same editing features can be used to update or modify an existing model. Alternatively, a selected component may be edited to change its functional description.

REFERENCES

Davis, R. Knowledge Acquisition in Rule - Based Systems — Knowledge About Representations as a Basis for System Construction and Maintenance. In: "Pattern Directed Inference Systems", Waterman, D.A. and Hayes - Roth, F. (eds), (1978), pp. 99–134.

DeKleer, J. Qualitative and Quantitative Knowledge in Classical Mechanics. MIT AI TR–352, 1975.

Harandi, M.T. A Tree–based Knowledge Representation Scheme for Diagnostic Expert Systems. Proceedings of the 2nd Conference on Intelligent Systems and Machines, Rochester, Mi, 1984.

Harandi, M.T. The Architecture of an Expert System Environment. Proceedings of the 5th International Workshop on Expert Systems and Their Applications, Avignon, France, 1985.

Harandi, M.T., Schang, T., Cohen, S. Rule Base Management Using Meta Knowledge. Proceedings of the ACM SIGMOD International Conference on Management of Data, Washington D.C., 1986.

Lange, R., M. T. Harandi, The Elements of a Distributed Knowledge Acquisition System. Proceedings of the 6th International Workshop on Expert Systems and Their Applications, Avignon, France, 1986.

Robinson, J. Problems and Trends for the Future of Logic Programming. In: "Introductory Readings in Expert Systems", Michie, D. (ed), (1982).

A CONTROL STRUCTURE FOR MICRO-BASED DESIGN EXPERT SYSTEMS USING THE SHELL EXSYS

Roswell A. Harris and Louis F. Cohn[1]

INTRODUCTION

The expert system CHINA (Computerized HIghway Noise Analyst) was originally developed to address a nationwide lack of experience in the acoustic design of highway noise barriers. The initial version of this system was developed on a VAX 11/780 using FRANZ LISP and a development tool called GENIE.

One of the primary goals of that research was to make the body of knowledge in the acoustic design of highway noise barriers easily accessible to those engineers involved with such work. Although the prototype produced acceptable results, efforts to refine the knowledge base and enhance the performance of the system were not aggressively pursued because of the limited availability of the VAX system to those engineers.

Since the original development of CHINA, industry emphasis has moved from the need for tools on mainframe computers to the need for tools in a microcomputer environment. This can logically be attributed to the low cost, improved speed, increased storage and memory capacity, and widespread availability of microcomputers. As a result, many expert systems related tools have been developed for the personal computer.

The research described in this paper relates primarily to the problems associated with implementing a rule based design oriented expert system in a microcomputer environment. The specific issues to be addressed are:

- Development of data conversion utilities to provide correct data transfer between CHINA and external programs;
- Development of a control structure that invokes the expert system;
- Development of an interface to an external graphics program written in BASIC; and,
- Development of a user interface that provides easy access to the expert system, the rule base editor, system parameters, and crucial design data.

The tool chosen for this project was EXSYS, developed by EXSYS, Inc. of Albuquerque, New Mexico. EXSYS is an expert system development shell written in C which encodes knowledge in the form of IF-THEN production rules.

METHODOLOGY

The FORTRAN program OPTIMA was modified to allow integration with the expert system. This involved the establishment of a flag system as well as the importing and exporting of parameters and design data. The OPTIMA output is not written in a format that is directly compatible with EXSYS. Instead, the data was written to an intermediate file and a routine developed to convert it to a format which is readable by EXSYS. Similarly, a routine was written to convert the output from EXSYS to a format that could be used by OPTIMA for subsequent iterations of the program. This approach was used so that future enhancements to the systems could be accomplished without further modification to the OPTIMA code.

During initial development, a simple batch file was used as the control structure for executing each module of the system. Once the system was functioning as intended, the design of a more sophisticated control structure was undertaken. The role of this control module was three-fold: (1) to invoke each required program and maintain data files regardless of their location in the DOS directory; (2) to provide redirected input where desirable; and (3) to maintain a virtual disk on those systems which can support it.

It may not always be desirable for all programs and data to be located in the same DOS directory. For example, the user may choose to place the EXSYS programs in a unique directory for use with other expert systems. Another case may be that CHINA is to operate on data files that are stored on a floppy disk. For such configurations, a control system would have to be able to access programs and data wherever they are located.

The CHINA rule base operates on many sets of data during each pass of the design cycle, prompting the user for an action when it has applied the rules for each set. For a large problem this would be both tedious and time consuming. It is therefore desirable to redirect the standard input to come from a file which contains acceptable responses to the EXSYS prompts. For this reason, the control module must be able to redirect input when appropriate.

In addition to issuing a prompt with each pass through the data, the rule base must read each set of data from disk as well as write the output disk. For a large problem, this would generate a substantial amount of disk I/O which would considerably slow the design process. This problem can be significantly improved by copying the rule base and the rule base variable file to a virtual (RAM) disk. As a result, all input and output would be read from and written to RAM, a process which is significantly faster than disk I/O. Thus virtual disk maintenance is another important task of the control system.

The language chosen for the control module and most of the other program modules was PASCAL. The primary reason for choosing PASCAL was to provide programs which could be easily revised by any user. The availability of TURBO PASCAL documentation, and its popularity with both advanced and novice programmers made it an obvious choice. Also, TURBO PASCAL provides easy access to MS-DOS function calls, a very desirable feature for loading and executing non-PASCAL files, for redirecting input, and for moving through the directory structure. Flexible file access tools provided in PASCAL are of great benefit since the same data must be read and written by FORTRAN, EXSYS, and BASIC programs. Lastly, PASCAL provides many screen formatting tools which allow an aesthetic display to be generated.

One of the primary functions of the control module is to provide a routine which could load any external program, pass to it command line parameters, and redirect the standard input without destroying the control program residing in memory. In order to maintain compatibility among various systems, most compilers and libraries do not provide such a function. Thus one of the first and most important tasks in developing the control module was to create a module to carry out this task.

A PASCAL procedure was written to complete this task. The program receives two parameters from the calling routine. The first is the name of the external function to execute. The second is the command line parameters. This parameter is examined for redirection arrows. If found, the characters following the arrows are examined for a file name, and the standard input is redirected to come from this file through a process which duplicates the standard input file handle, and assigns the user specified file to the standard input handle. Once this has been completed, the TURBO PASCAL MS-DOS function is invoked to load and execute the external program. This module can be used to load and execute any external program, if sufficient memory is available.

With completion of the control module, the next task was to develop a user interface that would provide truly useful design tool. In addition to the modules discussed above, the system must know the answer to several questions in order to run as intended. For example, CHINA must know the answers to the following questions: Where are the programs and data located? Will a virtual disk be used? Is the standard input to be redirected during the rule base pass? A powerful user interface should also provide design aids such as a graphics interface, design data printouts, the ability to change crucial design variables, and to revise or view the rule base.

SUMMARY

The program modules described above provide a very effective means of invoking and controlling a design oriented expert system. These modules provide the user with a friendly interface, as well as providing a means to completely control operation of the system. An interface is also provided to a graphics routine which provides the user with a graphic representation of the final barrier design. In addition, the interface provides the option of viewing final design information either on the screen or producing a hard copy. This system has proved to be very powerful, while remaining easy to use for the unfamiliar user. In addition, it is designed to be flexible, so that as the knowledge base grows, or the expert system is otherwise enhanced, these changes may be easily accommodated.

[1]Department of Civil Engineering, Univeristy of Louisville, Louisville, KY

AN EXPERT SYSTEM FOR CONSTRUCTION CONTRACT CLAIMS

Moonja Park Kim

SUBMITTED FOR PRESENTATION

AT THE FIRST CERL-ASCE JOINT EXPERT SYSTEMS CONFERENCE

JUNE 29-30, 1988

AN EXPERT SYSTEM FOR CONSTRUCTION CONTRACT CLAIMS

Moonja Park Kim[1]

INTRODUCTION

The expert system technology available today on microcomputers makes it possible to address a significant problem facing the construction industry: the need for expertise in construction claim analysis at the field level. Construction in the 1980's has become a very complicated industry, with many intertwined relationships and intense competition. There seems to be great potential for application of the state-of-the-art knowledge in expert system technology to the practical areas of construction. One promising area is using an expert system to help minimize some of these problems by providing field personnel guidance in handling legal issues related to potential claims.

The professionals in the legal field are also taking advantage of this new technology, as evidenced at the First International Conference on Artificial Intelligence and Law held May 27-29, 1987. Some law firms are even creating expert system groups to perform in-house research and development. For example, Watt, Tiedler, Killian and Hoffar, a law firm in Virginia, is developing a microcomputer expert system for claim identification and evaluation (Lester, 1987). The applications of this technology will assist lawyers in sorting out pertinent information for efficient discussion with the client.

Researchers at the U.S. Army Construction Engineering Research Laboratory (USA-CERL) have been developing an expert system called Claims Guidance System (CGS) to provide claims analysis expertise at the field level of the Corps of Engineers. This system uses an expert system shell for IBM-compatible microcomputers, Personal Consultant Plus by Texas Instruments. The objectives of this system are (1) to provide an inexperienced project engineer with prelegal

---

[1] Principal Investigator, US Army Construction Engineering Research Laboratory, Champaign, IL.

assistance in the analysis of potential claims from construction contracts and (2) to serve as a training device for new personnel in field offices, familiarizing them with the related legal issues.

The first module of the Claims Guidance System (CGS-DSC) will guide project engineers in analyzing "Differing Site Condition" (DSC) claims. Unknown subsurface conditions or latent physical conditions at the work site represent a very significant risk inherent in many construction contracts. The DSC contract clause represents an effort by the U.S. Government to reduce the risk to the construction contractors of such unknown or unanticipated conditions. This clause allows contractors to submit their bids based on reasonably foreseeable conditions, without contingencies to cover the unexpected or unusual. In return, the bidder is assured that in the event conditions prove different than should have been anticipated, an equitable adjustment will be made in the contract price and/or duration. Without this clause, the contractors' only alternative, in order to meet the requirement for submitting a fixed price, was to include contingency allowances in their bids to cover the cost of coping with possible subsurface difficulties, which in fact may not have occurred during subsequent performance of the contract. AS a result, the Government paid more than the actual work was reasonably worth.

Studies by Mogren (1986) and Diekmann & Nelson (1986) have shown that DSC claims are one of the most costly reasons for changes in U.S. Army Corps of Engineers construction contracts. Corps field engineers who are faced with such claims need to understand the legal issues involved so that they can supply the proper information to legal counsel and avoid lengthy litigations caused by incorrect decisions. Personnel who are unfamiliar with this process must rely on experienced engineers for help in analyzing a claim. The expert system for Claims Guidance is to provide the expertise of the experienced engineers in dealing with construction contract claims. This paper describes the development of the CGS-DSC.

METHODOLOGY

From an analysis of the Differing Site Condition Clause (FAR 52.236-2) used by the U.S. Government in its contracts, the following important issues in dealing with DSC claims are identified:

1. Final payment
2. Notice Requirements
3. Prejudice to Government
4. Government Action
5. Contract Provision (Type I or Type II)
6. Acceptable and prudent

In addition to the analysis of the DSC clause, the work of Dickmann and Kruppenbacher (1984) was considered, and their logic diagram was reviewed by an experienced Corps field engineer and was revised and simplified to fit the Corps office environment. Using the revised logic diagram and questions, rules were developed to create a test version of the CGS-DSC. Then a steering committee was formed to review the test version and to evaluate it for validity and completeness. The committee consisted of six experts: two experienced legal counsels from the Corps headquarters and four engineers with many years of experience in construction contract management within the Corps. The committee suggested many enhancements and necessary corrections to the logic diagram and identified the following additional issues as important in handling DSC claims :

7. Reliance on Contract Indication
8. Superior knowledge
9. Nature of Condition (Act of God)
10. Site Inspection.
11. Material(Substantial) Difference
12. Exculpatory Language
13. Anticipation: Usual and Known

Two legal case retrieval systems, Lexis[2] and Weslaw[3], were used to select appropriate cases which represent some of the issues listed above. Twenty-three

---

[2] Lexis is a Trademark of Head Data Central,Inc.

[3] Westlaw is a trademark of West Publishing Company.

relevant cases were retrieved. These cases were examined carefully in terms of the 13 issues and of the rationale for the decision of the Board of Contract Appeals and/or the Court of Claims.

Personal Consultant Plus was selected as the expert system shell for the development of the system in order to meet the limitations: (1) Consultation should be available on IBM PC/AT compatible computer with 640K memory, (2) Cost of developing and delivering system should be minimized. As we added rules to the knowledge-base, it was necessary to add 1.5 megabytes of memory for the development stage. However, every effort was made to make the delivery system run with 640K memory, by creating disk-resident text files for help screens and case summaries. As shown on Figure 1, the CGS-DSC delivery system environment was created by displaying text files for help screens, by presenting PC Paint graphic images for progress checks, and by running dBASE III Plus to search for the appropriate case name of the text file to be displayed and to generate a useful report for the users by organizing the information clearly and presenting in a fact sheet. This report generation is performed by dBASE III Plus because Personal Consultant Plus does not provide adequate reports.

At any time during the consultation before answering a question, the user can display a help screen for explanation of the question, as additional clarification. The user can also invoke the WHY option requesting to explain why the system is prompting for this information. An example of WHY screen is shown on Figure 2. At various points of the consultation, a graphic display of progress through the system is available. An example of this graphic display after checking the Government action is shown in Figure 3. This graphic display is an over-simplified version of the CGS-DSC logic diagram. It informs the user how many of the issues were covered through the system at various stages.

At the end of consultation, the following conclusion screen will be displayed.

```
CONCLUSIONS

Possible conclusions for entitlement are the
following:
    1. VERY-POOR-CHANCE
    2. POOR-CHANCE
    3. DIFFICULT-TO-DECIDE
    4. FAIR-CHANCE
    5. GOOD-CHANCE
    6. EXCELLENT-CHANCE

The chance of entitlement for this contractor
is as follows:  VERY-POOR-CHANCE
```

After displaying the conclusion reached by the CGS, the system will search for a case similar to the situation described by the user and the short summary of the retrieved case from cases database would be displayed. Then, the list of the pertinent factors in reaching the conclusion is displayed on the screen. After this, the user needs to save all his answers in a file in order to review it later, and he can print a report if he wants to keep a hard copy of fact sheet.

STATUS OF THE PROJECT

The field test version was delivered to two test sites with proper training in November 1987: at Fort Drum Area Office and New Orleans District. A one-year field test is planned; however, suggestions from the users will be incorporated as they are received. Then the final system will be distributed to a number of field offices for regular use.

The next steering committee is planned to meet in May 1988 to review the usage from field test sites and to plan the additional modules. Informal discussions with some of the users at the field test site indicate that the system is beneficial in handling potential differing site condition cases, but the potential cases do not occur very frequently. Therefore, in order to get more cases for the field testing, two additional field test sites were selected: at Fort Benjamin Harrison Area Office and

George AFB Resident Office. The results of these field tests will be available in September 1988.

FUTURE RESEARCH

As a continuing effort in developing the Claims Guidance System that will be most beneficial to field engineers, a preliminary investigation was performed to find the types of claims occurring frequently and found that claims resulting from contract interpretation disputes would provide the most benefit. Background research on this area to learn about basic legal issues involved in this type of claim was conducted and several cases were reviewed. Then the simplified prototype for this type of claims was developed as the second module of CGS and will be reviewed by the CGS Steering Committee in May 1988. Then the field test version for the second module will be developed and added to the first module at the field test sites by September 1988.

It seems that a great amount of research and development is expected in the near future, as evidenced at the First International Conference on Artificial Intelligence and Law held May 27-29, 1987. We may take advantage of this interest in the legal field and include more legal expertise in improving the CGS to include other types of construction contract claims. For example, including construction delay claims would involve integrating scheduling and network analysis with legal evaluation of claims. Design deficiency claims would involve integrating CADD systems with claims evaluation to examine drawings for deficiencies. These areas are challenging and hold potential benefit for the construction community.

REFERENCES

Dickmann, J. E., and Kruppenbacher, T. A. (1984) "Claims Analysis and Computer reasoning." Journal of Construction Engineering and Management, Vol. 110, No. 4, ASCE, 391-408.

Dickmann, J. E., and Nelson, M. C. (1985) "Construction Claims: Frequencies and Severity." Journal of Construction Engineering and Management, Vol. 111, No. 1, ASCE, 74-81.

Kruppenbacher, T. A. (1984) The Application of Artificial Intelligence to Contract Management. USA-CERL Technical Manuscript P-166 (U.S. Army Construction Engineering Research Laboratory).

Lester, J. L. (1987) "Lawyer on a Microchip." Civil Engineering Magazine ASCE, 68-69.

Mogren E. T. (1986) "The Causes and Costs of Modification to Military Construction." Master Thesis, U.S. Army Command and General Staff College.

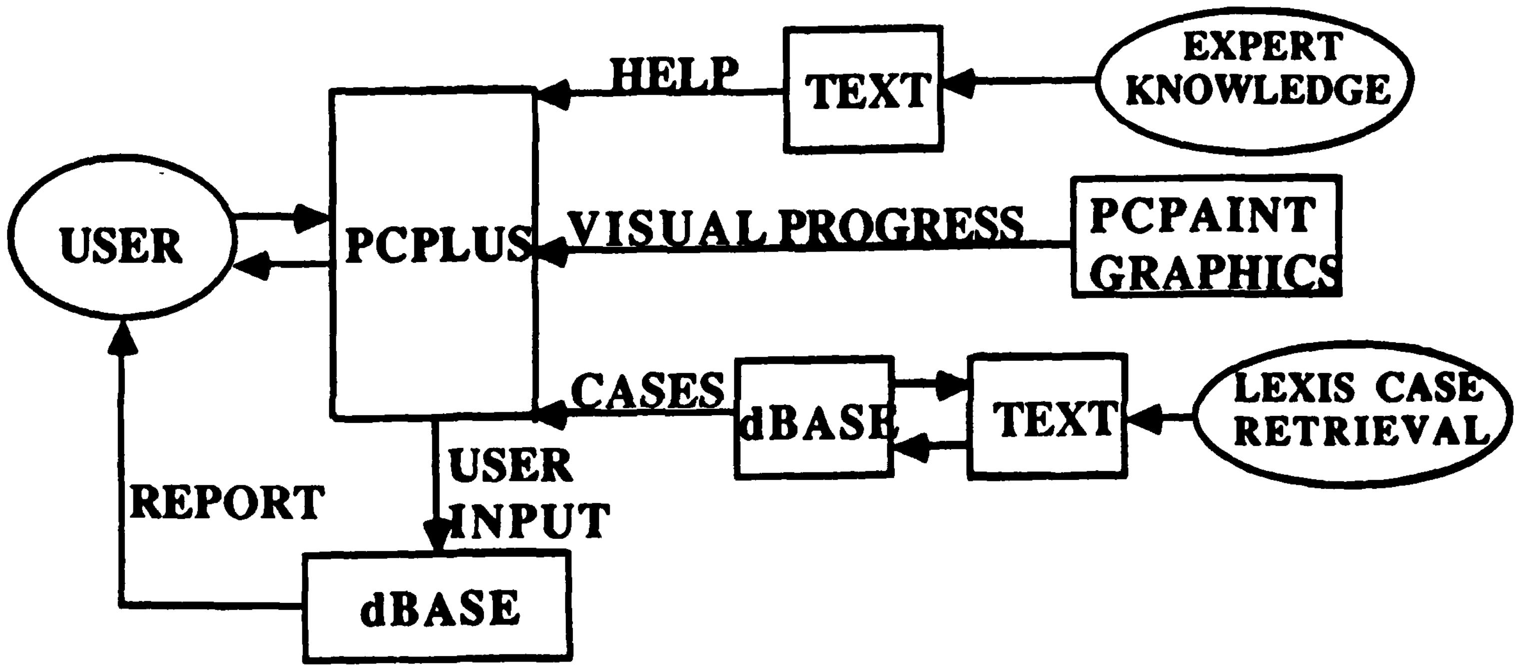


FIGURE 1. CGS-DSC ENVIRONMENT

WHY

Who was the first person to receive the notice is needed to determine if responsible person did receive the notice.

RULE 008
In order to check if the contractor complied fully with the requirements for the notice the following information is necessary:

(1) if responsible personnel received the notice and the date received,
(2) if the form of notice was adequate (written or oral)
(3) if the notice was prompt.

** End - RETURN/ENTER to continue

FIGURE 2. EXAMPLE OF WHY SCREEN

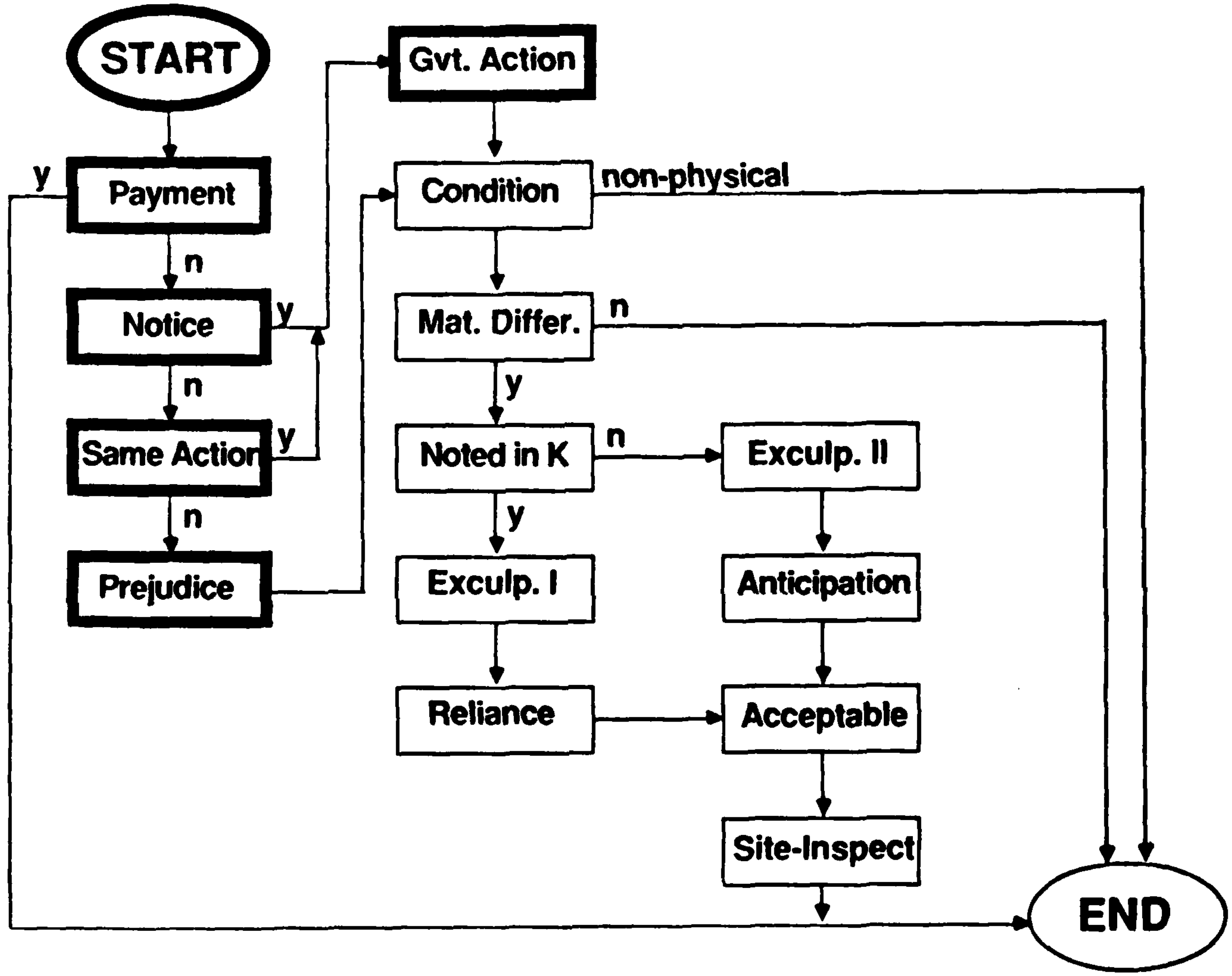


FIGURE 3. SIMPLIFIED LOGIC DIAGRAM FOR CGS-DSC

Expert System For Asphalt Paving

Philip W. Lawrence[1] and Frank Kearney[2]

NATURE OF THE PROBLEM

The Federal Highway Administration is charged with the problem of maintaining and rehabilitating the Nation's roadway system. Recently, the FHWA has been experiencing a shortage of experts in the field of transportation. An estimated one-third of the transportation professionals presently employed by the FHWA will retire within the next five years.

Due to the turnover rate of the experts in the area of transportation, inexperienced people are having to serve as construction inspectors. The training of the new personnel must be rapid in order for the quality of the Nation's roadways to be maintained. Assistance in the form of problem diagnostics and instruction on inspection procedures must be given to the new inspectors as they learn what is necessary to perform the tasks assigned to them.

In order to facilitate the training of the inexperienced construction inspectors and to diagnose problems encountered in the field, the knowledge of the transportation professionals should be institutionalized for recall by the new personnel. This should be

---

[1]Research Assistant, US Army Construction Engineering Research Laboratory, Champaign, Illinois.

[2]MQA Team Leader, US Army Construction Engineering Research Laboratory, Champaign, Illinois.

done in such a manner as to help the instructor diagnose problems encountered in the field and/or to instruct the inspector on proper inspection procedures.

OBJECTIVES

An expert system is being written to institutionalize the FHWA transportation professionals' knowledge in the field of asphalt paving. The system, called "Expert System for Asphalt Paving" ("ESAP"), should act as an expert and suggest possible corrective measures to problems encountered by the construction inspector. ESAP should also help facilitate the training of the inspector by instructing for proper inspection procedures.

ESAP is intended to cover the asphalt paving process from the delivery of the hot-mix paving material to the compacting of the final mat.

METHODOLOGY

Before the knowledge acquisition for the knowledge bases can begin, an agreed-upon breakdown of the domain into appropriate subdomains must be made. It is essential to decide on the breakdown and not make any revisions. Any changes to the breakdown of the subdomains will cause delays to the knowledge acquisition process.

It is the Materials and Quality Assurance Team's philosophy that the knowledge acquisition for expert systems should be an iterative process. More than one expert will, therefore, have to be consulted for each module. However, before a single expert's input is entered into the system, a working prototype has to be completed.

The working prototype should be assembled from basic knowledge found in manuals written on the expert system's domain. The domain should be divided into appropriate subdomains, thereby modularizing the knowledge bases. The initial versions of the modules will facilitate the knowledge acquisition process in a number of ways. Experts are usually more receptive to making changes to an existing system as opposed to building a system from the ground up. For experts to structure their input into the system, it is important for them to have an understanding of the direction and intended content. It is also easier for the knowledge engineer to make changes to an existing system.

The knowledge bases are to be structured for use with a shell developed by the MQA team. The shell (called CRITIC) uses four text files, one of which has to be compiled to be compatible with the program.

Once the prototype of a module is completed, the iterative process with the domain experts can begin. The expert must first become familiar with the purpose and intent of the system, and an introduction to expert systems must be given to the expert. For the first iteration, it is helpful for knowledge acquisition if the content of the existing system is displayed graphically, in the form of logic trees, to the expert. If the logic trees have been drawn, the knowledge engineer can sit with the expert and go through each tree individually and make any necessary changes.

These changes should be entered into the system; then the expert and knowledge engineer should go through the program and make changes until they are satisfied with the content of the system. If the logic trees can not be represented graphically, the two should just go through the system together and make necessary changes.

For subsequent iterations, the knowledge engineer and expert should go through the system (run through a consultation session), note any changes to be made to the system, and implement these changes.

STATUS OF THE WORK

ESAP is broken up into three parts: (1) diagnosing problems in hot-mix asphalt paving, (2) instructing the plant inspector for proper inspection techniques in the plant, and (3) instructing the roadway inspector for proper inspection techniques on the roadway. Each of these three parts are further broken up into a number of modules. Overall, ESAP contains eleven modules. Of these modules, nine have preliminary versions, three of which have had expert input (the first iteration). The other two modules have not been started.

The modules with expert input are: the problem diagnosing knowledge base for compaction of the asphalt hot-mix mat, and the modules instructing the roadway inspector for proper inspection of the laydown and compaction of asphalt hot-mix. These modules have been given to the FHWA for field testing.

PRELIMINARY RESULTS

The FHWA is pleased with the quality of the compaction module and field testing is going well. Having one module that was ready for field testing was essential for the FHWA's perception of the overall system. Upon receiving the compaction module the FHWA showed renewed interest in expert system research.

The knowledge acquisition process went smoothly, and proved effective. Refinements to this process are expected to accompany future sessions with the experts.

EXPECTED FUTURE RESEARCH

Preliminary results indicate that it would be wise to pursue one area of asphalt paving at a time. For this reason, we have decided to concentrate our efforts on the compaction subdomain. This would include the module currently containing expert input (the compaction module) as well as another module that is intended to instruct the user on proper inspection of the compaction process. We would like to have these two modules ready for field testing before any other work is started.

Eventually the expert system should contain eleven modules. The method of knowledge acquisition mentioned before should be implemented to all modules.

References

Lange, R., Hearn, L., Kearney, F., "Applications of Artificial Intelligence in Engineering Problems" Berlin, Springer, 1986.

INTEGRATED KNOWLEDGE-BASED SYSTEM FOR STRUCTURAL DESIGN

T. J. Ross[1] and F. S. Wong[2]

ABSTRACT

This abstract describes the on-going development of an integrated knowledge-based system for the design of structural facilities subjected to hazardous loadings. This development is a collaboration of the authors[1,2] under sponsorship from the Department of Defense. This engineering knowledge-based system explicitly collects together in one place: information from design hand-books, findings of recent research efforts, any expert knowledge, and stochastic and other uncertainty methods. The integrated knowledge-based system (denoted ***PSDesign***) will be capable of addressing three kinds of design-applications: (i) New designs, (ii) Retrofitted designs, and (iii) Post-damage designs.

Introduction

Structures designed to withstand the effects of hazardous loads (wind, earthquake, etc.) are built primarily according to deterministic design procedures. These procedures assume precise knowledge about the parameters that play a significant role in the structure's final design. Real-world variabilities in site characteristics, structural attributes like strength and stiffness, and loading characteristics are generally not accounted for in current design schemes. In fact current design procedures are overly conservative in that, to ensure high confidence in sustaining the facilities' mission, they presume a "worst case scenario" in selecting values for design parameters. For example, some typical "worst case" presumptions would involve underestimating structural strength, overestimating the loading imparted to the structure, ignoring complex details like three-dimensional and nonlinear effects, overestimating joint stiffness, and ignoring issues that typically are not quantitative in nature. Examples of the latter would include construction quality control, weather conditions during construction, ad-hoc construction changes and short-cuts, and the validity of the design procedure to emulate real-world situations. In addition, significant information which relates to the design process such as expert rules-of-practice, recent research findings, local code provisions, experimental data, and others typically are not included in the design algorithms in an explicit manner, but are rather used to assess the design outcome in an ad-hoc, intuitive fashion.

Based on the problem described here it was evident that a balanced (in the sense of accommodating various sources of data, knowledge, and design methods) and effective design tool was needed. This new tool would address random and nonrandom variability and would be flexible in use and adaptable to changes in user requirements. The integrated system would provide for various design applications: new construction, retrofitted construction, and post-damage construction (i.e. after a hazardous event). The flowchart in Figure 1 illustrates, conceptually, the scope of the project to develop an integrated knowledge-based design system.

---

[1] Associate Professor, Dept. of Civil Engineering, University of New Mexico, Albuquerque, NM 87131

[2] Senior Research Associate, Weidlinger Associates, 4410 El Camino, Suite 110, Los Altos, CA 94022

The integrated design procedure is predicated on the notion that the design will have four primary constraints, as shown in the schematic in Figure 2. Typically, a designer is given the location of the new facility (so he has an idea about local code requirements, etc.), its approximate budget, and an idea of the design criteria due to the facilities' function (a hospital should have different criteria than an industrial office building). The exact extent of the hazardous loading is something that may not be given directly to the designer, depending on the circumstances. The design problem can be thought of as: enhancing the operability of the facility while minimizing costs or repair time.

Also shown in Figure 2 are the three kinds of design-applications to be addressed by ***PSDesign***. The table shown below highlights some of the issues to be considered in each of these applications. ***PSDesign*** will be developed to take advantage of many of the same modules to be shared according to the application of interest. For example, a library of geometric shapes, materials, and loadings can all be shared by each of the applications. These applications are all important to the Defense Department in its efforts to modernize facilities (new construction), in its efforts to upgrade existing facilities (retrofitted construction), and in its efforts to achieve efficient repair strategies in the event of damage from a hazardous event (post-damage construction).

PSDesign **Applications**

New Designs	Retrofit Designs	Post-Damage Designs
• Preliminary design	• Structural Modification	• Damage Assessment
• Floor space	• New materials	• Expedient repair
• Costs	• Interrupted operations	• Reconstruction

Advantages of an Integrated Approach

There are at least four advantages of an integrated design system where numerical computing is combined with symbolic processing. The first advantage is the extension and enhancement of the capabilities of existing numerical design algorithms. Often times, even a complete mathematical model is inadequate for several tasks. The integration (or coupling) of these programs can bring several of these mathematical models together to satisfy the requirements of a multi-faceted design project.

A second advantage is the intelligent user interface, which is self-explanatory. Typical functions include helping the user to select and use numerical processes (intelligent front ends) and assisting the user to interpret the results from numerical processes.

A third advantage is that of learning and induction. A coupled system may be designed to extract new knowledge from numerical processes and data. Typical applications are: extracting classification and relational rules and structures from test and model data and extracting knowledge from a sensitivity or parametric analysis of numerical simulation and numerical algorithms.

The fourth advantage is that of intelligent processing. This advantage is different from the first, in that the system to be developed is aimed at reducing the expense of computational tasks performed by the traditional numerical processes. These applications are the opposite of those developed to extend system capabilities. For example, many number-crunching intensive problems cannot be solved because of insufficient computing resources due to the size of the program, number of cycles or iterations, or the number of combination or inputs is simply too large. Intelligent processing is aimed at reducing the computing resources required by streamlining the program, its operation and inputs.

The unique feature of the proposed ***PSDesign*** is that it can incorporate many sources of information and different kinds of information about the design process. The current work has shown how: (i) information from design hand-books, (ii) findings of recent research efforts, (iii) any expert knowledge, and (iv) stochastic methods, can be collected together in one place in an integrated form. For the novice user, a short tutorial can be used to bring him up to date. For the infrequent user, the design procedures can guide him along paths which he may have forgotten. For the experienced user, the system will offer another level of sophistication in the form of the stochastic module; uncertainties in the design strength and how it relates to the real-world behavior are important in structural design but yet not readily obvious from the design equations.

In the integration of numeric and symbolic (non-numeric or non-algorithmic) information one must consider not only the programming convenience of the integration, but also the proper use of the analytic and data elements within the numeric and symbolic computer programs. For example, in an earlier study by the second author both the integration and linking of graphical data and analysis elements were implemented in a conventional procedural programming environment using FORTRAN. The analytic elements were FORTRAN codes and the integrating software was also written in FORTRAN. Recent integration efforts have improved on the earlier conventional approaches by using either a commercially available expert system "shell" (Ross's DAPS code) or the symbolic language LISP (Wong's earlier study) as the programming language of integration. The versatility of LISP enables reasoning to be made about the proper usage of the analytic and data elements within a design problem, which is difficult to do within the procedural FORTRAN environment. By being able to apply some common-sense reasoning to the analytic elements and to interpret the results from each analytic module, a certain amount of intelligence and sophistication is added to the overall structural design problem. An example of the use of reasoning within the design problem would be on-line guides as to why certain material models may be appropriate or which shear failure criteria is applicable in a given situation. Some reasoning capability also becomes extremely useful in the area of conflict resolution. Often times, the use of different design approaches provide conflicting solutions for the same set of input data and a reasoning capability could assist the designer in deciding the most appropriate design for his situation.

The integration of numeric and symbolic elements can be rephrased as the integration of non-algorithmic design knowledge with existing/improved algorithmic design codes. Information about the design of structures subjected to hazardous loads comes from several sources: (i) active and passive measurements from research tests; (ii) computational results from preliminary designs, (iii) design handbooks and similar local or nationally sanctioned codes; (iv) experience and judgment from expert designers; and (v) historic data from previous designs of a similar nature, which exist primarily in the form of photographs, blueprints, drawings or other visual images.

Data Structure and Object-Oriented Knowledge Representation

The basic data/knowledge representation scheme of ***PSDesign*** is an object-oriented data structure formulated in the computer language LISP. In this approach, a knowledge base (kb) is an object which, in turn, is composed of rules-of-thumb for design (rule) and design data (data). Both the (rule) and (data) information can be thought of as *objects*. Each *object* can have a name, an attribute, and a value.

For example, the schematic shown in Figure 3 illustrates the object-oriented structure as a knowledge-representation scheme for a slab design. In this schematic the (kb) object is slab-design. Further, the data-objects within slab-design could have names like strength, aspect-ratio, and thickness. The data-object strength could have two attributes: concrete and steel; and each of these attributes would have values, such as 4000 psi and 60,000 psi, respectively. The rule-objects within slab-design, in Figure 3, would also be characterized by names, attributes and values. For example, a rule called shear-strength may be used to query the user about which design philosophy to use for assessing web reinforcement. Attributes of this rule could be: ACI-rule (American Concrete Institute philosophy), ECC-rule (European Concrete Committee philosophy), and others. The values of these attributes would also be specific rules; e. g. IF ACI criteria, and IF the beam does not have web reinforcement, THEN the first diagonal cracking shear is the shear carried by the concrete only.

Uncertainties in Design

Due to uncertainties about the response of structural components to many loading environments, the uncertainty in the recurrence of the hazardous event (e.g., earthquakes), the lack of adequate experimental data and proper physics relations, the information from any of the possible sources of information is almost never complete by itself. When used separately the pieces of information produce a very limited structural design. When combined, the various sources of information complement one another to enrich the design process. Uncertainty in the design process is usually reduced as information from more than one source is introduced or better data or physical relations are developed for evaluation purposes.

Therefore, it is highly desirable to have a design system which integrates information from various useful sources and which can assess the impact of uncertainty in the existing information and which can estimate the uncertainty due to missing information. Currently, much of the needed integration is being done manually, through intermediaries in the from of experienced designers. The degree of success of this approach naturally depends on the expertise level of the designer, his ability to recall from memory the relevant data, and his skill in putting the pieces together. Especially in preliminary designs, it is not uncommon to have designs produced by design staff members who do not have the experience in advanced structural dynamics necessary for a realistic and economic design and who, routinely, are forced to place blind trust in manuals and procedures they do not fully understand. Typically, there is a large gap between a preliminary design used for planning purposes and the final constructed design, the latter usually being accomplished by more experienced practitioners.

The representation of uncertainty in ***PSDesign*** is implemented into the object-oriented data/knowledge structure. This is easily accomplished when one uses a general representation to model uncertainties as being either probabilistic, fuzzy, or evidence. The latter two are related to nonrandom uncertainty while the first is used to represent random uncertainty. Such a general representation can be found in the simple, yet powerful, interval method. In fact, all the popular methods listed below have been shown by the authors in earlier works to be various forms of the interval model.

- simple intervals
- possibility measures
- probability measures
- evidence measures
- fuzzy sets

Simple intervals correspond to the least amount of information available. The value of a parameter, such as the depth of a geologic layer, is estimated to be within an interval [a,b], and nothing more is known. Possibility measures are built upon simple interval uncertainties but with varying degrees of possibilities; hence, they correspond to slightly more information on the value distribution of the parameter. When the probability of a particular value of the parameter is known (for all possible values of the parameter), then this is the probability measure of the uncertainty. Probability measure implies exact knowledge of the uncertainty. Although the value of the parameter is uncertain, there is no uncertainty about the character of its randomness; it is described exactly by a probability distribution. This may be unreasonable and inappropriate in certain situations. For example, knowing the probability of event A, denoted by p(A), implies that the probability of the event (not A), denoted by p(not A), is 1 - p(A). In other words, if there is evidence p(A) supporting A, then there is evidence refuting A of the amount 1 - p(A).

This stringent presumption of probability theory can be relaxed by introducing the concept of ignorance, as in the theory of evidence. Knowing p(A) should not say anything about p(not A) unless there is explicit evidence supporting "not A". When such evidence is not available, p(not A) is zero. There is ignorance as to whether there is something against A, and this ignorance is represented by 1 - p(A), i.e., everything in the unit interval not assigned to p(A) is assigned to ignorance. When explicit evidence against A exists, then ignorance is computed as 1 -p(A) - p(not A), or everything other than p(A) and p(not A) in the unit interval. It is understood that p(A) + p(not A) = ≤ 1, and ignorance is greater than or equal to zero. The evidence interval can degenerate into a point, and that point is the classical probability.

Finally, fuzzy sets can also be represented by intervals using the well-known alpha-cut decomposition. An alpha-cut is the real interval of the fuzzy set which corresponds to a constant membership whose value is alpha . Each interval is associated with a membership value. The vagueness or uncertainty as to whether an object belongs to a class or set is a question of membership. In classical set theory, an element is either within the domain of the set, or it is not. Mathematically, this binary notion of set membership is handled with the indicator function. In fuzzy set theory, the degree of membership of an element x in a set A can be any value in the interval [0,1].

PSDesign will use a general approximate reasoning module which can be superimposed on the object-oriented data structure. The module will be flexible, in that it will accommodate several kinds of uncertainty representation such as probabilities, certainty factors, fuzzy set theory, and evidence theory. More important, the way the uncertainty is represented can be customized for the particular parameter or model. In general, the uncertainty is carried within the object-structure as an attribute just like other attributes. The uncertainty attribute is manipulated using any of the interval methods. Combination of different kinds of uncertainty in the same design problem and propagation of these uncertainties through the design model as part of the inference process are equally flexible, with allowances for the different kinds of uncertainty.

Demonstration

A demonstration of ***PSDesign*** is provided in the format of a graphical representation (fault-tree format), which portrays a number of "slides" which serve as the interface between a knowledge-based system and the user. These "slides" are interlaced with "PC-screen image" dialog sessions which interactively guide the user through the ***PSDesign*** software.

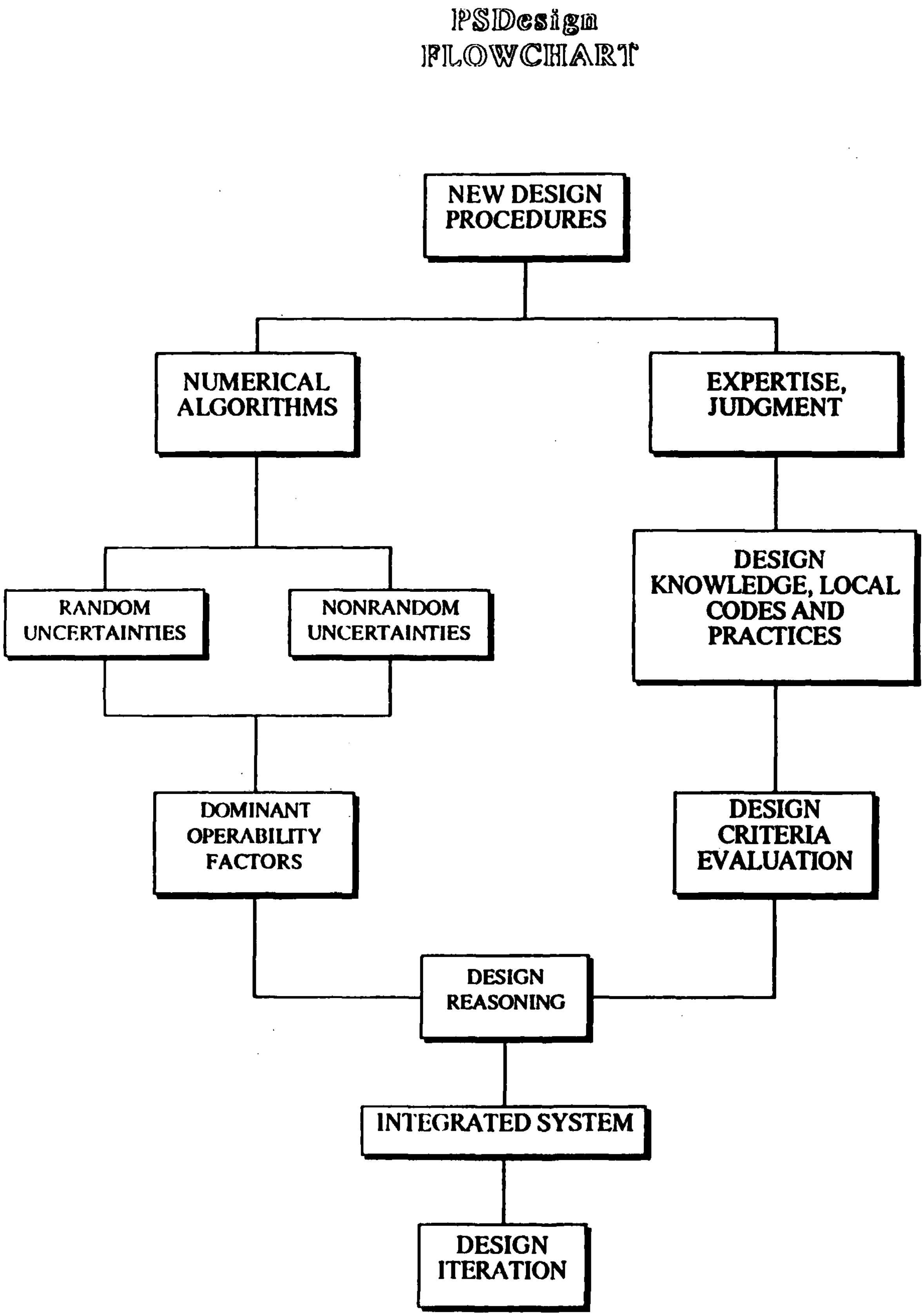


Figure 1. Project Scope

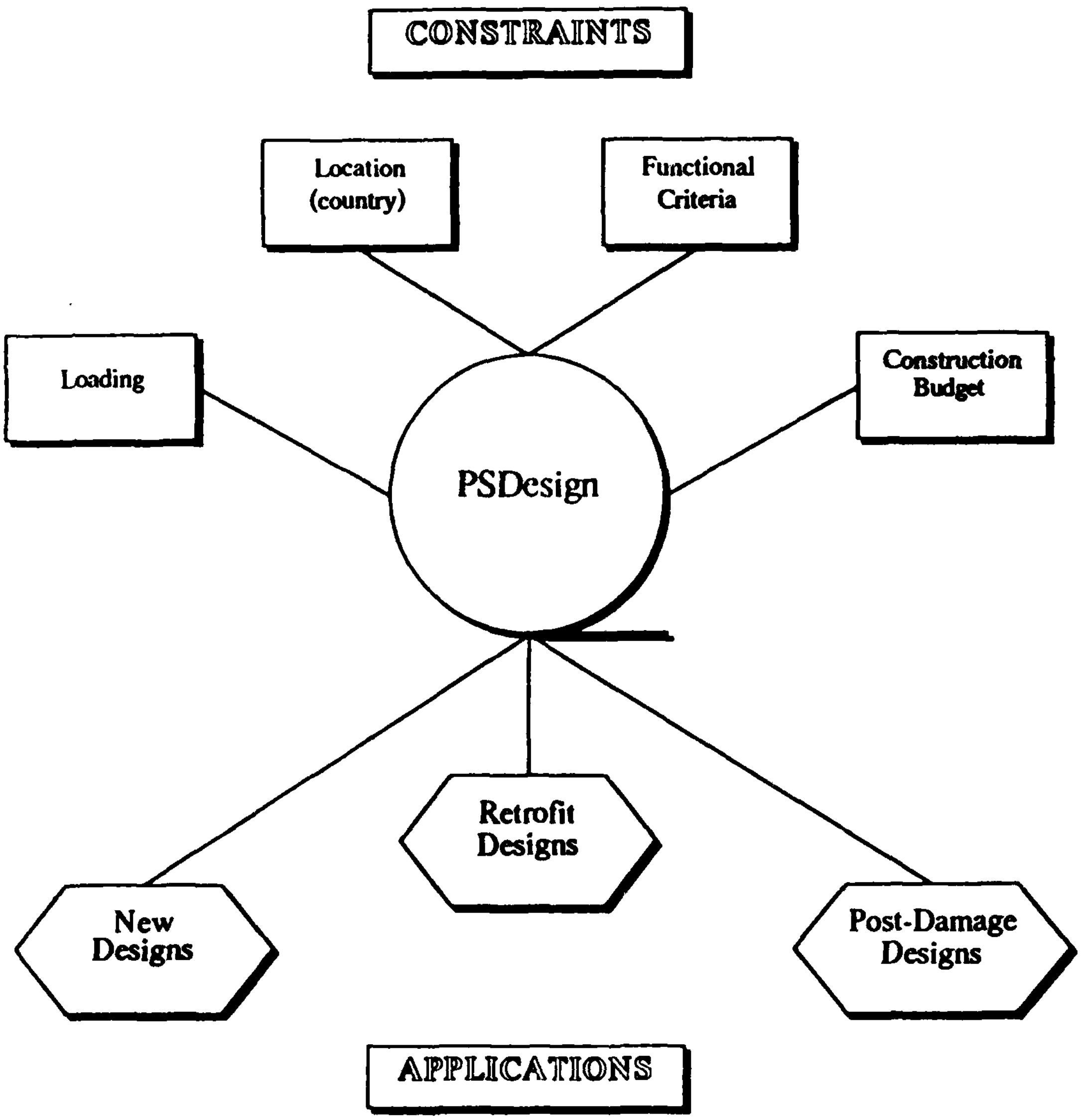


Figure 2. Constraints and Design Applications on the Design Problem

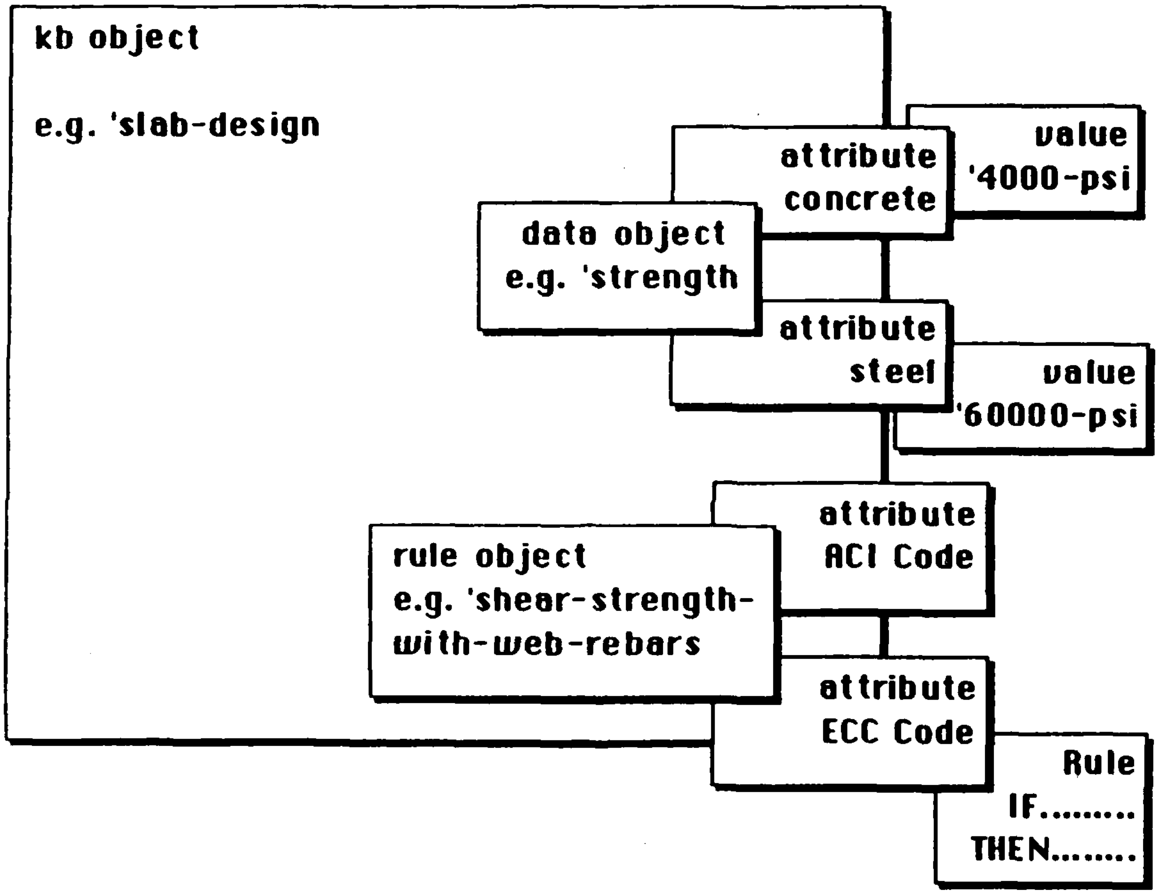


Figure 3. An object-oriented data/representation structure

KNOWLEDGE-BASED GRAPHIC DIALOGUES

Doris S. Shaw[1]

INTRODUCTION

The information that is stored in computer drawings has been the subject of educational research at USA-CERL. Most people think that the only use for the drawing data file is to reconstruct the image on the screen or plotter. We have found that careful analysis of the elements in the file can provide knowledge about the designer who created the file as well as report whether the drawing data will meet criteria imposed by standardization requirements or file transfer software programs. The techniques used to extract such information, actually conducting an intelligent graphic dialogue with drawing file, has practical value to educators and construction managers alike.

NATURE OF THE PROBLEM

Computer Aided Design (CAD) software allows a designer to use a graphic language to communicate with a computer. In most cases the computer is represented to the user as a screen to draw upon rather like a drawing board. The designer uses an input device as he would use a pencil on paper. The success of the software for many designers is measured by its transparency. The less they are conscious of the machine, the better. However, each CAD drawing has an underlying data base which contains facts and rules that may be used for decision-making analysis and information extraction as well as to reconstruct the drawing on the computer screen or paper. This data file may be used to abstract knowledge about the drawing and its designer. In the study reported here, the problem was to gain information from a drawing that would diagnose the needs of a student attempting to learn the CAD system, and ultimately to evaluate his state of knowledge of the system.

OBJECTIVES

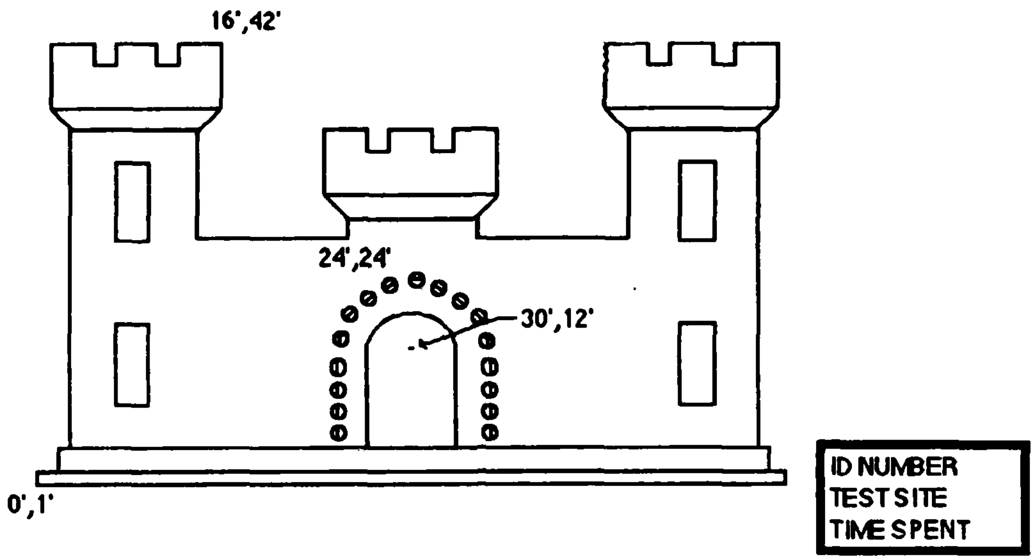
The objectives were to determine the status of a student from a dialogue with his drawing file. This took place both in a tutorial in which

[1]Principal Investigator, US Army Construction Engineering Research Laboratory, Champaign, Illinois.

he interacted with the computer by drawing and in a final draft which determined his level of skill. Two expert models were necessary for the operation of the system. One was the expert teacher who was involved in diagnosis of misconceptions and ultimately in prescription to remedy errors. The methods used to develop this model involved task specification in order to predict student errors (Birenbaum & Shaw, 1985). The other was the model of the expert designer whose drawing file differed in recognizable ways from the file of the novice (Shaw, 1986). The final drawing was used in the example presented here, though the adaptive diagnostics used in the tutorial employed many of the same approaches.

METHODOLOGY

In the study, students were given a course of embedded computer-aided instruction in introductory AutoCAD (SHAW, 1987), after which they were tested by drawing a Corps of Engineers castle (see figure) according to specifications drawn up by a Corps architect (Hsiung, 1987). Requirements included simple and complex entities; color, layer, and linetype conformity, naming conventions, accurate locations,dimensions, intersections, angles, and general setup. AutoCAD permitted the drawing data file to be accessed through an enhanced subset of Common LISP which included entity type names and graphic information. A program written in AUTOLISP was able to examine the file for drawing intersections, circles and arcs, angles of lines, layering, linetype, color, block creation and insertion of blocks.



Final Castle

The operation of the program allowed interaction with the drawing to the extent of turning on layers to allow for element checking and for input when there is a need for human judgement or recording of information such as time. The various procedural and conceptual components were reported to the user both as the program ran and in a text file output form. The output file has been successfully read into a database program for analysis purposes.

STATUS OF THE WORK AND RESULTS

The program was used to evaluate seventy-nine final drawings in a computer-based instruction project recently completed at CERL (Shaw & Golish, 1988). Its results have suggested that the techniques employed should be expanded for use in diagnostics and remediation in computer-based instruction for CAD, as mentioned before. We were able to use the knowledge of the drawing base to locate sources of student problems.

Differences recognizable in the drawing do seem to indicate levels of expertise.

EXPECTED FUTURE RESEARCH

For educational purposes, we hope to make the program more widely applicable by building a user interface which would allow input of parameters to evaluate in any given drawing. It is also possible to increase the interactive graphic dialogue with the computer by introducing drawing responses to the user, such as marking error points or printing corrections needed.

The applicability outside the educational environment suggests additional research. If the program can pinpoint errors in a drawing file, it may also be able to correct the errors for the purpose of meeting standardization criteria. That would require that the standardization rules become part of the grading program. This function might cut down upon translation problems which are common when a drawing file in one CAD system is transferred to another CAD system. Such programs may also be able to use rules for design decisions or for providing suggestions to the designer.

Continued work is needed to model expert design practices as well as to find ways to access the data in CAD systems other than AutoCAD. At present, we are working with the Intergraph system to extract this kind of information for embedded computer-based instruction. Observations and case studies are continuing for the purpose of creating better models for designing and for instruction.

REFERENCES

Birenbaum, M. and Shaw, D., "Task Specification Chart: A Key to a Better Understanding of Test Results," *Journal of Educational Measurement*, 22 (1985), pp. 219-230.

Hsiung, Chiway, T3B Final Castle Drawing Test, Waltham, Mass., 1987.

Shaw, D., "Case studies in architectural CADD education," In Turner, J. (Ed.), *Architectural Education, Research, and Practice in the Next Decade. ACADIA Workshop '86 Proceedings* (University of Houston, Oct.1986).

Shaw, D., *Teaching Assistant for AutoCAD* (Electronic Courseware Systems, Inc., 1987).

Shaw, D. and Golish, M., *Intelligent Embedded Instruction for CAD Systems,* Technical Report (in publication) (U.S. Army Construction Engineering Research Laboratory, 1988).

MAD, An Expert System to Monitor and Diagnose Packaged Boiler Operation

By: C. F. Blazek(IGT), M. Metea(IGT), and G. Schanche (USACERL)

Introduction

An expert system is being developed for the United States Army Corps of Engineers, Construction Engineering Research Laboratory (USA-CERL) by the Institute of Gas Technology (IGT).[1] This expert system, MAD, monitors and diagnoses combustion equipment. Specifically, MAD monitors data from various sensors on a package fire-tube boiler and determines whether or not the boiler is operating efficiently. If it is determined that the boiler is running inefficiently, MAD interacts with boiler personnel to determine possible causes for the inefficiency. A production rule-based paradigm is employed for knowledge representation, running on a 80286-based micro-processor platform. A backward chaining inference mechanism is used to reach conclusions concerning the state of the boiler. Knowledge acquisition was accomplished through extensive, tape-recorded interviews with boiler experts both internal and external to the Institute of Gas Technology. Additionally, at various points during the development of MAD, boiler experts were allowed to inspect and critique the system, with their suggestions being incorporated into the design.

Many small boiler control and monitoring systems, while useful, do not address the causes of boiler inefficiency. Furthermore, these systems only provide limited diagnostic information for troubleshooting boilers and educating boiler room personnel in the areas of combustion technology theory, operation, and maintenance. Typically, experienced boiler personnel draw on their many years of experience to conclude why a particular boiler is not operating efficiently. Additionally, the experience they have spent years acquiring is often lost to an organization when the personnel retires or finds work elsewhere. MAD is the result of an effort to monitor and diagnose a boiler in real-time, but also to draw knowledge from the mind of these experts and incorporate that knowledge into a computer so that an organization can retain a knowledge base after key employees have departed.

A variety of products exist for real-time monitoring and control of combustion equipment. In the past, these systems were mainly devoted to larger combustion equipment, as the high cost of sensors and control electronics prohibited their use in smaller, package boiler environments. In recent years, however, costs for zirconium oxide O_2 sensors, differential pressure transducers, thermocouples, and other sensors have fallen sharply. This, in conjunction with the availability of low-cost, mass-produced microcomputers has lead to the feasibility of intelligent monitoring and diagnostic systems for smaller combustion equipment. The MAD system is evidence that a cost effective tool can be developed to monitor, diagnose, and troubleshoot smaller combustion equipment, while simultaneously educating boiler personnel in boiler operation and maintenance theory.

Conscious of the fact that most boiler personnel may have little experience with microcomputers, a graphics package was employed to create a more "user-friendly" human-computer interface with MAD. Real-time data can be displayed in the familiar form of digital or analog gauges. Currently, automated data acquisition via remote sensors has not been accomplished, and input data is entered through a keyboard interface. In order to achieve a very simple interface, "arrow keys" on the keyboard are used to increase or decrease input values for various parameters. This effectively reduces the number of different keys an operator must cope with to three: the "up" arrow key, the "down" arrow key, and the "Enter" key. Extensive use of a "help" key is also available, as described later.

A variety of combustion technology experts were employed during the knowledge acquisition process in order to minimize any biases inherent in an individual. In addition to reviewing associated literature, audio tape-recorded in-depth interviews were conducted with independent combustion consultants, representatives of leading boiler manufacturers, manufacturer's field personnel, in-house combustion researchers, and in-house experienced boiler operators.

MAD was developed using a Texas Instruments expert system development tool called "Personal Computer Plus" on a 80286-based micro-computer. The target system for MAD includes sensors at various strategic locations on the boiler feeding into a 80386-based microcomputer with a hard disk and graphics capability.

Methodology

MAD was developed using a production rule-based paradigm. Production rules were chosen to represent expert knowledge since this particular paradigm is efficient in minimizing development time while allowing additional knowledge to be easily added to the system. Production rule-based systems (alternatively called production systems, or rule-based systems) can be viewed in a very general sense as a series of "IF-THEN" statements. The IF part is often referred to as the antecedent or condition, and the THEN part is the consequent or action. As an example, the statement IF A THEN B is interpreted as 'If condition A is true, then perform action B'. A whole series of these statements (henceforth referred to as 'rules') form the knowledge base. The knowledge base can be fed initial facts (the flue gas temperature, oxygen, etc.) and arrives at a goal, or conclusion (the boiler is inefficient) by applying its various rules.

Five basic areas of boiler operation and evaluation are covered by MAD. These are listed as "menu" options for the user of MAD to select upon beginning a session, and include:

1) Evaluate efficiency
2) Cause of inefficiency
3) Troubleshoot
4) Water status
5) Education

"Evaluate efficiency" is the area covered most completely by MAD. It requires keyboard input by the user of various parameters that eventually will be input on a real-time basis from boiler sensors. These parameters include, but are not limited to, the temperature of the flue gas, the amount of oxygen and combustibles exiting the stack, the steam pressure, and other input data. Once the input data has been gathered, MAD concludes whether the boiler is running efficiently or not. If it is not, it will display the general reason which leads to the conclusion. In reaching the conclusion, MAD calculates the optimal value of a parameter (for a given boiler under operating conditions specified), factoring in a certain tolerance before concluding that the boiler is in a state of inefficiency. For example, it might state that the input value of 4.0 percent excess oxygen in the stack is more than 1 percent greater than the optimal amount of 2.1 percent for that particular boiler. In the final stages of development, MAD will automatically input the data from the sensors, and if the boiler is running inefficiently, interact with the operator to determine the malfunctioning boiler component.

"Cause of inefficiency" determines which malfunctioning boiler component might be causing the boiler to operate inefficiently. Currently, the user must select this area from the first menu presented, although the finished version of MAD will enter this area automatically upon discovering inefficiency in the boiler. Operator interaction with MAD is essential in this area due to the prohibitive cost of installing feedback sensors to check the integrity of every valve, electrical wiring assembly, moving part, and noise associated with the boiler.

One criteria crucial to an accurate evaluation of any boiler is that the method used to gather the input data is performed in a rigorous manner. Accordingly, there is a series of questions during the consultation which interrogate the user as to the procedures employed in gathering the data. The operator's responses to these questions are then used to determine the degree of confidence that the system (and hence, the user) has in it's opinion of what is causing the boiler's inefficiency. If the data acquisition method is sloppy or unknown, MAD prints a qualifying statement that any opinion rendered must be deemed questionable due to the fact that its input is unreliable. Otherwise, MAD prints a message that the opinion rendered is only trustworthy to the percent indicated at the end of that message. The percent of certainty listed is also a function of the rigor or certainty of the data acquisition process as stated by the user.

The "Cause of inefficiency" area does not currently cover all possible conditions that might exist in an operating boiler. The most important parameters in easily determining boiler efficiency are the temperature, oxygen, and combustibles found in the flue gas. At a given point in time, each of these parameters might be either at an excessively high level, a normal or acceptable level, or at a abnormally low level. Recognizing that each of these three parameters has three possible conditions yields a matrix of 27 possible conditions. MAD will accommodate conditions where two of the three input parameters are at normal levels and the third parameter is excessively high. Still to be covered are conditions where more than just one parameter is high, and also conditions where one or more of the parameters are low. An exception to this is the condition where oxygen and temperature are high while combustibles are normal; MAD will handle this condition.

"Troubleshoot" is under development. Extensive knowledge acquisition (interviews with boiler experts) has been accomplished, but further development effort is required before significant progress can me made in this area, as it will most likely prove to be the most complex of the five areas. Selection of this item from the initial menu merely produces a message stating that further development of the area is needed. Eventually, "Troubleshoot" will diagnose why a boiler fails during operation or upon start-up. It is anticipated that future versions of this section will include extensive use of graphics such as the isometric drawing presented in Figure 1. Through the use of such graphics, the expert system will not only indicate what the problem might be, it will also show the operator where to look for the problem and check for proper operation.

"Water status" is a simple diagnostic system for the water quality of the boiler. Currently, it asks the operator about the physical condition of pipes and valves in the water system, as well as helping to evaluate various alkalinity readings of the water. It performs basic diagnostics, but needs to be developed to a much greater extent. Once automated water quality sensors and treatment instruments become cost effective, MAD can be easily interfaced with these devices to provide "hands-off" water conditioning.

"Education" is a totally different area from any of the above. It is envisioned as a supplement to educate novice boiler personnel as well as those with considerable years of experience, as it seems that no one person, regardless of the number of years spent in boiler rooms, will know all there is to know about boilers. If such experts do exist, they definitely will be the exception. Furthermore, it was felt that many young and inexperienced boiler personnel would find great benefit in an educational tool possessing negligible operational cost, as well as being available at all times in the boiler room itself.

The "Education" area can serve as a colorfully illustrated, on-line textbook on the principles of combustion technology, boiler operation, and maintenance. The user may study an area of interest by choosing one of various options offered on multiple "menus". The system could alternatively quiz the user by setting up various scenarios, requesting an answer from the user, and then giving and explaining the correct answer. These sessions would include diagrams such as that presented in Figure 2. which can graphically illustrate the proper procedures for installation or maintenance of boiler components.

At present, the "Education" area is devoid of knowledge and when selected, will only state that it is under development. However, a rudimentary educational system has been incorporated into the current version of MAD to demonstrate the degree and level of knowledge that would be transferred to the user. Many of the prompts (questions) displayed to the user include a note stating that a further explanation of the prompt is available by pressing a help key. This help key has two purposes; to instruct the user on how to enter input for the prompt displayed, and also as an educational tool as mentioned above. This auxiliary educational feature will be left in after the full "Educational" area has been developed so that the user is exposed to learning through various avenues of the system.

Status of Project

The first phase of an expert system has been developed which will monitor and diagnose package fire-tube boilers. Eventually, it is envisioned that this system will receive real-time input from sensors located on the boiler. This data would then be analyzed to reach a conclusion concerning the boiler's efficiency. Should the boiler be operating inefficiently, MAD interacts with an operator to determine the possible causes. In addition to performing monitoring and diagnostic functions, MAD will also have the ability to troubleshoot inoperative boilers, analyze water quality and recommend procedures to rectify water quality deficiencies, and finally to educate boiler personnel on boiler operation and maintenance fundamentals.

MAD presently does not have the capability to interface with the various feedback sensors associated with the boiler. Boiler output data must be entered into MAD by boiler personnel via a keyboard interface. However, real-time data acquisition will eventually be accomplished through the use of a Texas Instruments software product designed for that purpose (PC Online), and by upgrading the hardware to a 80386-based microcomputer. Due to the relatively slow response times involved with typical boiler operations, this platform should be adequate for real-time expert system monitoring.

As mentioned above, knowledge acquisition and system development is still required in all of MAD's modules. The system presently functions in a rudimentary manner. In order to offer a more comprehensive system, additional development is required, especially in the areas of "Cause of Inefficiency", "Troubleshooting", and "Water Quality." The "Education" module is presently void of knowledge except for that contained in the program's help files. However, this module is intended solely as a teaching aid. After completion of MAD's development, actual field testing will still be required.

References

1. Metea, M., Blazek, C. F., "Expert System for Diagnosing Heat Plant Equipment Failure", Draft Final Report by the Institute of Gas Technology under Contract DACA88-86-D-0016 Task C035, April, 1988.

For additional information please contact:

Mr. Christopher Blazek
CERL Task Order Project Manager
Institute of Gas Technology
Energy Development Center
4201 W. 36th Street
Chicago, Illinois 60632
Phone No. 312-890-6466

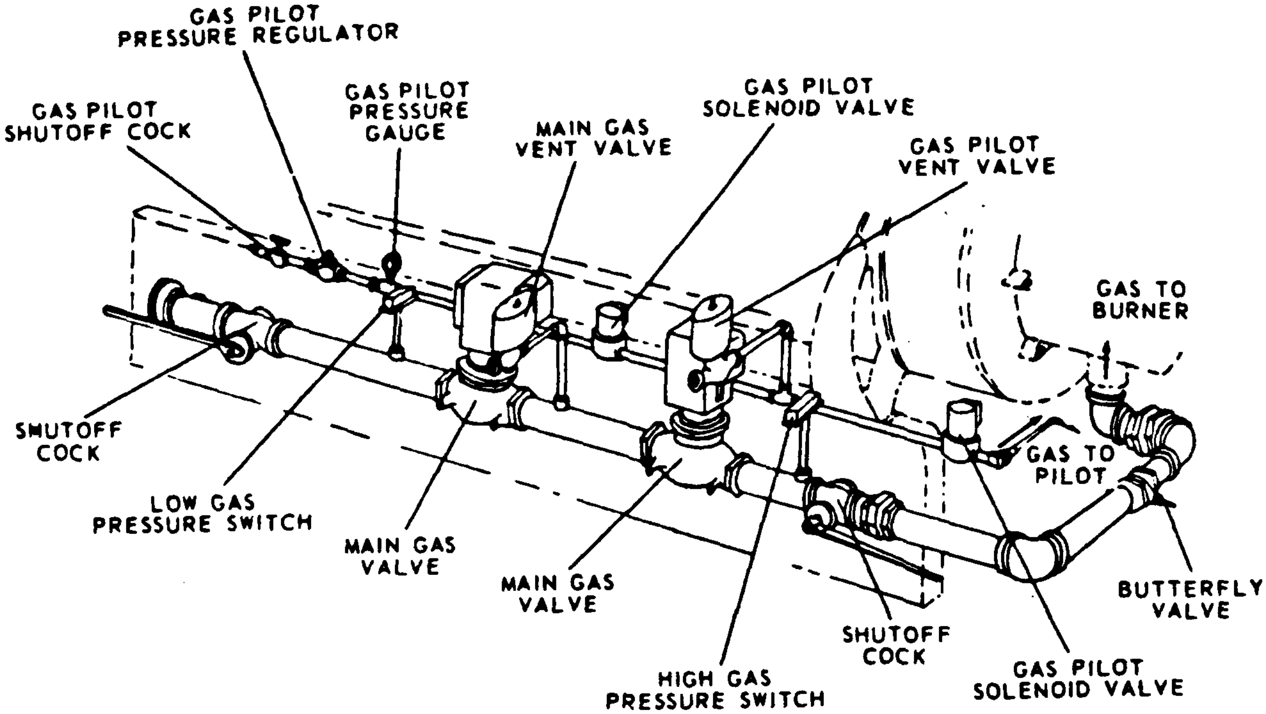


Figure 1. Isometric Drawing of Typical Gas Fuel Train.

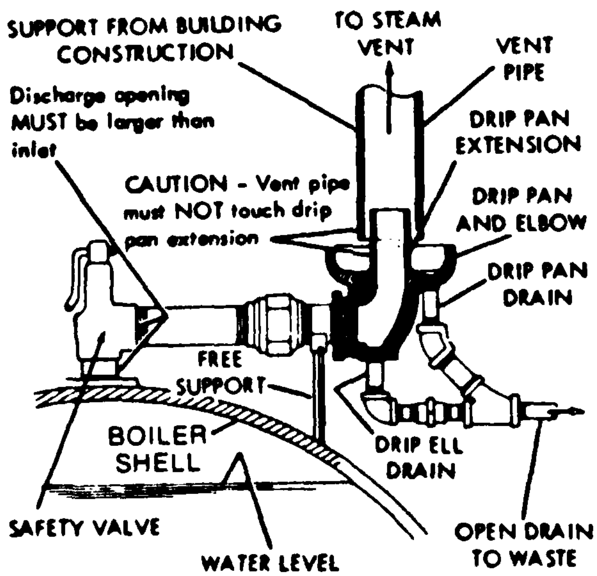


Figure 2. Drawing of Recommended Safety Valve Installation Proceedure.

KNOWLEDGE WORKER SYSTEM

Beverly T. Coskunoglu and Loretta Y. Hsui[1]

NATURE OF PROBLEM:

The Office of the Assistant Chief of Engineers (OACE) is the Army Staff element responsible for formulation of the Military Construction Program which, eventually, is submitted to the Congress for funding. OACE action officers define requirements, allocate resources, review execution, gather and disseminate project and program information essential to sound facilities-related decisions. For the purpose of this project, the OACE action officers who perform functions such as these are referred to as knowledge workers.

The activities of an individual knowledge worker who plays a role in this complex process are driven by continuously shifting events and dates. Timely readjustment and response to the changing demands are critical, even if the knowledge worker is new on the job and does not completely understand either the process or the task.

In the Army this is a particularly vexing problem because turnover is very high. Other aspects of the Army organizational climate which impact continuity of operations are organizational changes, the work environment, the technology environment and the workload. The work environment encourages last-in-first-out reaction and short-range time scheduling. Survival in this environment does not favor thoughtful, systematic review of information.

---

[1] Co-Principal Investigators, US Army Construction Engineering Research Laboratory, FS Division, Champaign, Illinois.

OBJECTIVES:

The Knowledge Worker System (KWS) is being proposed to assist action officers in fulfilling their responsibilities. KWS has the following goals:

o Capture the institutional knowledge which is often lost when action officers change positions.

o Perform dynamic scheduling and rescheduling of organizational activities/processes/events.

o Assist action officers in both keeping track of and in performing their daily responsibilities. Explain to employees what must be done, when, how and why.

o Automatically generate those items amenable to automation and free employees from repetitive tasks.

o Assist in workload leveling.

o Be applicable to a wide range of organizational settings.

FUNCTIONAL REQUIREMENTS:

To accomplish these goals, the following five functional modules are required:

1) KNOWLEDGE CAPTURE AND ACCESS: The subfunctions accomplished by this module are knowledge acquisition; maintenance; and retrieval.

2) DYNAMIC SCHEDULING: This module will include Activity Network, Activity Scheduling and Tracking, Impact Analysis, Resource Leveling, Daily "To Do List", Knowledge Worker's Calendar, "Look Ahead" and "Look Back" capabilities, Status Report, and Management Report.

3) AUTOMATIC EXECUTION: This module will involve automation of activities such as initiating batch programs, checking and sorting electronic mail, downloading/uploading data, generating status and management reports, and communications between personal computers and minis/mainframes.

4) USER INTERFACE MANAGEMENT: The User Interface will be designed with characteristics such as consistent appearance, customizable configuration, timely responses, context-sensitive help, error prevention and correction, windows and automatic reminders, etc.

5) SYSTEM INTERFACE MANAGEMENT: The System Interface Management program will furnish the overall orchestration between disparate software and hardware components. This module will provide the functions of coordinating, routing, instructing, and monitoring.

METHODOLOGY:

The first step is the analysis of the data which will comprise the system's knowledge base. Standard operating procedure (SOP) handbooks have been developed for three knowledge workers who will eventually be served by KWS. After classifying the types of data to be stored, a knowledge representation scheme will be developed and tested.

After a conceptual model is developed, researchers will focus on producing a knowledge acquisition package. The purpose of this package is to expedite and standardize the process of knowledge capture. Other potential benefits include early user involvement, decreased knowledge acquisition costs, and stand-alone software that will prototype some KWS functions.

Concurrently, system requirement analyses will be conducted. Software and hardware requirements will be defined as part of this process. The use of Computer-Aided Software Engineering (CASE) technology will be investigated as a potential development and production tool. Potential applications of CASE for this project include: system planning and design, activity modeling, data structuring, rapid prototyping, and code generation.

Well-defined functional requirements will be developed. The conceptual model will be further evaluated and refined, as part of the design stage. Both the functional requirements and the knowledge representation model will provide the framework for design and implementation of the system. Design details will be checked, attempted, and iteratively refined as part of the development and testing stage.

Preliminary investigation indicates that the system will be developed in hybrid artificial intelligence environment consisting of a relational database management system, expert system, project management system, and mainframe-micro communication linkages. The system will be fielded as it is developed with the same users for whom the SOP handbooks and knowledge acquisition packages were developed. After fielding and evaluation is completed, the production version of the system will be put into use.

STATUS OF WORK:

Analysis of data that will comprise the system's knowledge base is underway; development of a knowledge representation model will follow.

PRELIMINARY RESULTS:

(1) Functional Description:

A draft version of the KWS functional analysis has been produced.

(2) Standard Operating Procedure Handbook:

The handbook describes how to perform each type of activity for a specific position.

(3) Data base and retrieval software:

The rudimentary software developed to date presents the data about when each activity must be done and by whom, and is indexed to activities in the handbook.

EXPECTED FUTURE RESEARCH:

(1) Knowledge representation

(2) Knowledge acquisition tools

(3) Dynamic scheduling

(4) System integration

(5) CASE technology

(6) Human factors engineering

(7) Mainframe-micro links

POC:

Beverly Coskunoglu, PI, FS Division Ext. 728

Loretta Hsui, PI, FS Division Ext. 387

US Army Construction Engineering Research Laboratory

P.O. Box 4005, Champaign, Illinois 61820-1305

REFERENCES:

1. Drucker, Peter F. "The Coming of the New Organization", Harvard Business Review, January-February, 1988.

2. Hart, Anna. Knowledge Acquisition. McGraw-Hill, New York, 1986.

3. Hayes-Roth, Frederick, Waterman, Donald A., and Lenat, Douglas B., (eds). Building Expert Systems. Addison-Wesley, Reading, MA, 1983.

4. Hillier, Frederick S., Lieberman, Gerald J. Introduction to Operations Research. Holden-Day, San Francisco, 1980.

5. Harmon, Paul, and King, David. Expert Systems. Wiley Press, New York, 1985.

A Knowledge Engineering Approach to the Analysis and Evaluation of Schedules

Jesus M. De La Garza[1], E. William East[2], and C. William Ibbs[3]

Objective:

This paper describes a knowledge engineering study which synthesizes the operational knowledge needed to perform a project management expert task. The particular task in question is criticizing and comparing construction schedules for vertical construction [De La Garza 88]. The methodology utilized in the study is aimed first and foremost at extracting, formalizing, and articulating empirical judgmental knowledge about construction schedule analysis. The secondary purpose of this research is to explore the issues that arise in the development of a Knowledge-Based System (KBS).

This research advances the state-of-the-art by: 1) adding to and formalizing an amorphous mass of construction scheduling knowledge; 2) recognizing and making construction schedule criticism a formal project management task; 3) proposing a knowledge engineering methodology capable of transforming ill-defined construction scheduling knowledge into the specifics of an operational KBS; and 4) providing an object-oriented infrastructure of knowledge representation schemes.

Motivation:

Although Project Management Systems (PMSs) have become more substantial and complex, they are still widely acknowledged as incomplete project control aids. Specifically, their inability to collect data and interpret qualitative and subjective project information requires construction managers to interpret such information largely unassisted. This lack of computer-aided interpretation leads to variations of construction managers' and firms' performance. This is especially true regarding schedule analysis and control in which experience and the resulting subjective judgment play a major role.

In addition to these problems, construction managers are dissuaded from using the Critical Path Method (CPM) and the Program Evaluation and Review Technique (PERT) methods effectively by the need to perform time-consuming network searches and laborious hand-calculations, which are typical of "what if" analyses.

---

1 Ph.D. Research Assistant, Dept. of Civil Engineering, University of Illinois at Urbana-Champaign, and Associate Investigator, US Army Construction Engineering Research Laboratory, Champaign, Illinois.

2 Principal Investigator, US Army Construction Engineering Research Laboratory, Champaign, Illinois.

3 Associate Professor, Dept. of Civil Engineering, University of California at Berkeley, Berkeley, California.

Technological Opportunities:

Artificial Intelligence (AI) technology provides a way to capture and replicate empirical judgmental knowledge in the form of computer programs called KBSs. During a National Science Foundation conference held in May 1985 at the University of Illinois, several researchers pointed out that the nature of the construction industry makes these KBSs especially applicable to construction engineering and management [Ibbs 85, Ibbs 86].

Researchers and practitioners agree that in this industry there is a high dependence on empirical rules and procedures derived from experience, rather than from a scientific knowledge source. Concurrent to the conference, researchers were beginning to isolate construction scheduling and control as a prominent example for KBS application [Avots 85, Levitt 85, McGartland 85, Nay 85].

A subsequent workshop [O'Connor 87] confirmed and endorsed the fact that construction schedule criticism and comparison represent a crucial element of effective project management. Participants of this symposium focused their discussion on the use of KBSs in construction scheduling. KBSs were identified as an excellent technology to: 1) capture compiled and idiosyncratic knowledge about construction schedule criticism, comparison, and generation; and 2) develop graphical user interfaces for operational systems.

Two other workshops [Kitzmiller 87, Wilson 87] outlined the criteria for and purpose of needed intelligent interfaces for current algorithmic software. [Wilson 87 p.16] summarizes the consensus of the participants at his workshop, with respect to intelligent software interfaces, as follows:

> "Many current software packages are large algorithms, or collections of linked algorithms, that are difficult to use or whose results demand serious, informed interpretation...Knowledge-Based Systems (KBSs) offer the potential of providing intelligent pre- and post-processors of such packages...As post-processors, the KBSs can be used to interpret program results as to their accuracy, validity and applicability. A major research issue within this approach is the assessment of how much of the knowledge required for such application is compiled and generic, and how much is empirical and idiosyncratic...Generic tools for such applications should be able to represent the (widely-shared) compiled knowledge, while at the same time allowing the user to augment the knowledge base with his/her own experiential knowledge."

Definition of Construction Schedule Analysis:

The construction schedule analysis problem domain is characterized by the use of expert knowledge, judgment and experience. Management of dynamic schedules requires knowing not only what has changed, but also requires reacting to those changes before their impact occurs. Comparison and analysis of schedules is typically performed at three different stages: (1) at the time of the initial version; (2) across different schedule versions during project execution; and (3) within the last schedule version, reflecting actual and baseline expectations. This research project focuses only on the construction expertise required to accomplish the first and second phases. In these stages, the issues of maintaining an on-going historic database play a less important role than in the third stage.

Initial schedule analysis is defined as the verifications owners and/or contractors perform on the initial construction planning schedule. Project managers, whose task at this stage is to assess whether the proposed schedule is reasonable, need answers to such questions as: Does the schedule meet the contract requirements?, Is the critical path reasonable?, Are owner-controlled activities included?, Have major subcontractors participated in the formulation of the plan?, What is the overall degree of schedule criticality?, Do procurement activities precede special installation tasks?, Does the cost estimate comply with the contract documents?.

In-Progress schedule analysis is defined as the type of evaluations owners and/or contractors perform across different schedule versions during project execution. Project managers face these and other issues such as: Are we on schedule?, How much should we pay?, What is different about these schedules?, Are winter-sensitive activities being scheduled during winter?, Is the progress payment request reasonable?, Should the duration of future activities be modified based on past experience?, How can I tell if activities are in trouble?

Both Initial and In-Progress schedule analyses are considered with four major categories of issues: 1) General Requirements; 2) Logic; 3) Cost; and 4) Time.

Scope of the Knowledge Base:

The scope of the knowledge base has been restricted to one specific class of buildings in order to maintain a narrow focus and to concentrate the study on the scheduling process itself. The class of buildings the knowledge base addresses are mid-rise reinforced concrete or steel structures. The advantages of using this type of construction are: 1) it represents a large segment of the building construction market; 2) it is characterized by repetitive operations; and 3) solutions to early-diagnosed problems may be propagated to the rest of the structure.

There is an upper limit on the number of stories, roughly 19. The mid-rise structure was chosen mainly because the construction of high-rise buildings, 20 stories and above, requires another set of elevators, different construction equipment, and different construction methods. The scope of the knowledge base has also been limited to commercial buildings, since the construction methods for these types of buildings are similar. In contrast to this type of construction are projects such as hospitals, factories, warehouses, homes, schools, dams, bridges, highways, and so forth.

Knowledge Metamorphosis:

The conversion of words of advice into executable procedures is an important task. The knowledge transformation is briefly presented here to provide an overview of the sequenced process and to clarify the transition from one stage to another. Figure 1 depicts the evolution of the knowledge base. The following subsections explain each of the products generated by the knowledge metamorphosis process.

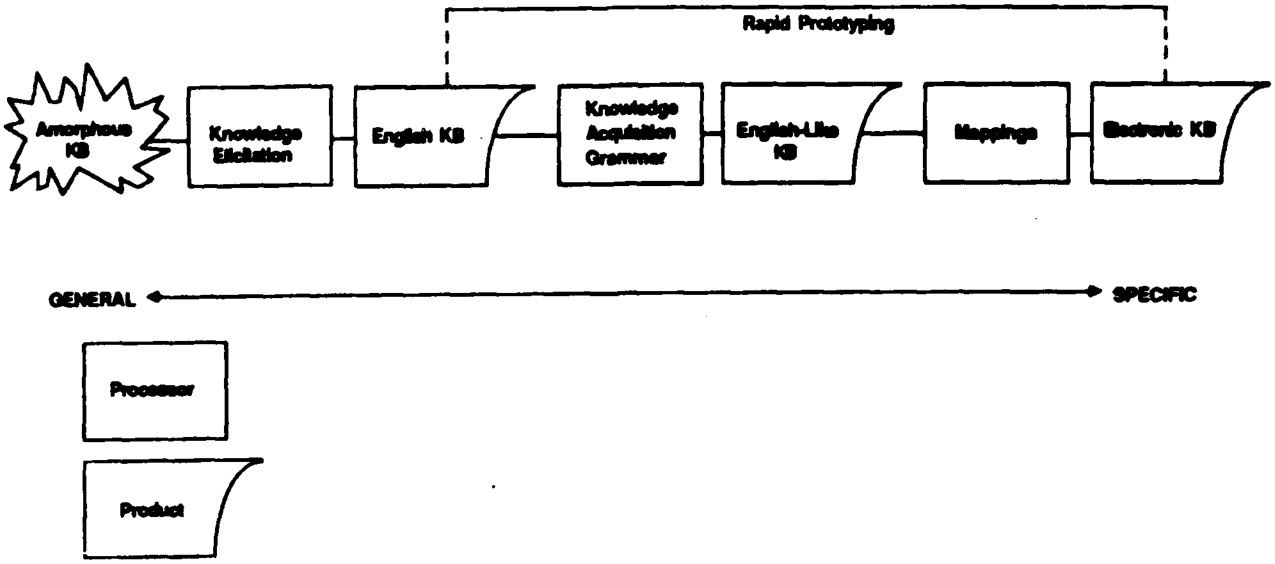


Figure 1. Knowledge Metamorphosis

<u>Amorphous Knowledge Base.</u> Despite the economic benefits of performing schedule analysis, many project managers do not do so because: 1) it requires project managers to have a high level of technical expertise, which can only be acquired with years of experience; 2) the time-consuming nature of most procedures dissuade project managers from analyzing the schedules effectively; and 3) reports of on-going work are often one or two weeks behind real time, thus, construction managers do not take management information systems seriously.

Expertise-related problems occur because senior project managers cannot transmit all their knowledge to junior project managers. Some of the reasons are: 1) Senior project managers know more than they can explain about construction schedule analysis. They can best discuss their knowledge in the context of concrete problems; 2) Most expert knowledge is heuristic and experiential; large parts are not grounded on first principles; 3) Senior project managers learn to recognize thousands of associations between data and solutions to the extent that they are unaware of the process. They learn to recognize solutions as opposed to constructing them; 4) The major difference between a junior and a senior project manager is the quantity, quality, retrievability, and organization of his/her knowledge; and 5) Much of the knowledge to which senior project managers have access is idiosyncratic to the company they represent and to the type of projects with which they are most familiar.

Despite the proliferation of computerized PMSs, there has been little effort to quantify and structure the knowledge required to interpret the PMS information. Thus, the title amorphous knowledge base is appropriate.

<u>English Knowledge Base</u>. The English knowledge base consists of a set of conceptual scheduling provisions and procedures written in plain English. These provisions and procedures are the product of investigating and applying three knowledge elicitation techniques to the amorphous mass of construction scheduling expertise. Figure 2 shows the techniques used in this research and their output. The overlapping depicted in the provisions and methods boxes symbolically implies some duplication in their contents.

The union of all provisions defines the breadth of the knowledge base. An important attribute of these conceptual provisions is that they represent the "what" and do not explicitly indicate "how" they should be interpreted or implemented. For example, a provision in the "time" category basically states: "Float should be broad enough to support the premise that it has not been manipulated". In this case, the intent of the provision is not intertwined with any float sequestering technique.

The union of the methods generated by each knowledge elicitation technique delineates the depth of the knowledge base. These procedures symbolize "how" a conceptual provision may be interpreted. The conceptual provision stated above may be interpreted by relating it to preferential scheduling techniques, such as preferential logic ties, lead/lag activities, or extended activity durations.

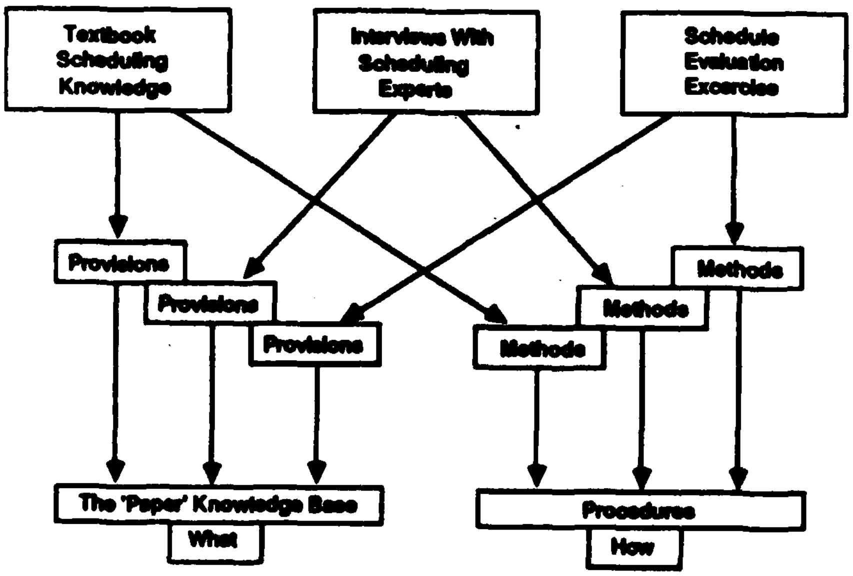


Figure 2. Knowledge Elicitation Techniques

English-like Knowledge Base. Since much of the construction scheduling knowledge is procedural, the knowledge engineering approach at this stage has as its main objective the transformation of conceptual provisions into executable procedures.

Some knowledge a project manager exercises is generic in nature while other knowledge is idiosyncratic to the company with which he/she is affiliated. Thus, there can be several ways to implement a conceptual scheduling provision. The purpose of this phase is to identify at least one type of implementation and allow flexibility for further modifications or additions.

Once a procedure is attached to a provision, its representation is no longer in plain English. Rather, it is represented using a notation that is less idiomatic and more procedural. However, with this format, a person who is not familiar with the specifics of the computer language in which the KBS is written can still read and record additional knowledge.

Electronic Knowledge Base. The computer implementation of conceptual provisions requires the selection of programming languages. Commercially available AI tools that are applicable to the derivation end of the problem solving spectrum have been considered.

Two different programming environments have been utilized for developing these ideas to the proof-of-concept level; that is, having enough functionality to demonstrate concepts but not really developed yet to the operational level: 1) Texas Instrument's Personal Consultant Plus; 2) Inference Corporation's ART. A third tool, e.g., Gold Hill's GoldWorks, has recently been selected to develop an operational KBS.

Each of these, or any other computing platform, requires a set of mappings whose function is to transform the knowledge written in the English-like notation into the specifics of the chosen programming language syntax. Once in this format, modifications to the knowledge base require the expertise of a programmer.

Knowledge Implementation:

The Personal Consultant Plus implementation represented the first attempt to codify the scheduling concepts. The main thrust of it was to find out whether it was possible to embody construction scheduling expertise in a computer language. For this reason, the code did not have to be elegant or efficient. In addition, little consideration was given to execution speed. Since the USA-CERL specifications called for a microcomputer-based tool, the following hardware-software environment was selected:

Software:

Personal Consultant Plus; Inference Engine; Versions 1.0, 2.0
Primavera; Project Management System; Versions 2.5, 3.0
dBASE III; Relational Database Management System; Version 1.1

Hardware:

IBM PC/XT, AT, compatibles, and TI Professional

The first system design problem faced was how to make the PMS information available to the inference engine, other than by re-typing it. The solution included the splicing of a relational database management system (RDMS) between the PMS and the inference engine. This RDMS would act as a repository of schedule information, in addition to providing its own command language to perform computations on schedule information. It was evident that by solving this obstacle in this manner, that it was also conceptually possible to access all kinds of external databases, e.g., a database consisting of estimating data.

Since the PC Plus versions utilized during this implementation lacked database-like relational capabilities and a pattern-matching language, all such work was done by dBASE III. This system's design decision suggests that for every value to be retrieved from or calculation to be performed on the schedule database, there has to be a dBASE III command file capable of executing the PC Plus request. In this regard, the function of the PC Plus rules was mainly to decide which command file to select and execute. By the time the prototype was finished, it became evident that it could have been implemented in dBASE III alone without using PC Plus as the anchor.

The system was developed to the proof of concept level, that is, having enough functionality to demonstrate concepts but not really usable. The PC-based tools imposed severe constraints on the system's development. They were, among others, 1) insufficient computational power; 2) limited potential for system growth; 3) lack of a truly frame-based representation language; 4) lack of a pattern-matching language; and 5) lack of object-oriented programming capabilities. These limitations led to the selection of a computing platform able to raise some of these constraints, eliminate some others, and by the same token, impose additional ones, e.g., the development environment is too expensive to be considered a viable delivery platform.

In the ART implementation, rather than re-coding the scheduling concepts previously implemented on the personal computer, it was decided to develop a conceptual semantic network infrastructure. This would permit and stimulate system growth, as well as serve as the foundation for rule creation.

Current efforts are being devoted to the development of an operational KBS. Such system is being developed in GoldWorks.

References:

[Avots 85] Avots, I., "Application of Expert Systems Concepts to Schedule Control," Project Management Journal, Vol. 16, No. 1, March, 1985, pp. 51-55.

[De La Garza 88] De La Garza, J. M., "A Knowledge Engineering Approach to the Analysis and Evaluation of Schedules for Vertical Construction," Ph.D. Thesis, Dept. of Civil Engineering, University of Illinois at Urbana-Champaign, May, 1988.

[Ibbs 85] Ibbs, C. W., Proceedings of a Workshop for the Development of New Research Directions in Computerized Applications to Construction Engineering and Management Studies, University of Illinois at Urbana-Champaign, Construction Research Series, Tech. Report No. 19, 1985.

[Ibbs 86] Ibbs, C. W., "Future Directions for Computerized Construction Research," Journal of Construction Engineering and Management, ASCE, Vol. 112, No. 3, September, 1986, pp. 326-345.

[Kitzmiller 87] Kitzmiller, C. T., and Kowalik, J. S., "Coupling Symbolic and Numeric Computing in Knowledge-Based Systems," The Artificial Intelligence Magazine, Vol. 8, No. 2, 1987, pp. 85-90.

[Levitt 85] Levitt, R. E., and Kunz, J. C., "Using Knowledge of Construction and Project Management for Automated Schedule Updating," Project Management Quarterly, Vol. 16, No. 5, December, 1985, pp. 57-76.

[McGartland 85] McGartland, M. R., and Hendrickson, C. T., "Expert Systems for Construction Project Monitoring," Journal of Construction Engineering and Management, ASCE, Vol. 111, No. 3, September, 1985, pp. 293-307.

[Nay 85] Nay, L. B., and Logcher, R. D., "An Expert System Framework for Analyzing Construction Project Risks," Tech. Report CCRE 85-2, Dept. of Civil Engineering, MIT, 1985.

[O'Connor 87] O'Connor, M. J., and De La Garza, J. M., Editors, Proceedings of a Workshop on Expert Systems for Construction Scheduling, U.S. Army Construction Engineering Research Laboratory, P-87/13, Champaign, Illinois, August, 1987.

[Wilson 87] Wilson, J, L., Proceedings of a Workshop on Construction Automation: Computer-Integrated Construction, Dept. of Civil Engineering, Lehigh University, Bethlehem, PA., April, 1987.

Project Management System Selection Guide

E. William East, P.E.[1] and Nie-Jia Yau[2]

Introduction

Although mainframe programs for Critical Path Method (CPM) scheduling have been available for the past 30 year, their use has been typically restricted to only the largest firms. The dramatic decrease in price of hardware and software has made the use of microcomputer programs, generally called Project Management Systems (PMS), for scheduling feasible. By 1986, over 200 vendors and over 100,000 packages have been sold. The use of personal computers and project management software within the construction industry has increase substantially since 1986.

What most project management system purchasers have in common is the large amount of time and effort expended to select a program. The time and effort (as much as one person-year) is required for the software reviewer(s) to develop a background in the capabilities of project management systems. Without practical scheduling this background research often results in an inappropriate selection of a project management system. (EAST '88-1).

The purpose of the Project Management System Selection Guide is to assist in the selection of project management systems by providing the inexperienced software reviewer advice about: (1) the impact of project workload on necessary scheduling features and (2) the impact of the scheduling system on the construction office. The system may also specify several project management systems which meet the offices needs.

Project Management System Selection Guide

The Project Management System Selection Guide attempts to model two types of consultation which have been conducted by the Construction Management Team. The first type of consultation provides: (1) an explanation of features project management system features which should be specified and (2) list of items to consider in the implementation of the system. The second type of consultation provides: a list of project management systems which best match the required project management system features. The following paragraphs of this section will explain the flow of data and inference through the selection guide as illustrated in Figure 1.

Construction Office Environment

The first task of the system is to obtain the information necessary to make judgement about the needs of the particular office. This is illustrated in Figure 1 by "INPUT OFFICE INFO." Four areas of the construction office are considered: (1) projects to be scheduled, (2) construction contract restrictions, (3) personnel considerations, and (4) available computer systems. Each of these items is briefly described below and illustrated in Figure 2.

---

[1] Principal Investigator, U.S. Army Corps of Engineers, Construction Engineering Research Lab, Champaign, IL.

[2] Research Assistant, Department of Civil Engineering, University of Illinois, Champaign, IL.

(1) Projects to be Scheduled: The total size and type of an office work load are the most important factors in selecting a project management system. The number of activities and projects is proportional to the level of sophistication necessary to access that data. The type of office is proportional to the need to manage workers and profit; the construction manager typically has the least need and the contractor the greatest.

(2) Construction Contract Restrictions: There are three ways that an owner may use contract language: (a) to limit the network model which may be used, (b) to require a specific project management system, and/or (c) specify electronic data transfer.

(3) Personnel Considerations: The four items which an office should be evaluated on prior to the selection of a system are: (a) computer systems experience, (b) available time, (c) staff turnover, and (d) multiple sites. These factors will impact the need for training and system customization.

(4) Available Computer Systems: There are four computer related issues to review prior to purchasing any software. These are: (a) operating system, (b) Random Access Memory, (c) hard disk storage and (d) current computer usage. These factors may actually prohibit use of the scheduling system at the construction office.

Filter Rules

As the user enters information into the frames which contain the Construction Office Environment described in the previous section forward chaining rules may be fired. The authors have called these rules "filter" rules because the result of applying these rules is a "refined" view of the office in terms of the system features required for a particular office situation. An example of one of these rules is shown in Figure 3.

These rules also generate an explanation of the conclusions drawn from the office information. Figure 4 shows one portion of the output a user may obtain in conjunction with the checklist of required project management system features. The authors have attempted to (1) identify assumptions made and (2) provide practical guidance on system implementation at each feature which is required at a particular office.

Project Management System Environment

This section briefly describes the representation which is used by both the required system features and the project management system database. The project management system environment, shown in Figure 5, is composed of six general frames: (1) system flexibility, (2) input and editing, (3) analysis tools, (4) cost and earned value management, (5) resource management, and (6) system requirements.

(1) System Flexibility: Flexibility in project management systems provides the user the ability to focus on the data necessary for a particular task. On large projects flexibility is essential to the effective use of a project management system (EAST '88-3). Six features provide increasing levels of flexibility in a project management system: (a) number of activities, (b) activity codes, (c) activity code libraries, (d) customized reporting, (e) customized menus, and (f) data base access.

(2) Input and Editing: Selecting specific activities to be edited using selective sorting may be, over the long term, one of the most time saving features of a project management system for large projects or large work loads.
(3) Analysis Tools: Analysis tools are available to assist in determining the correctness of a contractor's schedule. These tools are (a) on-screen reports, (b) reports which

compare two versions of the same schedule, (c) on-screen graphics, and (d) statistical progress analysis.

(4) Cost and Earned Value: Cost management features are one of the most diverse aspects of project management systems since contractors, construction managers, and owners all have different cost management needs proportional to their need to monitor profit.

(5) Resource Management: Resource management is a difficult area to review because of inconsistent terminology. Contractors in particular would like to have an automated tool to assist in planning manpower allocation.

(6) System Requirements: Offices with IBM or compatible computer systems should have no problem finding several systems which meet their needs, however, the additional hardware support a project management system requires include (a) additional RAM, (b) hard disk storage, and (c) additional computers.

Matcher

Now that the checklist of required system features and the frames containing those features have been completed the user may use the system to find the best match between their office's needs and the capabilities of systems in the project management system database. This is accomplished through a set of LISP functions collectively called the "matcher." The matcher takes the data from the system requirement frames and compares it to the data in the project management system database. The result of this matching is a list of project management systems which identify those features which coincide.

Project History

The prototype of the Project Management System Selection Guide was completed in December 1987 to complete a course requirement at the University of Illinois. This system programmed in GCLISP also included user interface functions found in OPS-5. The program was developed on IBM compatible computers with a minimum of 1.0 MB of extended memory. This development included approximately forty rules and required approximately two person-months to complete[3].

Once the system was completed porting to GOLDWORKS was begun (1) to learn that expert system shell and (2) to develop a more pleasing user interface. Figure 3 shows a typical GOLDWORKS's rule. Figure 4 shows a section of what the user might see on the screen if thier office had a high rate of turnover and utilized multiple project offices. The GOLDWORKS development was conducted on IBM PS/2 model 80 computers with a minimum of 8 MB of extended memory. This development included approximately forty additional rules to manipulate user interface functions and required approximately additional one person-month to complete.

System Verification

Verification of this prototype will be a very important issue with project management software vendors. In the current guidance system only example project management systems are included to avoid any potential conflicts. Although some have suggested that

---

[3] Development times noted includes time required to learn the referenced program language or expert system shell.

every system vendor should comment on the selection method the authors do not believe that this is necessary.

The authors believe that most vendors will support the system because, if kept current, the system provides basic market research about the needs of their construction customer. The selection guide indicates: (1) which markets their product do not adequately serve and (2) the features needed to enter these additional markets. Vendors will be able to more easily assess the impact of additional features on their product with this information.

The most important reason why the authors feel this system will move beyond the prototype with minimal protest is that if the system is completed every organization which uses it will save a minimum of one person-month in the selection of a project management system.

A peer review verification process has already begun with the publication of a background paper (EAST '88-1) and the knowledge base contained in the CGLISP prototype (EAST '88-2). It is anticipated that readers of these articles, including project management system vendors, will comment on the validity of the selection method.

Conclusion

There are significant problems with the way many offices select project management software. These are (1) high resource commitment, (2) reliance on unsubstantiated data, and (3) incomplete reviewer perspective. The Project Management System Selection Guide, once completed will reduce these problems to a great degree by sharing the knowledge of: (1) features which impact an office's ability to manage construction schedules, (2) practical problems which may occur during implementation of the system, and (3) overall project management issues which need to be addressed.

In the final analysis, the use of any automated system depends not on programming techniques but on the benefits received from using the program. Educating buyers about the use of project management systems at their particular office will significantly reduce the time required to purchase a project management system. The authors are confident that this savings will drive future project development.

References

(EAST '88-1) East, E. William, "Approaches to Selecting Project Management Systems," ASCE Fifth Annual Conference on Computing in Civil Engineering, March 1988. ppg. 52-60.

(EAST '88-2) East, E. William, "A Knowledge-Based Approach to Selecting Project Scheduling Systems," ASCE Journal of Computing in Civil Engineering, scheduled for October 1988.

(EAST '88-3) East, E. William, Kirby, Jeffrey, "Using Project Management Systems at the Construction Field Office", USA-CERL Technical Report (submitted March 88).

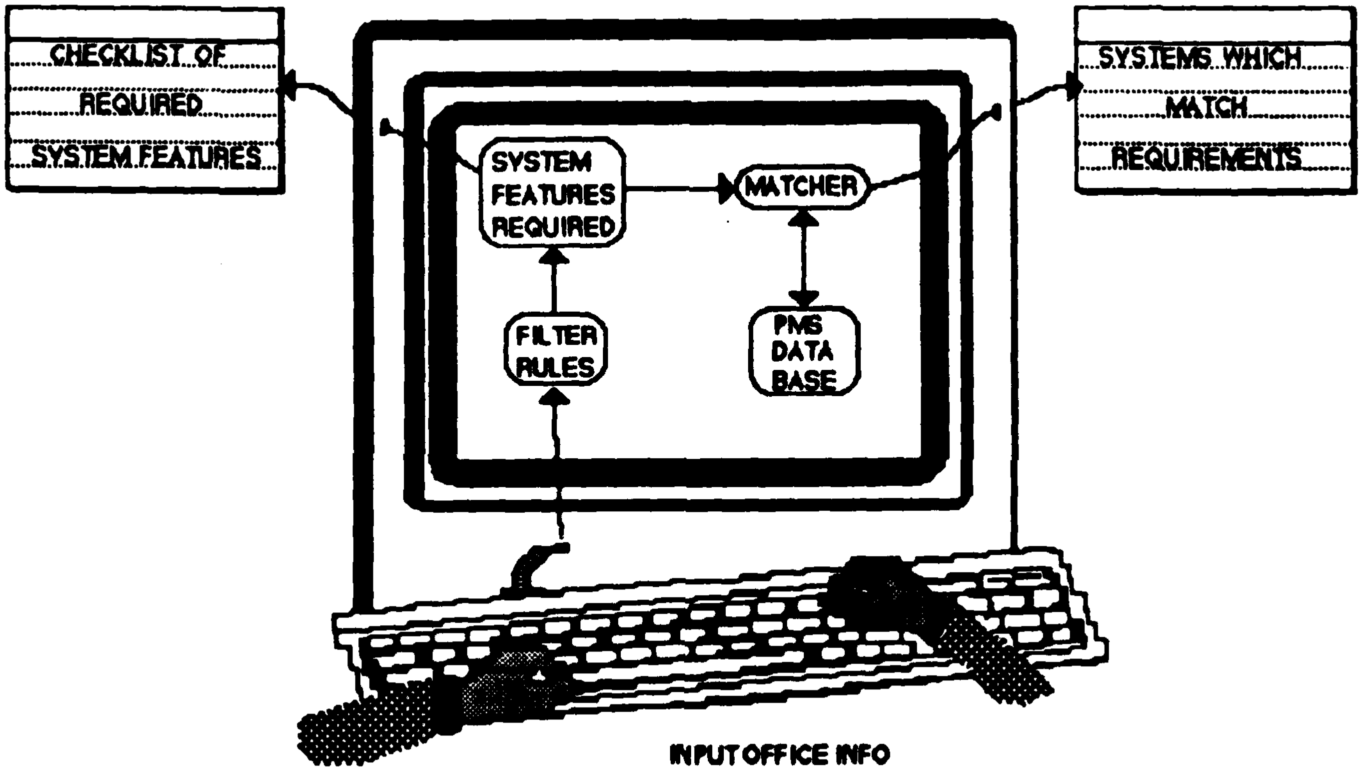


FIGURE 1. SYSTEM ARCHITECTURE

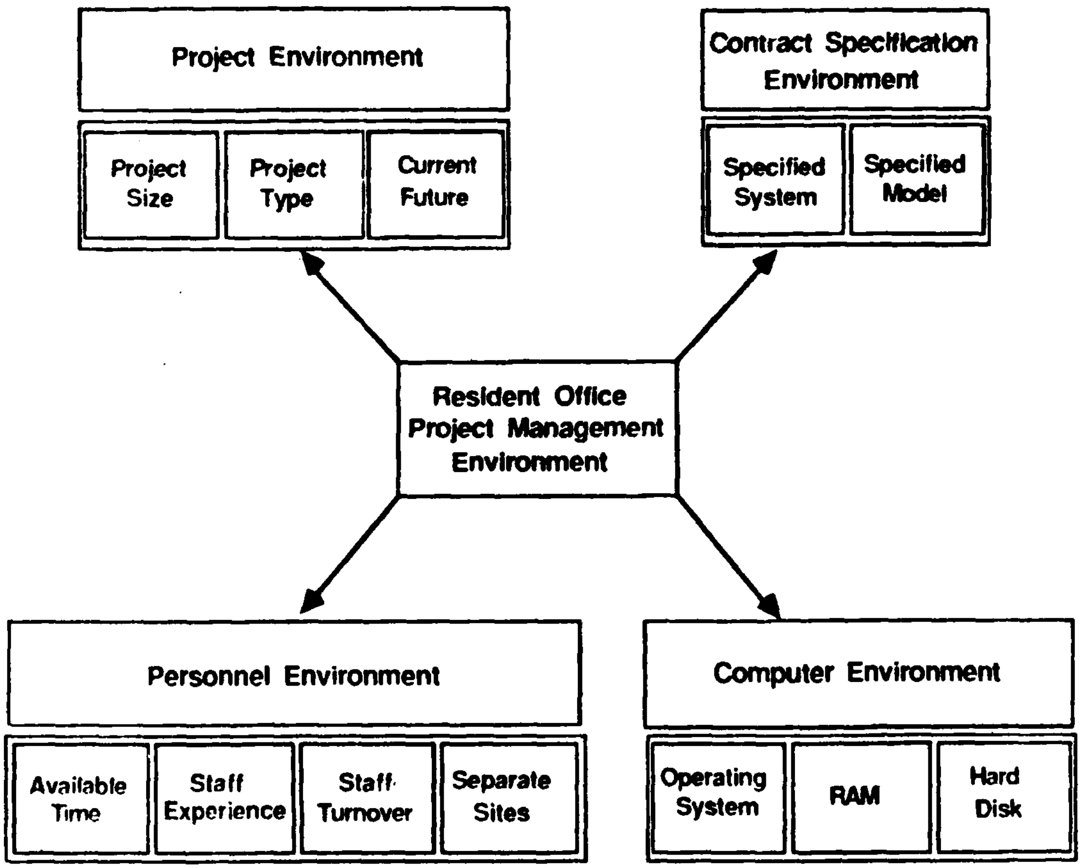


FIG. 2 -TAXONOMY OF RESIDENT OFFICE ENVIRONMENT

```
(DEFINE-RULE PROJECT-ACTIVITY-CATEGORY-HIGH
      (:print-name "IMPLICATIONS OF HIGH NUMBER OF ACTIVITIES"
       :doc-string ""
       :dependency NIL
       :direction :FORWARD
       :certainty 1.0
       :explanation-string "
Since your office's typical project will be between 500 and 1000 activities your system
should provide many features to assist in the INPUT- EDITING of the activity data.
ANALYSIS-TOOLS features should also be provided to assist you in working with the
large amount of project data found on printed reports. Your system should also provide
moderate FLEXIBILITY which will assist in integrating the system into your office
procedures.
"
       :priority 823
       :sponsor GATHER-OFFICE-DATA)
      (INSTANCE AVERAGE-PROJECT-ENVIRONMENT IS
            PROJECT-ENVIRONMENT WITH ACTIVITIES-ONE-PROJECT ?A)
      (< 500 ?A)
      (=> 1000 ?A)
THEN
      (INSTANCE REQUIREMENTS-INSTANCE IS REQUIREMENTS
            WITH RQMT-INPUT-EDITING HIGH
            WITH RQMT-ANALYSIS-TOOLS MODERATE
            WITH RQMT-FLEXIBILITY MODERATE)
)
```

Figure 3: Typical "Filter" Rule

<RULE> ...

<RULE>**GET-STAFF-TURNOVER
<RULE>**GET-SEPARATE-SITES
<RULE>**IMPLICATIONS-HIGH-TURNOVER+MULTIPLE-SITES
Since your office has both high turnover and multiple project offices implementing a system successfully presents certain unique problems. The problems are that the system must be simple enough for new people at the office to use and complex enough for data to be transferred between you various project offices. The recommended means to accomplish this is to select a project management system which allows customization of reports, menu structure, and data transfer protocol. An ASCII protocol is suggested because it is the most common protocol available as of Spring 88. Keep in mind that the ASCII protocol does not specify the position of fields and values. The user must be careful to insure consistency between the data in the files and the way the systems utilize the data.

<RULE> ....

Figure 4: Sample Output from System Checklist

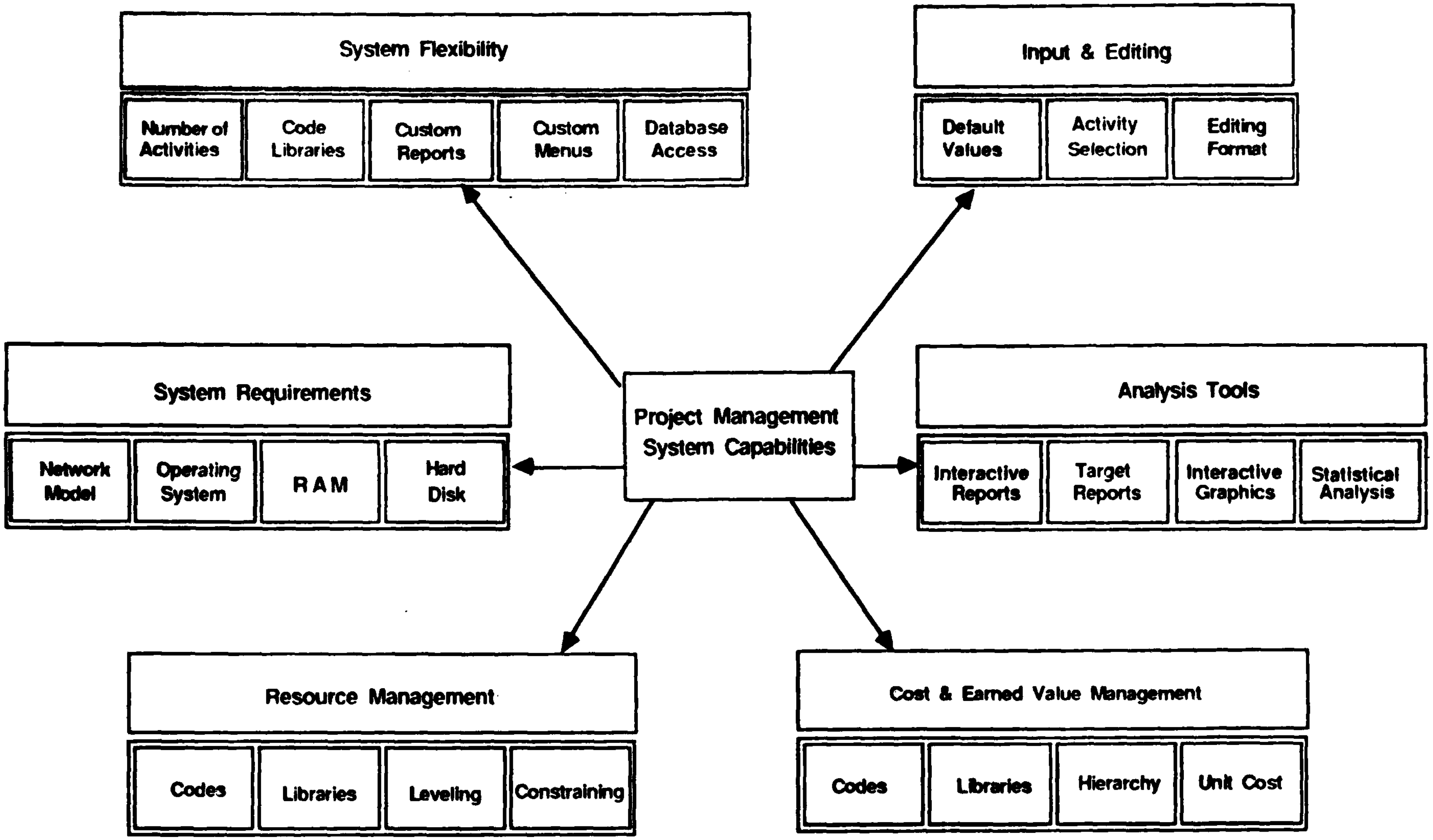


FIG. 5-TAXONOMY OF PROJECT MANAGEMENT SYSTEM CAPABILITIES

MULTI-LEVEL USER DOMAIN KNOWLEDGE IN EXPERT SYSTEM DEVELOPMENT

Thomas M. Gatton[1], Debbie J. Lawrence[1]

1. BACKGROUND

The development of expert systems for real applications must take into account the level of expertise that users have about the domain. Often, however, the range of all the users' knowledge about the domain varies from a limited amount to being close to expertise or the level of the user's knowledge is unknown. The wide range of users' familiarity with the domain introduces problems with systems that only provide heuristic knowledge. While expert systems that provide expert heuristics may be suitable to those users who understand the domain, users who are not knowledgeable about the domain will have difficulty in understanding many of the explanations and reasoning that the expert system may provide. A method for developing expert systems that are independent of the user's domain knowledge is needed to improve the user interface of expert systems.

One method of dealing with different levels of user domain knowledge is to ask the user at the beginning of the session what level of domain knowledge they are at. The system then selects appropriate levels of response for explanations and reasoning for that level. Another technique is to question the user about the domain, at the beginning of the session, and determine the user's level of expertise from the given answers. There is also the more complex approach of developing a "user

1 Principal Investigators, US Army Construction Engineering Laboratory, Champaign, Illinois.

model" that learns what the user knows and tailors its responses accordingly [Berry87].

2. OBJECTIVES

A survey has indicated the lack of integrated instructional knowledge in expert systems and identifies instructional applications as a unique application [GATTON87]. However, there have been expert systems that provide not only expert level knowledge but also the deep knowledge that reasons and explains to basic principles. These types of systems have fallen under the category of intelligent computer-aided-instruction although some of them have been implemented as expert systems.

In a tutoring expert system, there are two basic categories of micsonceptions that a student can make, each with its own type of remedial activity. These two basic groups are:

1) Reasoning errors - The user does not understand the relationships between the parameters and how they are used to reach a conclusion.
2) Factual errors - The user does not understand basic definitions and elements associated with the domain.

The types of errors differ for each user, particularly when a wide or unknown range of domain knowledge exists across the user spectrum. Although explanation facilities are provided with commercially available shell systems, These may be insufficient for some users. Similarly, the level of reasoning that an expert provides for a conclusion may not be clear to all users. A solution to these two user interface problems must include greater depth in the knowledge base as well as an organization of the rules to provide clarity at a level that is comfortable to the user.

3. METHODOLOGY

When the user lacks factual data about the domain, one method of handling the situation would be tutoring with information that would develop the necessary competence. Alternatively, the system could allow diagnosis at a lower level, one that would more closely adhere to the domain knowledge of the user. For example, when a user answers "I don't know" to the system, it may be inferred that this particular user is not very knowledgeable about the domain and either a different type of diagnostic must be performed, or the tutoring must be provided. If tutoring is provided, it must bring the user to a level where the questioning can be resumed at the point where it was left, or allow the user to leave the program with the assumption that the tutorial was the desired consultation. If lower level diagnostics are required, then new rules must be generated to provide adequate reasoning and explanation facilities for the user. These rules could either be transparent to the user or utilized directly to set the value of the parameter that the user needed further direction about.

4. STATUS OF THE WORK

Although it is possible to provide a suitable range of knowledge levels through rules and appropriate structure, this does not address the source of the problem: the limitations of both shell systems and our own lack of understanding about the representation and presentation of knowledge.

Consider this simple example from automobile mechanics: upon reaching a conclusion by unifying all of the parameters in a rule, the following reasons for that conclusion are generated by the system:

> When the starter is working normally
> and the gas gauge indicates that there is gas
> and there is gas going to the carburetor
> and there is no spark at the spark plugs
> Then it is usually the case that there is a
> malfunction in the ignition system.

Upon analysis, the only parameter that has any relationship with the conclusion is the last one. In fact, a lack of spark alone indicates that there is a malfunction in the ignition system. It is evident that the rule actually has two types of knowledge in it:

1) knowledge about the causality of the mechanism, and
2) knowledge about the diagnostic procedures to identify malfunctions.

Yet, in the development of an expert system, these two types of knowledge are intermingled, causing confusion to the user about the logic of a conclusion or a heuristic. Expert system shells should provide a mechanism to incorporate these two types of knowledge so that the user interface can be improved.

5. PRELIMINARY RESULTS

In order to further improve the explanation facilities, a method of integrating deeper knowledge into an abstract domain theory that can be utilized for providing tutorial information would be of great value. This would eliminate the need to cleverly code decision models which have their own limitations, as discussed. The research that we are presently performing involves an abstract domain model that separates diagnostic heuristics from actual component conditions. The implementation will allow a graphic interface to the expert for direct input into the model, thereby eliminating the knowledge engineering bottleneck. Work to combine models from different experts is also underway, thereby allowing distributed knowledge acquisition.

For the present, the decision models, as discussed in this paper, provide an additional tool for expert system developers to improve the explanation facilities and tutorial capabilities of presently available shell system. They offer a method for improving applications through an improved user interface until better shell systems are developed and our own understanding of the communication process is substantially increased.

REFERENCES

Berry, D.C. and Broadbent, D.E. "Expert systems and the man-machine interface", in Expert Systems, , October, 1986.

Gatton, T. M., "Decision Models for Expert System Development: Designing for Multi-Level User Domain Knowledge," Master of Computer Science Project, University of Illinois, 1987.

Symbolic Unified Project Representation

by

Francois Grobler[1] and Simon Kim[2]

Introduction

Construction projects represent the combined efforts of many people and their interactions with subject matter which is complex and knowledge-rich. The building of production quality knowledge-based systems for the construction domain is for this reason difficult and requires much energy and effort. Most of the effort is demanded by the knowledge engineering task; eliciting knowledge and encoding it in a suitable representation scheme. Integrating knowledge with project data represents a further formidable problem.

While many experimental expert systems had been developed for the support of construction projects, most of these systems used data and knowledge representations unique to the individual systems. Until recently formalisms for the representation of project data and related knowledge received little attention. A number of studies now attempt to address this problem [Fenves88, Garrett88, Grobler88, Sanvido88, Scarpon88].

These studies focussed on a level of representation appropriate for base details and integration at the detailed level. As a starting point for this type of research, that approach is entirely appropriate. However, from a top management point of view these capabilities need to be enhanced to operate at higher levels of abstraction, and reflect the transformation process as the project progresses from one stage to the next. With some degree of maturity being approached in the base research, it is now feasible and prudent to investigate an overall framework and a unifying representation scheme. This study will focus on these issues.

---

[1] Visiting Assistant Professor at the University of Illinois at Urbana-Champaign, and Associate Researcher, USA-CERL.

[2] Team Leader, Construction Management Team, USA-CERL.

Nature of the Problem

Incompatibility of representation models is the root problem addressed by this research. The problem manifests itself firstly in incompatibility of data generated at different phases as the project progresses through its life-cycle, and secondly in incompatibility of knowledge representations of knowledge based systems designed to provide assistance in the different project phases.

The processes of Planning, Design, Construction and Operation consist of a myriad of serial and parallel subprocesses. Traditionally these processes demanded their own project representation models. For example, the owner may emphasize operational needs, the architect space utilization, the structural engineer loading conditions, etc, each representing their view of the project differently. Yet there is the necessity to interact and integrate efforts in a conceptual model of the project.

In the last decade it became practical, if not desirable, for nearly every subprocess to be computerized. Alas, the technological capability to exchange files in computer-readable form only served to highlight the incompatibility of the underlying representations used in the subprocesses. For example, the output of estimating systems cannot be used directly by current scheduling systems.

In order to enable the hypothetical scheduling system to use the estimating system output, a large amount of knowledge is necessary to "make sense" of both the data as represented, and the underlying information contents.

A further element of the problem is how to capture knowledge in a form that lends itself to reusability. Reuse is a necessity because of the large amount of knowledge involved, and the fact that independent expert systems operating on subsets of the project data, often require the same or similar elements of knowledge. Knowledge about weather and labor productivity are, for example, needed by both estimating and scheduling systems.

Objectives

The project representation model under development has the following objectives:

- •The integration of all the relevant elements of project information. The term "unified" is used to denoted the integrated representation of graphics, knowledge, and business-oriented data.
- •Allow continuity in the stepwise refinement of the project model as it evolves towards greater specificity.
- •Capture knowledge about the project domain in a reusable way.

The goal of this project is the conceptual development of a Symbolic Unified Project Representation (SUPR) Model and a set of guidelines to allow researchers of knowledge based systems to define their respective object libraries/ knowledge bases in a coordinated manner in terms of the SUPR Model. In so doing researcher will be able to build upon the work of others as well as contribute to the body of knowledge available to the industry as a whole.

Scope

Since the implementation of such a scheme represent a very large body of work, this research will be conducted as a feasibility study to determine the potential and desirability of the proposed SUPR Model, as well as determining the required features and attributes in an structured manner. The primary phases in a project life-cycle, i.e. definition by the user, design, construction, and operation, will be considered. In conjunction with concurrent related research at CERL, the focus of the study will be further restricted to mid-rise office/residential buildings.

Solution Approach

The project model is expected to be based on an object-oriented, knowledge-based representation. This approach was found to perform satisfactory in a number of other studies [Delagar87, Hendric87, Levitt87], albeit of smaller scope. The emphasis here will be to develop a formalism which provides the required capability over the entire scope of this research.

The need for knowledge accumulation and reusability suggests the use of generic, predefined data objects. This approach, proposed for the unification of selected construction data, was successfully applied to progress control at a detailed level [Grobler88]. Generic objects classes contain descriptions of their behavior and their instances can be assembled into semantic models representing the relationships between project data objects. The major conceptual problem addressed by this work will be the extension of that approach to accommodate higher levels of abstraction and provide continuity of representation in the various transformations of the project model.

Methodology

The intended methodology includes the following steps:

Review of user and information needs.

- Definition of the needs of users of project data at the various phases of the project development, i. e. what information is required to perform the function and what transformations take place. For purposes of the feasibility study only the most important needs ("Critical Success Factors") will be considered.

Compile the requirements for the representation scheme.

- Review state-off-the-art knowledge engineering environments to develop an insight into what capabilities they currently provide. Although the representation scheme under development should not be limited by the existing capabilities, the review will provide a useful frame of reference.
- Study the representation schemes used in previous studies such as [Echever88, Grobler88, Hendric87, Levitt87]. This research will also closely collaborate with the project at the University of Illinois [Garrett88], which, in its current phase, emphasizes detailed design and construction aspects.

- Survey developments which may materially change the construction or computing environment in the future, and anticipate their influence on the requirements of the representation scheme. One such development, as an example, is the emergence of neural net computing.
- Develop the requirements for the model and report in generic terms as a set of requirement specifications.

Synthesize a Symbolic Unified Project Representation Model.

- Develop a SUPR Model based on the stated requirements.
- Develop guidelines, to be used by researchers, for the definition of data objects, message interfaces and knowledge representations in terms of the SUPR Model.

Apply model manually to the project definition phase.

- Explore the application of the model to the conceptual description and initial design parameters of projects and simulate, by hand, the encoding of a simple project in terms of the project model.

Implement the model in a prototype system.

- Implement the SUPR Model in a prototype system in a suitable computer environment. In this implementation a limited number of generic data objects, construction procedures and knowledge structures will be defined.
- Examine its performance based on experimental project data, and evaluate the efficacy of the scheme. Special consideration will be given to the usefulness and reusability of the knowledge base constituted by the body of predefined data objects.
- Report the findings, and plan further work.

Future Research

This research is currently in its initial stages. The long-range goals of the project include the refinement of the model, the implementation of a full-blown system, field testing, and stepwise improvement of the system and guidelines of the SUPR Model.

References

Delagar87 De la Garza, Jesus, M. and C. William Ibbs, "Issues in the Construction Scheduling Knowledge Representation", Proceedings of the CIB W-65 Symposium, London, October 1987.

Echever88 Echeverry D., S. Kim and C. W. Ibbs, "The Construction Schedule Generator: AI Tools for the Generation of Initial Schedules", Proceedings of the First Joint ASCE/CERL Conference on Expert Systems, USA-CERL, June 1988.

Fenves88 Fenves, Gregory L., "Object Representations for Structural Analysis and Design", Computers in Civil Engineering: Microcomputers to Supercomputers, ASCE Conference Proceedings, Alexandria, Virginia, March 1988, pp502-511.

Garrett88 Garrett, J. H. Jr. et al., "An Object-Oriented Building System Model for Design and Construction", Proceedings of the First Joint ASCE/CERL Conference on Expert Systems, USA-CERL, June 1988.

Grobler88 Grobler, Francois, Object-Oriented Data Representation of Construction Project Information, Ph.D Dissertation, University of Illinois at Urbana-Champaign, January 1988.

Hendric87 Hendrickson, Chris, et al, "Expert System for Construction Planning", ASCE, Journal of Computing in Civil Engineering, Vol. 1, no. 4, Oct 1987, pp253-269.

Levitt87 Levitt , Raymond E. and John C. Kunz, "Using Artificial Intelligence Techniques to Support Project Management", AI EDAM, Vol 1, No. 1, 1987, pp3-24.

Sanvido88 Sanvido, Victor E., An Integrated Building Process Model, Interim Report on the Computer Integrated Construction Project Sponsored by NSF, Pennsylvania State University, March 1987.

Scarpon88 Scarponcini, Paul, "Integrated Data Base for Buildings", Computers in Civil Engineering: Microcomputers to Supercomputers, ASCE Conference Proceedings, Alexandria, Virginia, March 1988, pp512-520.

Development of Construction Contractor Resource Balancing Algorithms

By Amr Hassanein[1] and John W. Melin[2]

Introduction

The development of automated, integrated cost and scheduling systems for buildings is one of the major research objectives in the construction industry. One of the problems in the development of such systems is the need to determine the reasonable number of crews and the composition of each crew for each trade that can be effectively used in the construction process. This information is needed to determine the activity duration, since varying the crew size and/or makeup will obviously change the activity duration.

The determination of the number of crews and the composition of each crew is a process that requires a large body of knowledge and expertise. The objective of this study is to determine the feasibility of developing algorithms that could, using knowledge and expertise, help in estimating crew sizes and compositions for the different trades involved; based on project parameters. The algorithms could significantly improve the accuracy of construction cost estimating, especially at the early feasibility stage of a project.

---

[1]Research Assistant, Department of Civil Engineering, University of Illinois, Urbana, Illinois

[2]Professor, Department of Civil Engineering, University of Illinois, Urbana, Illinois

Background

This work is part of a larger on-going project to develop an automated integrated cost and scheduling control system; both at the University of Illinois and at CERL. Other efforts have been, and are being implemented to develop other components of such a system [O'Connor, 86], [De La Garza, 88], [Echeverry, 88].

Methodology

The approach followed in this study is the structured interview approach in order to capture the knowledge used by contractors in allocating labor resources to specific tasks, and the determining parameters. The purpose of the structured interview approach is to consider the type of work implemented, the activities involved, and the major jobs the contractors performed, as well as the general factors that affect labor resource allocation to construction activities. The approach is also designed to include specific cases, in order to evaluate what would be done under well defined conditions.

The scope of this feasibility study was limited to four (4) subcontracting areas in general building construction. These four areas are: masonry, reinforced concrete, mechanical and electrical.

Status of the Work

An interview questionnaire was prepared following the structured interview approach in order to capture the required knowledge as mentioned in the previous section. A representative sample of contractors representing each of the subcontracting areas mentioned previously was selected, and interviews were conducted. The interviews were open-ended and were designed to collect comprehensive descriptions of the techniques used.

The information collected was analyzed. An initial crew design process was developed for estimating the crew size and composition for masonry and reinforced concrete contractors engaged in normal building construction. It was concluded that developing final algorithms appears feasible for other trades such as different aspects of mechanical and electrical.

The General Process of Crew Design

As was mentioned in the previous section, a general process of crew design was developed. This process (Figure 1) consists of 14 items. Four types of information are incorporated in the process as follows:

a. Job data which are included in items 1, 2 and 3.

b. Contractor data which are included in items 4, 5, 7, 10 and 11.

c. Construction industry common data which are included in item 13.

d. Data analysis which is included in items 6, 8, 9, 12 and 14.

A brief description follows:

a. Job Specific Data:

(1) the client time constraints (durations),

(2) the quantities of work to be done, and

(3) the job specifics such as the complexity of the work to be done.

b. Contractor Data Base:

(4) Basic productivity rates: These are based on the basic generic unit crew (see item 7) under normal job conditions.

(5) Productivity adjustments: These adjustments take into account the different factors of job specifics that may exist.

(7) Generic crews: These consist of the basic mix ratio that the contractor would like to maintain in his crew for each type of activity.

(10) Crew constraints: These combine union constraints as well as contractor constraints that would insure an efficient output and adequate control.

(11) Crew adjustments: These adjustments determine the final working crew. They depend mainly on job specifics and may differ significantly from one contractor to the other.

c. Industry Data:

(13) These are rules that were found to be common among most of the contractors interviewed. They reveal different issues that contractors consider, and the way they handle such issues as adverse factors, crew size, and work flow.

d. Data Analysis:

(6) Adjustment factors are selected and applied to the basic productivity rates.

(8) The adjusted productivity rates, combined with the quantities of work on the job, would then yield the required number of days of generic unit crews.

(9) The number of work days of generic unit crews combined with the client time constraints would next yield the basic crew for the job.

(12) The crew constraints would then be applied to this basic crew size, and also the crew adjustments would be applied in order to reflect job specific conditions.

(14) The common rules would finally be applied to the adjusted crew in order to adapt it to the various stages of work.

Future Research

The following step will be to validate the initial algorithms that were developed for masonry and reinforced concrete. The contractors previously interviewed will be revisited and the algorithms will be tested on actual projects, and adjusted accordingly.

A representative sample of mechanical and electrical contractors will be selected and interviewed in order to determine the feasibility of developing initial algorithms for mechanical and electrical trades.

The knowledge captured in the process will then be prototyped to demonstrate the function of the resource balancing algorithms unit as part of the large automated integrated cost and scheduling control system.

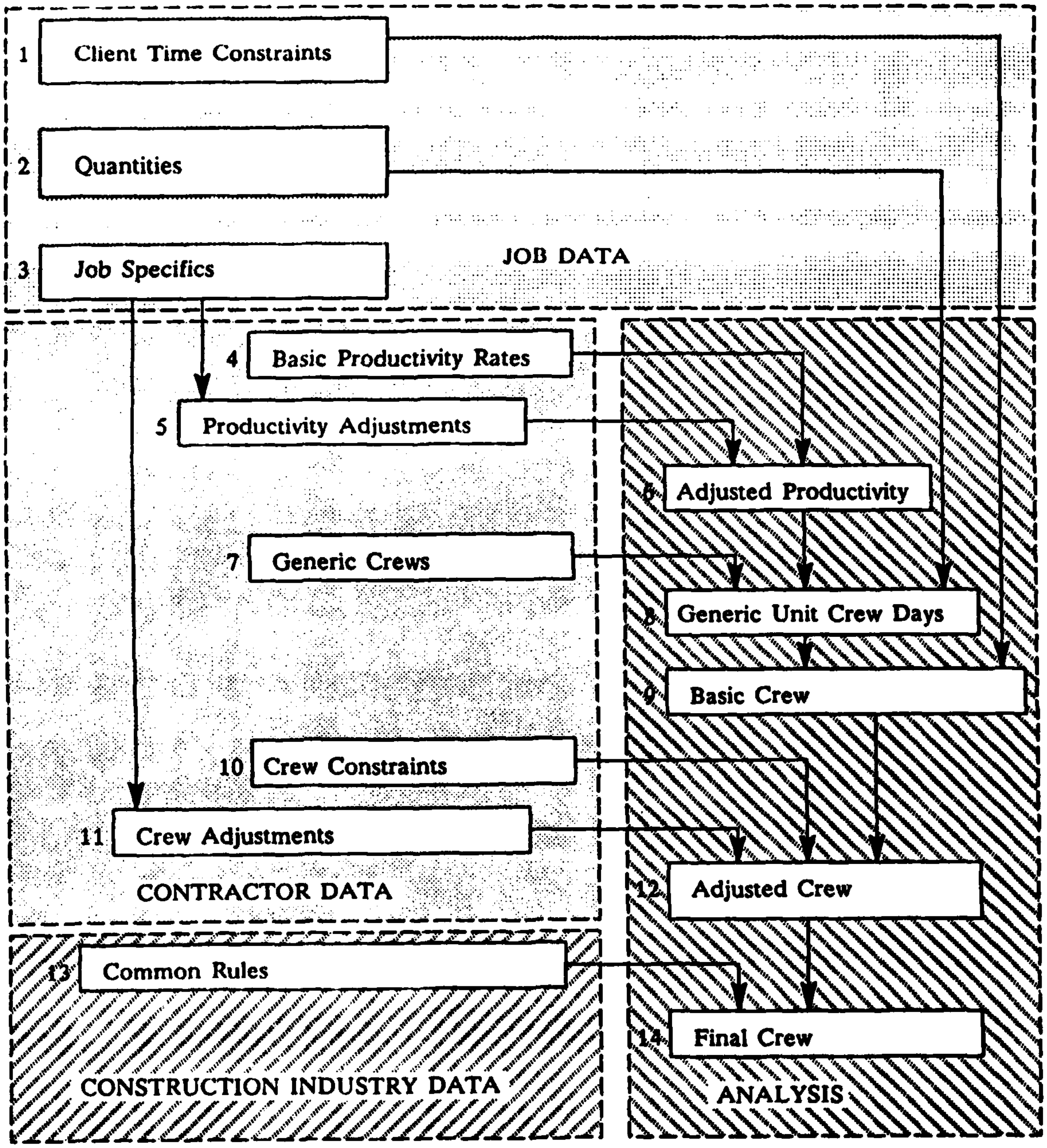


Figure 1. The General Process of Crew Design

References

[McGartland, 85] McGartland, M., Hendrickson, C.T. "Expert Systems for Construction Project Monitoring," Journal of the Construction Engineering and Management, ASCE, Vol. 111, No. 3, September 1985.

[Hendrickson, 87] Hendrickson, C., Martinelli, D., Rehak, D. "Hierarchical Rule-Based Activity Duration Estimation," Journal of the Construction Engineering and Management, ASCE, Vol. 113, No. 3, September 1985.

[De La Garza, 88] De La Garza, J.M. "A Knowledge Engineering Approach to the Analysis and Evaluation of Construction Schedules for Vertical Construction," Unpublished Ph.D. thesis, Department of Civil Engineering, University of Illinois, 1988.

[Echeverry, 88] Echeverry, D., Ibbs, C. W. and Kim, S. "The Construction Schedule Generator: A I Tools for the Generation of Initial Construction Schedules," Unpublished research, University of Illinois.

[Hendrickson, 87] Hendrickson, C., Zozaya-Gorostiza, C., Rehak, D., Baracco-Miller, E., and Lim, P. "An Expert System for Construction Planning," Publication No. EDRC-12-16-87, Civil Engineering Department, Carnegie Mellon University, May 1987.

[Levitt, 87] Levitt, R. E. and Kunz, J. C. "Using Artificial Intelligence Techniques to Support Project Management," Journal of Artificial Intelligence in Engineering Design, Analysis and Manufacturing, Vol. 1, No. 1, 1987.

Knowledge-Based HVAC Diagnostic System : An Artificial Intelligence/Expert Systems Approach

Jason Huffer[1]

Introduction

This project represents an effort to apply new expert system technology to energy conservation programs. To achieve this goal in a most cost-effective way, all of the work has been restricted to personal computers (IBM-compatible) and existing data sensors.

Nature of Problem

The variety and nature of malfunctions in currently installed Army HVAC systems are often beyond the repair experience of site maintenance personnel. Consequently, it is often necessary for DEH to contract diagnostic and repair problems. This increases Government costs, and results in additional downtime, occupant discomfort and lost productivity.

Objectives

The objective is to develop a continuously operating diagnostic system for standard Army HVAC control panels that will monitor the HVAC system, perform range-checking, and do trend analysis on the data. When the system finds an error or the operator identifies a malfunction, an expert system diagnostic session will begin. The expert system will pinpoint the nature of the malfunction, query the user for additional information if necessary, and recommend appropriate actions. This technology will create a cost-effective HVAC diagnostic tool for Army maintenance personnel.

Methodology

Past work in HVAC diagnostics such as that at the University of Colorado (1) is being expanded to include more sophistication, requiring less operator knowledge. The varied knowledge bases required for HVAC diagnostics and data input requirements have ruled out the use of commercially available expert system shells. Therefore, a root language was chosen.

Much of the graphics work at CERL has been done in Turbo Pascal 3.0 (2). An interface between Turbo Pascal and Turbo Prolog was considered; however, the interface would have only allowed Turbo Prolog to pass information to Turbo Pascal and not vice-versa. A solution was suggested by Chabris (3), which presents several search strategies and inference engines in Turbo Pascal. Using these algorithms, and with modifications to the University of Colorado knowledge base, a program was developed using the flow chart in Figure 1 as the goal.

---

[1] Research Assistant, Department of Nuclear Engineering, University of Illinois, Urbana, Illinois

Note from Figure 1 that, during the installation of the software, graphic screens are drawn for each system being configured. System setpoints and alarm conditions are set and specific problems recurrent in each system are entered. When the systems are configured, data input starts and system data points are read continuously and sequentially through the systems. Upon identification of a problem, data analysis component passes its information to the inference engine. The inference engine then forward-chains through the context file (see Figure 2) and then backward-chains through the rulebase file. If additional information is required to reach a conclusion, the inference engine will query the operator until the conclusion is reached. The operator is then presented with potential solutions and the given list of options.

The diagnostic program has six different permanent knowledge bases: temp.knb, press.knb, hum.knb, odor.knb, noise.knb and airflow.knb. The data acquisition tools may fire the temp., press., hum. or airflow knowledge base to be joined with the specific system knowledge base(*.kbs) made during the initial configuring. The context file is generated in a similar way: the latest data facts are joined with system facts from the initial configuring. Therefore, each diagnostic session that the operator performs has a unique context/rulebase file.

As the number of HVAC components in a system increases the number of possible problems with those components and their interaction with each other explodes. This combinorial explosion of field problems is matched by the combinorial potential of the context and rulebase files.

There is also an operator break-in capability, so system faults not detected by the program can still be diagnosed, (e.g. odor and noise problems). This mode represents a traditional expert system with direct query/answer. Once the problem is found, the operator will be given suggestions of possible solutions and queried as to his/her actions. There are also options that allow the operator to see the line of reasoning to the conclusion and to view the current rulebase.

A routine maintenance list is also presented and printouts of the routine list and the graphics are available. Interpretation of the temperature, pressure, flow, and humidity data points is being handled by commercially available software (4).

Status of Work

A Pascal Inference Engine has been modified to accept the different forms of input and the prototype knowledge bases are in place. The data acquisition tools are currently being modified to generate the required text files. The expert system portion of the program is complete and field testing on an actual Army HVAC system will begin this summer.

Preliminary Results

As a logic test, the Pascal Inference Engine was given the same knowledge base as several commercially available expert system shells. In each case, the Pascal Inference Engine arrived at the same conclusions as the commercially purchased shells, with only a slight difference in the logic path and confidence factor. Therefore, the performance of the Pascal Inference Engine is almost identical to other expert system shells in terms of functionality and reliability.

Pascal is a root programming language so that software can be easily expanded as new developments occur. Since the program is written in a general form, any Army mechanical system can be monitored and diagnosed with the appropriate knowledge bases installed.

Expected Further Research

Pascal has the capability not only to monitor the data signals but to act as a proportional-integral (PI) controller. This task involves making coarse adjustments to the output signal proportional to the change in the input signal, followed by fine tune adjustments to stair-step to the desired setpoint (integral).

By using this PI capability CERL plans on having the Pascal Expert System literally controlling an HVAC system. The expert system would ask for information such as building schedules and average populations to make the most efficient use of the HVAC system. Thus, CERL has the potential to use one operating system to monitor, control, analyze data and diagnose HVAC systems.

References

1. Hildebrand, A., "A Knowledge System for HVAC Diagnostics" Graduate School of the University of Colorado, 1986

2. Turbo Pascal and Turbo Prolog Development Languages Borland International,4585 Scotts Valley Drive, Scotts Valley, Ca 95066

3. Chabris, C.F., "Artificial Intelligence and Turbo Pascal" Dow Jones - Irwin, Homewood, IL 60403, 1987 pp 132-134

4. "Turbo Data Acquisition and Control Tools for Metrabyte Dash-8 and Dash-16 (IPC-TP-)", Quinn-Curtis, 49 Highland Ave. Needham, MA 02194, ch. 9

HVAC Diagnostic System Flow Chart

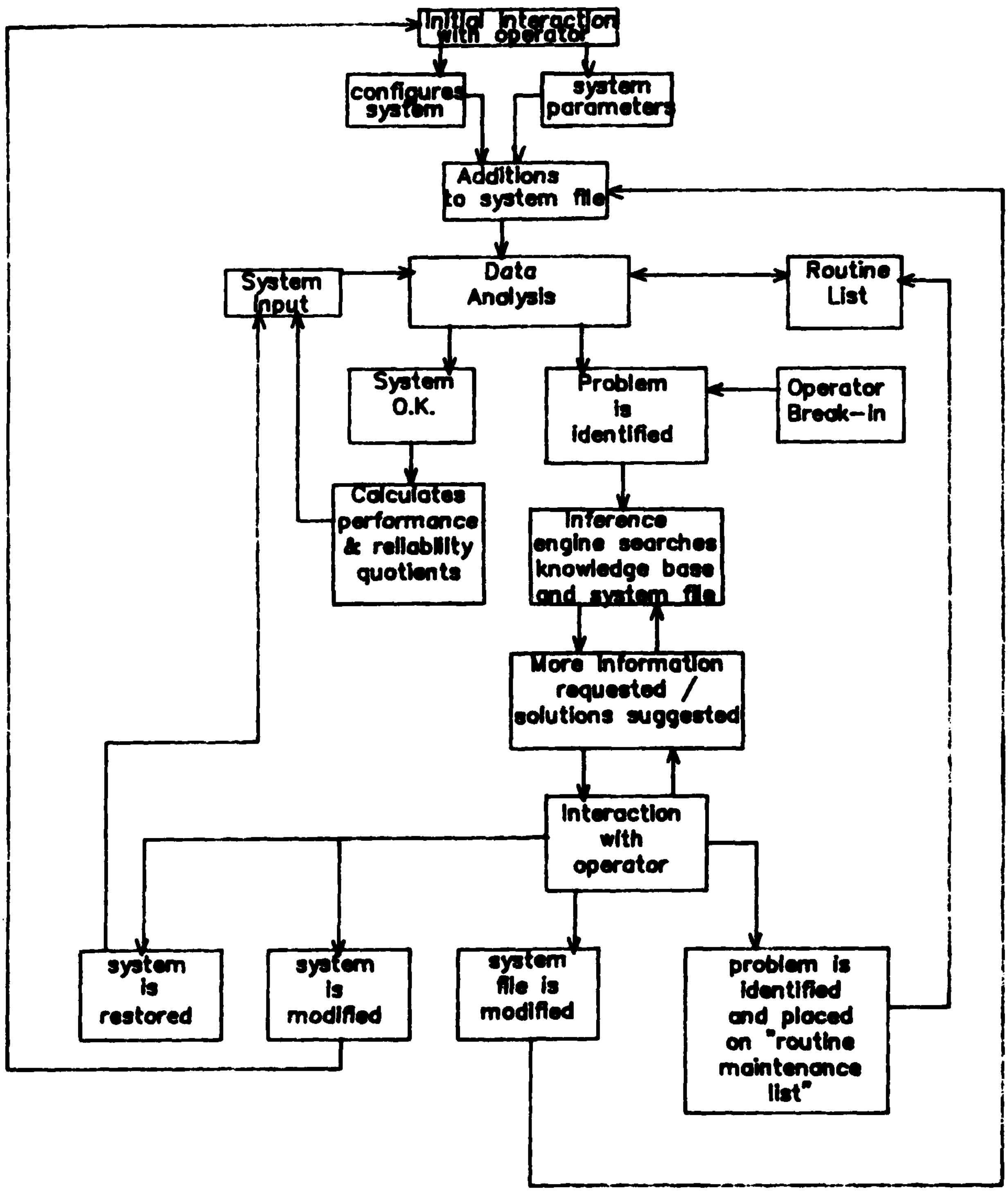


Figure 1

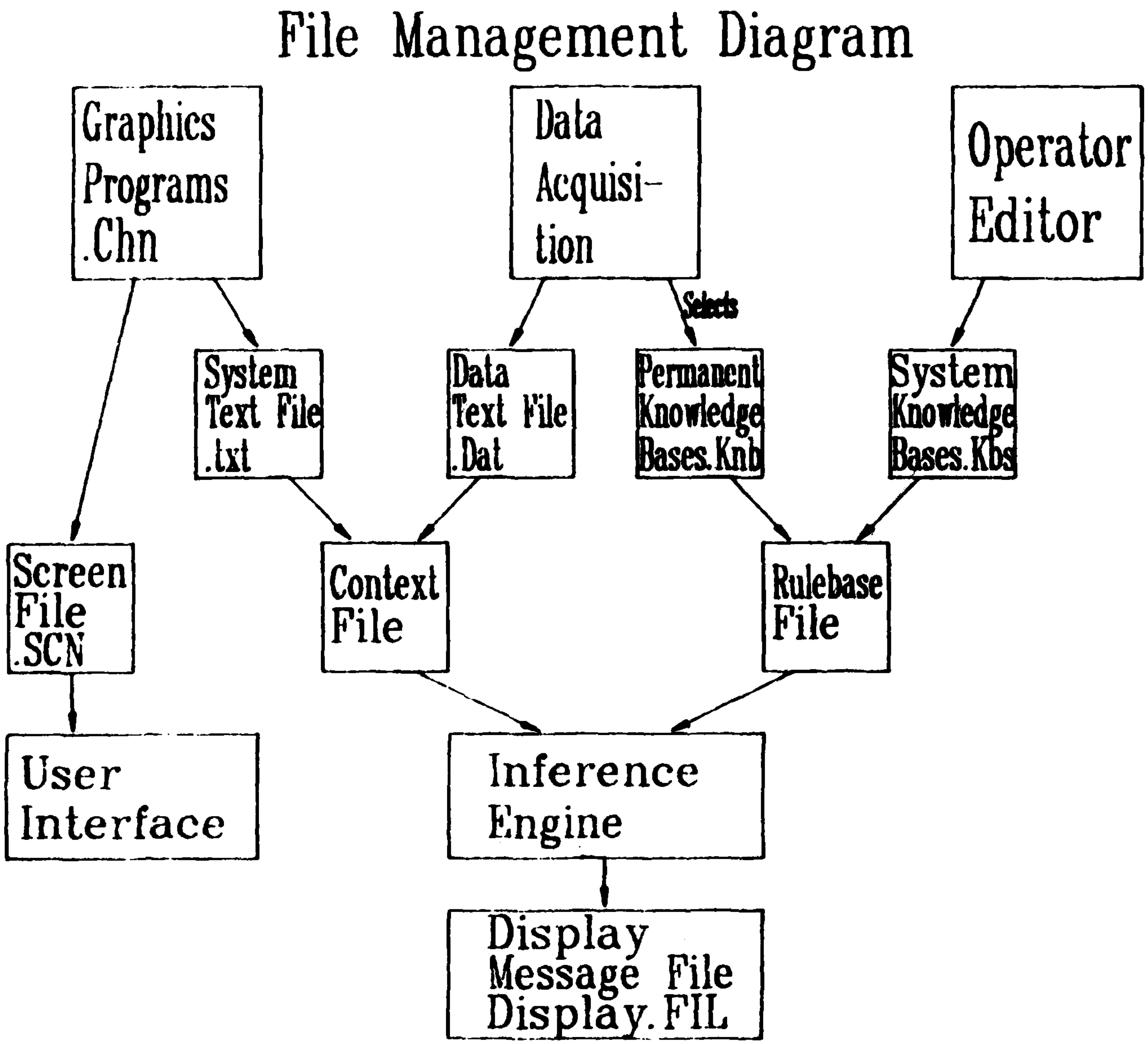

File Management Diagram
Graphics Programs .Chn
Data Acquisition
Operator Editor
Selects
System Text File .txt
Data Text File .Dat
Permanent Knowledge Bases.Knb
System Knowledge Bases.Kbs
Screen File .SCN
Context File
Rulebase File
User Interface
Inference Engine
Display Message File Display.FIL

Figure 2

Expert-MCA

A Knowledge-based Natural Language Interface to the Construction Appropriations, Control and Execution System

Sandra Kappes
U.S. Army Corps of Engineers
Construction Engineering Research Laboratory
Champaign, IL 61820-1305

<u>Background</u>

Before any construction begins on U.S. Army facilities, each project must meet the requirements defined by the Military Construction, Army (MCA) program development process. This process ensures that every project follows the guidelines established by the Department of Defense (DOD) to meet combat capability through the balanced allocation of resources. The DOD five Year Defense Program (FYDP) lists the Army's requirements for construction. It is designed to provide a construction program that is consistent with current Army stationing plans, resources and budget objectives.

As a project steps through the MCA development cycle it must be justified, reviewed, revised, programmed and budgeted. Finally, before construction can begin it must be approved at the Congressional level. As a result, this process generates considerable amounts of data. The Construction Appropriations, Control and Execution System (CAPCES) is the database system used to store this information. It contains descriptive information on more than 15,000 active projects with approximately 500 data elements stored per project. CAPCES allows tracking of a project from inception to disposal. It contains data on individual projects such as cost, activity start and completion dates, approval status and codes for grouping and sorting projects.

CAPCES was developed in the FOCUS database environment. It resides on an IBM mainframe in Dallas, Texas and provides support to users throughout the Army Corps of Engineers. These include the facilities engineer/master planner at the installation level, Major Subordinate Commands (MSCs), Major Army Commands (MACOMS), Office of the Chief of Engineers (OCE), and Headquarters, Department of the Army (HQDA).

To access the data in CAPCES, a user must have considerable knowledge of the structure and content of CAPCES, the MCA process, and the FOCUS database language. Because of this, effective use of this data is inhibited because the retrieval process is too complex for the casual user. In response to this problem, USA-CERL is developing a knowledge-based natural language interface (Expert-MCA) to the CAPCES database. By providing a front-end to CAPCES that has domain knowledge and a natural language interpreter, the casual user will be able to retrieve data using "ordinary" English queries.

Capabilities

Expert-MCA processes queries by relating phrases within the query to its knowlege base and producing FOCUS database commands to access CAPCES. Queries range in their degree of difficulty. The simplest type is the direct query, which relates terms directly to database fieldnames. An example is **"Show Family Housing projects at Ft. Sill"**. The system relates the terms **"family housing"** and **"Ft. Sill"** to the database field values **"CATCD5 EQ 71$$$ OR 72$$$"** AND **"STATION EQ Ft. Sill"**, respectively.

Other queries must be translated through the use of procedures. A procedural query can be extremely simple. For examle, in the query **"Show**

the cost overrun amount for projects at Ft Sill", the phrase **"cost overrun amount"** is translated using the calculation: **"cost overrun amount = current working estimate - program amount"**. A more difficult procedure can involve both the retrieval of data values and the system's knowledge of the MCA Cycle. To interpret the query **"What is the status of FY90 MCA design?"** Expert-MCA would have to retrieve all FY90 projects, its knowledge of project status and each project's current location within the MCA cycle. This requires a deep understanding of the MCA process to ensure the accuracy of the report produced. Based on this understanding, the system will also have the ability to answer user queries concerning the MCA process itself. An example of this would be the query: **"What is the latest date to submit projects to OCE for the FY 89 program?"**.

Expert-MCA also has the capability to retain the context between queries. This allows successive refinements to the query without repeating the initial request. An example of this is shown in the following queries:

Query #1: List the FY86 projects at Ft Dix.

Query #2: Which of those are for training?

The key to expanding the knowledge contained within Expert-MCA is its ability to acquire knowledge from the user. When the system tries to interpret a phrase it has never seen before, it interacts with the user to obtain a definition of the phrase. Once it has been defined, it is stored permanently within the system's memory. This ability also allows the user to tailor the system to his/her own specific needs. For instance, one user might define "status" based on the project's design progress, whereas another might define it as the project's status during Congressional review.

Because of these differences, in order to maintain the integrity of the primary MCA knowledge base, and individual knowledge base is created for each user.

Status

A pilot system has been developed using Gold Hill Common LISP software. This system takes English queries and translates them into FOCUS code which is then used to retrieve data from CAPCES.

Terms are the basic units recognized by Expert-MCA's language analyzer. A frame-based format is used to represent these terms. A term can be either a single alphanumeric word or group of contiguous words which constitute a phrase. There are six different term types recoginized by the system: CAPCES field name, CAPCES field value, word/phrase, synonym, procedure, or rule. There is a different frame representation for each of the six different term types. Slot values can be words, phrases, logical or mathematical oeprations between words or phrases, procedures or rules. By defining a term in relation with another defined term, slot values can establish relationships with other frame instances. In the case of synonym frames, a defined term inherits the properties of the frame it is a synonym of.

The language analyzer has two primary modules - a syntactic analyzer and semantic analyzer. The syntactic analyzer is responsible for decomposing the sentence into different term components by using a set of context-free rules. On the other hand, the semantic analyzer scans the meaning of each term and uses the characteristics of the six different term types to refine the internal representation produced by the syntactic analyzer for the input sentence. At this point the internal representation of the user's query consists of three major parts: requested items, sorting items and screening conditions. This representation is similar to the FOCUS command format.

The internal representation is then sent to the reasoner which utilizes different knowledge sources, such as knowledge about the CAPCES database, the user, and the MCA domain to determine how to respond to the user's query. This end result of this process is the generation of the FOCUS commands required to access the CAPCES database.

Current efforts involve the refinement of the existing system. The pilot system contains limited knowledge of the MCA Cycle and CAPCES contents. The knowledge used by the reasoner must be expanded to include more complex reasoning capabilities. One example of this, is the ability to choose between terms with multiple meanings based on the context of the input query.

The Expert-MCA system was developed through a contract with Professor Robert Logcher in the Massachussetts Institute of Technology Civil Engineering Department.

References

1. Military Construction Army (MCA) Program Development, Army Regulation 415-15, Headquarters Department of Army, Washington D.C. December, 1983.

2. R.D. Logcher, M.T. Wang, F. Chen, Expert-MCA: An Expert System for CAPCES, Civil Engineering Department, M.I.T., January, 1988.

EQUIPMENT PROBLEM DIAGNOSIS SYSTEM

Michael R. Kemme[1] and Walter J. Mikucki[2]

Introduction

The U. S. Army Construction Engineering Research Laboratory (USA-CERL) is modifying an ambient air quality monitoring trailer (AAQMT). As originally configured, the AAQMT contains a number of ambient air quality and meteorological monitoring devices. Associated with these devices is an intricate gas sampling system. A microcomputer controlled data acquisition and analysis system was installed during modification of the AAQMT. USA-CERL designed the AAQMT for semi-autonomous operation. Only a weekly preventive maintenance visit is normally required to keep the trailer operating properly. Lower labor costs are realized compared with similar monitoring stations.

Nature of the Problem

When the AAQMT malfunctions, often only a person very knowledgeable with the intricacies of the AAQMT (an expert) can properly diagnose the problem. A number of status signals, available to the microcomputer, track the condition of various sub-systems within the AAQMT. An expert would normally check the status signals and then perform a series of logical diagnostic tests until the problem was determined. The knowledge base an expert uses to proceed with a diagnosis could be used to form an equipment problem diagnosis expert system.

Preliminary Research

Initial research on the equipment problem diagnosis system for the AAQMT has

[1]Principal Investigator, US Army Construction Engineering Research Laboratory, Champaign, Illinois.

[2]Team Leader, US Army Construction Engineering Research Laboratory, Champaign, Illinois

consisted of a contractor developing example equipment problem diagnosis procedures, evaluating microcomputer based artificial intelligence (AI) techniques and determining computer requirements of the recommended AI technique.[3]

Example Equipment Problem Diagnosis Procedures:

The contractor developed example equipment problem diagnosis procedures by simulating six failures in the pneumatic system of the AAQMT. The contractor then debriefed an equipment repair expert to determine some of the reasoning patterns employed while diagnosing each simulated failure. The expert would first examine the status signals and data from the monitoring devices to isolate the general system of the AAQMT most likely to contain the failing component. After the general system was isolated, the expert would use a dependency network to determine the locations for diagnostic testing. By moving sequentially through the functional dependencies associated with the various components, the expert was able to isolate the general system, then subsystem, then specific components likely to be the sight of failure. When the failure was isolated to a set of linearly linked components of equal functional dependency, the expert would consider both the relative level of difficulty in performing component tests (conservation of time strategy) and the relative costs of performing the test and replacing the component (conservation of resources strategy). The expert would continue testing components in a stepwise fashion until the failed component was found.

Evaluation of AI Techniques:

The contractor evaluated three rule collection mechanism AI techniques for use in developing the expert system. Traditional production rule expert systems

---

[3]"Development of an Equipment Problem Diagnosis System," Cognitive Systems Limited, November 1985.

were not considered because these representational approaches do not allow the establishment of collections of rules with various collections applicable in various instances. The contractor concluded that the ability to use different collections of rules in various circumstances would help avoid heavy computational requirements in evaluating the knowledge base and minimize the complexity of the elements the expert must consider as the knowledge base is constructed. An augmented rule based frame system, a multiple rule set system, and an augmented dependency network analysis representation were the three rule collection mechanism techniques evaluated.

Each AI technique was evaluated by the following criteria:

(1) Its ability to represent the supporting knowledge base used by the expert in diagnosing problems,

(2) The ease of building such a system from the supporting information from the expert,

(3) The system's ability to replicate the expert's reasoning process used during diagnosis of component failure,

(4) The difficulty of use in diagnosis,

(5) The ability to build a supporting knowledge base which allows a helpful explanation process, and

(6) The generality to other similar systems for monitoring equipment.

The augmented rule based frame system did not perform well against the comparison criteria. This technique places information associated with each component in a "frame". Nearby components (both physically and connectively) are linked to this frame. When it has been determined that a component is not failing, the linked components would then be investigated. This is somewhat like the process used by humans when a large number of items must be brought under consideration and no hierarchy can be placed over them. Since this technique does not allow for a general view of where a malfunction may be occurring, the contractor concluded that it does not replicate the expert's reasoning process very well. The contractor also determined that the technique would not be easy to use in the diagnosis procedure and would not allow an easy implementation of an explanation process.

The multiple rule set system is an extension of the augmented rule based frame system and eliminates some of the problems associated with the latter technique. This technique allows the use of diagnostic information and rules associated with the specific component under consideration. The information brought under consideration can impose a unifying logic over the component systems moving from generality to specificity. The contractor concluded that this ability to consider the system in general, coupled with subsequent analysis with specific frames of reference would allow this technique to replicate the expert's reasoning process fairly well. The contractor also concluded that this technique would also fulfill the remainder of the comparison criteria and could potentially be used to create the expert system.

The contractor concluded that, of the AI techniques surveyed, the Augmented Dependency Network Analysis Representation (ADNA) technique best met all the comparison criteria. This technique is evolved from work done at the University of Illinois in the Department of Computer Science.[4] ADNA provides planes of rule collections to be used by all component systems. The planes move from increasing generality at the higher levels to greater component

---

[4]Schuster, David 1984, "Functional Dependency Graphs as a Tool to Teach Problem Solving," MCS Report, Department of Computer Science, University of Illinois at Urbana-Champaign.

specificity at lower levels. The actual functional relationships between components as well as specific information associated with components can be captured in a logical network. The contractor concluded that this technique could represent the knowledge base almost as it is drawn out by the expert, that the system would be relatively easy to construct, and that, with a proper inference engine, this technique could replicate the expert's reasoning process very closely.

Computer Requirements:

The contractor conducted experiments with respect to the computational requirements of the ADNA. The experiments were conducted on a 6800 processor running under a UNIX operating system with computational demands occurring from other stations in a multi-user environment. This is a close approximation to conditions in the AAQMT with its data acquisition and analysis software running as additional tasks in parallel. Under these conditions and with component systems of three to four logical levels, the system required less than five seconds to completely update its understanding of all the functional components under consideration as failure sites. It is possible that insufficient CPU time will be available to maintain data acquisition and analysis activities during catastrophic multi-component failures. This is due to the rippling effect of functional damage to other components. However, data acquisition activities would be compromised in these situations anyway.

Future Research

Future research would consist of creating and testing the equipment problem diagnosis expert system. However, funding for such research is not available at this time.

EXCEPTION HIERARCHY BASED KNOWLEDGE REPRESENTATIONS FOR EXPERT SYSTEMS

Rense Lange
Department of Computer Science
University of Illinois at Urbana - Champaign

Sue Chien
Mathematical Systems
Sangamon State University, Springfield

BACKGROUND

Even a cursory overview of the literature will indicate that production rules are by far the best known and most widely used basic knowledge representation scheme for expert systems. Production rules have shown to provide a versatile vehicle on which to base efficient inference mechanisms. However, with the increasing acceptance and use of expert systems, issues related to knowledge acquisition and the nature of domain knowledge have gained in importance. Because the last two issues determine to a large extent the costs involved in developing future systems, improvements in our understanding (and mastery) of the relations between knowledge representation, knowledge acquisition, and system behavior, may turn out to be crucial to the viability of the entire expert system approach.

OBJECTIVE

From a practical point of view, there appear to be two important ways to achieve greater cost efficiency and transparency. First, it is obviously desirable that the knowledge representation language should agree as much as possible with the way in which experts prefer to express their domain knowledge. Further, the representation should cause the inference engine to follow the implicit and explicit reasoning patterns of domain experts. This paper discusses these basic requirements as embodied in the semantics of so-called "exception hierarchies," followed by a presentation of their implementation in the CRITIC shell as currently in use by the US Army Corps of Engineers. Finally, we describe the development of a knowledge engineering interface geared at making the relation between the form and the content explicit to the knowledge engineer, as well as to the end user of the finished expert system.

METHODOLOGY

Our approach to achieve the goals mentioned above is motivated by the following basic theoretical considerations concerning the form and intended meaning of classical production rules.

Notice that the logical form of production rules follows the general format:

(1) <u>if</u> A1 ... and An and NOT B1 ... and NOT Bm <u>then</u> C,

where: C is the conclusion of the rule, and Ai, Bj are the antecedents of the rule.

It can easily be shown that (1) is logically equivalent to:

(2) if A1 ... and An then C or B1 or B2 ... or Bm.

This syntactic change in the format of the rule masks an important difference in the intended meaning of rules (1) and (2). For example, using the format (2) we could write:

(3) if material = acid
then material = corrosive OR material = properly_contained.

A much more intuitive, but logically equivalent, rendition of this rule is:

(4) if material = acid and material = NOT properly_contained
then material = corrosive.

Notice that in (4), the negated condition "material = NOT properly contained" acts as a sentinel for firing the general rule that acids are capable of causing unwanted corrosion. In that sense, negated antecedents can be thought of as exceptions to the more general rules in whose antecedents they occur. Because exceptions themselves are allowed to be derived from other rules (with their own exceptions), nested sets of exceptions can arise. Hence the name "exception hierarchies."

STATUS OF THE WORK

The notion that some antecedents determine the exceptions to a rule, rather than the rule itself, has important practical implications for the knowledge acquisition process. For example, we have found [Lange, Kearney, and Hearn, 1986] that experts tend to state their knowledge in the form of broad rules, which unfortunately often turn out to be too general. It is nevertheless extremely useful to collect such rules because they provide an excellent framework for the entire knowledge base.

In addition, the rules can be shown to expert, either directly, or as used by an inference engine. The expert can then see which rules are too general and proposes exceptions. A typical reaction of the experts is: "Y is correct here, but of course it won't work for X." Given the nature of the representation, the specific conditions surrounding "X" can now be stated as a rule which forms an exception for concluding "Y". We have found that the general outline of the rules is usually correct and that revisions tend to be limited to the addition of appropriate exceptions. Thus, a considerable gain in efficiency is achieved.

PRELIMINARY RESULTS

The CRITIC system implements the basic shell for reasoning with exception hierarchies, and has lead to the development of the now fully operational ESRAM expert system for railroad maintenance. In addition, further work has indicated that rules of format (1) are of considerable theoretical interest. For example, they have a clearly defined logical model which is computable in most practical cases [Przymusinski, 1988]. This means that it is possible to determine exactly how additions or changes affect the content of the knowledge base, and therefore the conclusions that can be reached from this knowledge base. Knowledge engineers can take advantage of such information because it greatly facilitates the debugging process.

To streamline the knowledge acquisition stage, we have developed an editing system which allows rules to be specified in the format:

(5) if A1 ... and An then C unless B1 ... or Bm.

Notice that this statement is equivalent to (1), but its syntax enforces a correct statement of the rules, thus ensuring that the semantic content of the rule base can be determined. The semantic aspect of the editing tool is currently being implemented. Incorporation of this feature into the inference engine, i.e., as an explanatory device for the user, is under study.

REFERENCES

Lange, R., Hearn, L, and Kearney, F. The use of Knowledge Engineering Teams as a Method for the Development of Expert Systems. In: D. Sriram and R. Adey, Applications of Artificial Intelligence in Engineering Problems. Springer Verlag, 1986.

Przymusinski, T.C. On the Declarative Semantics of Deductive Databases and Logic Programs. In: J. Minker, Deductive Databases and Logic Programming. Morgan Kaufmann, 1988.

Knowledge Base on Alternative Construction Methods

by: Thomas R. Napier [1]
Ruth K. Garrett [2]

INTRODUCTION

The U. S. Army Corps of Engineers most commonly employs a single approach in its acquisition (design, procurement, and construction) of military facilities. Other facility acquisition strategies and innovative building technologies, however, are finding increasing acceptance and use in other construction markets, both private and public. The House of Representatives Committee on Appropriations (HAC) has recognized potential advantages in these non-traditional practices and methods, and has requested the Department of Defense to expand their uses where advantageous. The HAC referred to various examples of "Alternative Construction Methods" citing both building technologies and procedural methods, including design/build contracting, modular and prefabricated construction, standard designs, the use of performance specifications, and exploration of nontraditional methods and materials.

NATURE OF THE PROBLEM

There is, however, little experience within the Corps upon which to build expertise equal to that of the traditional construction practices. Therefore, a system is needed to capture the experiences of military projects using alternative construction methods and transmit this experience and knowledge to personnel involved with future projects.

---

1 Principal Investigator, US Army Construction Engineering Research Laboratory, Champaign, IL.

2 Associate Investigator, US Army Construction Engineering Research Laboratory, Champaign, IL.

OBJECTIVES

The Knowledge Base will support decision making and management of military construction projects involving alternative construction methods. The Knowledge Base provides lessons-learned and advisory information based on similar project experience. Personnel without extensive first-hand experience with a given alternative construction method will be able to access information relevant to the planning and execution of projects using that method. Such information will contain data for similar projects, such as (costs, durations, etc), as well as cause-and-effect relationships, conclusions, and advisory information. The basis for decision making is improved, enhancing the probability of a successful project.

The first topic being developed for the Knowledge Base is "Design/Build Construction". This method of design and construction differs fundamentally from traditional Corps practices by integrating both facility design and construction responsibilities under one contractual entity. By contrast, the typical Corps facility design is executed under one contract; that design is competitively bid for the construction contract. Other topic areas to be developed include "Architectural Fabric Structure Technology", "Modular Construction", and "Third Party Contracting". Still other topics may be identified, depending on the Corps requirements.

METHODOLOGY

An automated Database has been developed to provide lessons-learned and advisory information to Corps project management and technical personnel who are involved with alternative construction methods. Inputs for the Database include quantitative information (such as costs, time durations, occurrence of change orders, etc) and descriptive data (such as specification contents, technologies' descriptions, quality levels, etc). Inputs also include interpretive information such as project results and conclusions, cause and effect relationships, lessons-learned, and recommendations for future projects. This interpretative information is generated by project participants from the Corps and the contractors. It represents more of the "base of knowledge" for the project, beyond simple numerical data. Entries for all projects follow a prescribed format. Information is consistent project-by-project. As additional projects occur, the Database will be updated by USA-CERL personnel or other supporting entity.

With the Database, the user (such as the Project Manager for a specific Corps project) makes decisions by identifying similar projects from the Database, then retrieving the relevant information. The user can retrieve all data fields for any specified project, any specific data field from any specific project, or a specific data field from all of several projects. Quantitative data, such as cost and time information are intended to assist the project manager in initial project planning activities. Other descriptive and interpretive information, such as technical specification contents and description of project results, are intended to support activities occurring throughout the project's execution. Given experiences and lessons-learned drawn from previous similar projects, the user formulates his/her own decisions on a project-specific basis.

The final stage of development is to produce an expert system, utilizing the Knowledge Base from the previous stage. The Database has its limitations. It is based only on specific project experience. Similar conditions may reoccur only infrequently, if at all. Therefore, a user must still make his/her own interpretations and applications to any specific project. This is a useful aid, in that it does provide institutional experience that would not otherwise be available to the user. However, it is acknowledged that the Knowledge Base is not an example of true knowledge engineering. A more broadly applicable knowledge base will be developed, generalizing expert input to provide an expert system that can be queried in an advisory capacity.

STATUS OF THE WORK

Twenty-three Military Construction, Army (MCA) projects employing an Alternative Construction Method have been or are being monitored. The majority of the projects involve Design/Build construction; some involve Architectural Fabric Structure Technology and Modular construction. Their progress has been recorded. Standard data formats have been developed for each Alternative Construction Method. The appropriate data fields have been defined and an automated, menu-driven Database management program has been developed. Project data available to date has been entered and is available for retrieval.

PRELIMINARY RESULTS

Military construction projects involving Alternative Construction Methods are being monitored through direct contact with project participants, including Corps project management and engineering personnel and contractor personnel. The information is being interpreted and input into the Database by USA-CERL personnel. To date, the Database has been consulted by HQUSACE personnel for lessons-learned information, which is supporting decisions and activity relative to Design/Build, Fabric Structure, and Modular construction projects.

EXPECTED FUTURE RESEARCH

Future military projects involving Alternative Construction Methods will continue to be input into the Knowledge Base. In addition to the Database described above, development of an expert system has been initiated. Design/Build construction is the first topic to be addressed. Others will follow. The rules-based expert system will allow the user (Project Manager or engineering personnel) to query it and will provide advisory information based on generalized knowledge and expertise, not simply specific project experience. Expert input will be sought from both non-Corps and Corps sources.

Automated System for Generating Facility Requirements

Pram Reddy[1], Annette Stumpf[2], Dr. Roger Brauer[3], and Jerry Brown[4]

INTRODUCTION

The construction of new facilities requires the development of a building program. A building program typically describes the spaces and equipment within the building as well as special or technical requirements needed by the building users. An architect usually develops the building program with the aid of the client (building owner or user). The location, climate, type of construction and building type will affect both the facility requirements and cost of construction. The building design and a cost estimate are prepared after the total square feet is estimated and costly items are identified.

The Army has established a procedure for programming, funding, designing and constructing new facilities. First, a Project Development Brochure (PDB) is developed by the future building users and District personnel. The PDB contains both the Functional Requirements and Technical Requirements. Next, a DD Form 1391 is prepared to request Congressional funding for design and construction. After the 1391 is approved by Congress, the facility is designed and built.

Installation personnel develop the Functional Requirements (part of the PDB) for buildings being programmed in the MCA-Army Building Delivery process. Because

---

[1]Operations Research Analyst, U.S. Army Construction Engineering Research Laboratory, Champaign, Illinois.

[2]Research Architect, U.S. Army Construction Engineering Research Laboratory, Champaign, Illinois.

[3]Team Leader, U.S. Army Construction Engineering Research Laboratory, Champaign, Illinois.

[4]Principal Investigator, U.S. Army Construction Engineering Research Laboratory, Champaign, Illinois.

most building users are inexperienced, building programs are often incomplete or inaccurate. This results in unforseen costs and missing information needed by the building designer.

The **Automated System for Generating Facility Requirements** was developed by researchers at USA-CERL to help facility users develop the Project Development Brochure for a new facility. This prototype tool will enable inexperienced personnel to create the list of spaces and Functional Requirements needed to estimate total square footage and develop the DD Form 1391.

NATURE OF THE PROBLEM

The Functional Requirements summary is necessary to prepare the initial DD Form 1391 and to estimate the size of the facility with reasonable accuracy. All space types needed in the facility are listed as well as the net area of space and major requirements needed to support operations and equipment. The organizational units to be housed and the number of people in each unit are established. Anticipated future changes and the impacts of these changes on the facility are discussed.

District personnel complete the development of the Technical Requirements for the facility. These Technical Requirements are often complicated and vary by climate, location and building type. Accurate descriptions of the building systems, materials, communications and automation requirements are essential to estimate construction costs for the facility. The total square feet of programmed space, the technical requirements and other facility information guides the construction of a facility to support the user's mission. How well the facility meets the user's mission depends on the accuracy and completeness of the Functional and Technical Requirements used to develop the building program.

OBJECTIVES

Our research sought to automate the process of generating Functional and Technical Requirements. We wanted to automatically estimate Functional Requirements with minimum user input, and provide the user with editing capabilities to accommodate special situations. Development of a concept for an Expert System to generate Technical Requirements was contracted to the University under a separate project.

Literature search/related solutions/deficiencies

Expert systems have been built by other researchers to interpret, diagnose, monitor, predict, instruct, plan, and design. It is important to recognize that these tasks are very different in their respective formulations, and will require different architectures for searching their solution spaces (Stefik, M. J., 1983). Many expert systems narrow the search space until the best solution to the desired goal is found. Our application seeks to help a user plan a building. Planning solutions are guided by regulations or guidelines and require the user to gather appropriate information in an organized format. This information is updated during the planning process until all necessary information is available for the building designer. An expert system for planning is inherently different than other expert systems.

METHODOLOGY

Selection of appropriate software

We investigated expert system technology to generate the facility requirements. The expert system shells we evaluated included Insight, Expert Edge, Expert Ease and Personal Consultant Plus. None of these shells were suitable for our application because we needed other capabilities in addition to the expert system.

Our domain is not narrow enough for a pure expert system application. We needed a combination of data base management, expert system and software engineering technologies. Our system requires query and edit capabilities to change the resulting requirements for special situations that could not be predicted. Financial and time constraints limited our software selection to existing expert system shells and environments available for microcomputers. We decided on a deductive system since Government rules and Army regulations had to be followed.

Guru, by Micro Data Base Systems (MBDS), an integrated software package with expert system, data base, spreadsheet, and programming language for PC applications, provided the required capabilities for our application. Guru's expert system offers both a forward and a backward chainer, reasoning with uncertainty, full interaction with a spreadsheet and database, and the ability to execute any command or program. The ability to use both forward chaining and backward chaining rules was an attractive feature of the Guru package.

System structure

The problem domain was too large to fit into one expert system with a specific goal. The domain consists of various types of facilities, spaces, occupants, and activities within the building. Sequential processing was used to create menus, give the user choices between levels of detail and to input activity, personnel and equipment data.

The solution was divided into three levels of detail, depending on the type and amount of information known by the system user. Each of these solution sets was developed as a separate system, connected by a selection menu. The three options are: (1) Approximate (2) Standard and (3) Customized. (See the Main Menu in Figure 1). Option 1 is useful when estimating the total square feet of a facility to accomodate a given number of people. Option 2 uses standard space allowances,

MAIN MENU

An approximate estimate may be needed for planning purposes prior to the functional requirements survey. Option 1 gives the approximate estimate. The second option uses standard space designs and allowances to size the individual spaces, and to get the other requirements and equipment for individual spaces. The third option gives the customized space requirements.

1. APPROXIMATE
2. STANDARD
3. CUSTOMIZED
4. EXIT

ENTER CHOICE:

Select choice number. 'EXIT' will take you out of the system.

FIGURE 1.

typical workstations and equipment for various occupations, and functional and technical requirements sorted by space type to infer the functional requirements and space requirements. Option 3 (not completed) would allow customized functional and space requirements to satisfy unique situations. Figure 2 shows a report created using the "approximate" mode for a sample facility.

The space types and required sizes are shown in a table, followed by the references used to calculate the requirements. Some of these requirements are generated by numerical calculations and data base management. Rules were used to generate some requirements when the task was more logic directed and when there is an uncertainty factor. Guru's built-in procedural language tied everything together, and made the system user friendly.

STATUS OF THE WORK

A prototype system is developed for approximate and standard modes. Further funding is needed to expand the system capabilities to include utilities, circulation, environmental conditions, design features, etc. To successfully develop an Expert system, the task should be clearly defined, rich in reasoning, and fairly narrow and domain intensive. For our application of developing functional and technical requirements, the task requires some general and common sense knowledge, some number-crunching, and some heuristic and judgmental knowledge. To solve this problem, more than expert systems technology is required. We need a combination of symbolic programming, distributed computing, powerful interface tools and programming environments. There is also a practical need to interface with CAD systems. Integrating all these technologies to solve a practical problem is a true challenge.

automated requirements

APPROXIMATE ESTIMATE, METHOD 1 (162 SF PER PERSON)

Net Building Area	31,914 SF	REF: DG 1110-3-104
Mechanical Equipment	1,596 SF	REF: DG 1110-3-104
Gross Building Area	33,510 SF	REF: DG 1110-3-104

APPROXIMATE ESTIMATE, METHOD 2 (130 SF NET PLUS)

Net Office Area	25,610 SF	REF: DOD 4270.1-M
Net Reception Area	150 SF	REF: FPMR 101-17.310-2
Net Conference Area	2,130 SF	REF: GSA March 1977
Net Washroom Area	455 SF	REF:TM 5-810-5
Net Storage Area	941 SF	REF: GSA March 1977
Mechanical, Access, Etc.	4,393 SF	REF: DOD 4270.1-M
Gross Building Area	33,679 SF	REF: DG 1110-3-104
Number of Handicapped Parking Spots	4	REF: DOD 4270.1-M
Total Vehicle Access and Parking Area	80,719 SF	REF: DG 1110-3-104 DOD 4270.1 M
Site Coverage	5.25 Acres	REF: DG 1110-3-104
Est. power demand due to computers		888 Amps
Est. heat generated by computers per hour		333 KBTUs

Press any key to continue...

FIGURE 2.

EXPECTED FUTURE RESEARCH

Research on developing automated methods to create the Functional Requirements of the Project Development Brochure will continue in Fiscal Year 1989. Work is needed to clarify which data are actually essential to the Project Development Brochure for different facility types and standard versus unique designs. Research to automate the development of Functional Requirements will integrate expert systems technology with other software engineering technologies in an integrated package similiar to the prototypical Automated System for Generating Facility Requirements.

REFERENCES

Stefik, H. J., et al. (1983), Basic Concepts for Building Expert Systems. In: Building Expert Systems (F. Hayes-Roth; D. A. Waterman; D. B. Lenat).

APPENDIX A:

NAMES AND AFFILIATION OF AUTHORS

Arciszewski, T.
Civil Engineering Department, Wayne State University, Detroit, MI.

Basten, J.
Department of Civil Engineering, University of Illinois at Urbana-Champaign, IL.

Blazek, C. F.
Institute of Gas Technology, 4201 W. 36th Street, Chicago, IL 60632.

Brauer, Roger
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Breslin, J.
Department of Civil Engineering, University of Illinois at Urbana-Champaign, IL.

Brown, Jerry
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Buchner, Brian
Department of Computer Science, University of Illinois at Urbana-Champaign, IL.

Case, Mike
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Chien, Sue
Sangamon State University, Springfield, IL.

Cohn, Louis F.
Department of Civil Engineering, University of Louisville, Louisville, KY.

Coskunoglu, Beverly
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

De La Garza, Jesus
Department of Civil Engineering, University of Illinois at Urbana-Champaign, IL.

East, E. William
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Echeverry, D.
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Ferguson, Bruce
Department of Computer Science, University of Illinois at Urbana-Champaign, IL.

Garrett, J. H., Jr.
Department of Civil Engineering, University of Illinois at Urbana-Champaign, IL.

Garrett, Ruth K.
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Gatton, Thomas M.
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Grobler, Francois
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Harandi, Mehdi T.
Department of Computer Science, University of Illinois at Urbana-Champaign, IL.

Harris, Roswell A.
Department of Civil Engineering, University of Louisville, Louisville, KY.

Hassanein, Amr
Department of Civil Engineering, University of Illinois at Urbana-Champaign, IL.

Hsui, Loretta
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Huffer, Jason
Department of Nuclear Engineering, University of Illinois at Urbana-Champaign, IL.

Ibbs, C. William, Jr.
Department of Civil Engineering, University of Illinois at Urbana-Champaign, IL.

Johnson, Peggy
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Kappes, Sandra
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Kearney, Frank W.
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Kemme, Michael
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Kim, Moonja Park
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Kim, S.
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Lange, Rense
Department of Computer Science, University of Illinois at Urbana-Champaign, IL.

Lawrence, Debbie J.
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Lawrence, Philip W.
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Melin, John W.
Department of Civil Engineering, University of Illinois at Urbana-Champaign, IL.

Metea, M.
Institute of Gas Technology, 4201 W. 36th Street, Chicago, IL 60632.

Mikucki, Walter J.
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Napier, Thomas R.
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Quek, Lee Hian
Department of Computer Science, University of Illinois at Urbana-Champaign, IL.

Reddy, Pram
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Ross, T. J.
Department of Civil Engineering, University of New Mexico, Albuquerque, NM.

Schanche, G.
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Shaw, Doris S.
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Stumpf, Annette
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Wong, F. S.
Weidlinger Associates, 4410 El Camino, Suite 110, Los Altos, CA 94022.

Yau, Nie-Jia
Department of Civil Engineering, University of Illinois at Urbana-Champaign, IL.

APPENDIX B:

NAMES AND AFFILIATION OF ASCE EXPERT SYSTEMS COMMITTEE MEMBERS

Allen, Robert
Department of Mechanical Engineering, University of Houston-University Park, Houston, TX.

Arciszewski, Tomasz
Department of Civil Engineering, Wayne State University, Detroit, MI.

Ashley, David B.
Department of Civil Engineering, University of Texas at Austin, Austin, TX.

Bowlby, William
Department of Civil and Environmental Engineering, Vanderbilt University, Nashville, TN.

Cohn, Louis F. (Chairman)*
Department of Civil Engineering, University of Louisville, Louisville, KY.

Demetsky, Michael
Department of Civil Engineering, University of Virginia, Charlottesville, VA.

Dym, Clive L.
Department of Civil Engineering, University of Massachusetts, Amherst, MA.

Finn, Gavin, A.
Stone & Webster Engineering Corp., 245 Summer Street, Boston, MA 02107.

Fricker, Jon
School of Civil Engineering, Purdue University, West Lafayette, IN.

Garrett, James H.
Department of Civil Engineering, University of Illinois at Urbana-Champaign, IL.

Gero, John
Department of Architectural Science, University of Sydney, Sydney, Australia.

Gosling, Geoffrey D.
Department of Civil Engineering, University of California-Berkeley, Berkeley, CA.

Harris, Roswell A.
Department of Civil Engineering, University of Louisville, Lousiville, KY.

*Committee offices held at the time of the conference are shown in parentheses.

Houck, Mark H.
School of Civil Engineering, Purdue University, West Lafayette, IN.

Ibbs, C. William, Jr.
Department of Civil Engineering, University of California, Berkeley, CA.

Kim, Michael
School of Architecture, University of Illinois at Urbana-Champaign, IL.

Kim, Simon
U.S. Army Construction Engineering Research Laboratory, Champaign, IL.

Law, Kincho H.
Department of Civil Engineering, Rensselaer Polytechnic Institute, Troy, NY.

Levitt, Ray
Civil Engineering Department, Stanford University, Stanford, CA 94305.

Levy, Andrew B.
BDM Corporation, 7915 Jones Branch Drive, McLean, VA 22102.

Maher, Mary Lou (Vice Chair and Secretary)
Civil Engineering Department, Carnegie-Mellon University, Pittsburgh, PA.

Mohan, D. Satish
Department of Civil Engineering, State University of New York, Buffalo, NY.

Mullarkey, Peter
Schlumberger-Doll Research, Old Quarry Road, Ridgefield, CT 06877.

Orenstein, Glen (TCCP EXCOM Contact Member)
Stone & Webster Engineering Corp., 245 Summer Street, Boston, MA 02107.

Ortolano, Leonard
Department of Civil Engineering, Stanford University, Stanford, CA.

Palmer, Richard N.
Department of Civil Engineering, University of Washington, Seattle, WA.

Rasdorf, William
Department of Civil Engineering, North Carolina State University, Raleigh, NC.

Ritchie, Stephen G.
Department of Civil Engineering, University of California, Irvine, CA.

Rooney, Martin F.
Digital Equipment Corporation, 115 Elm Street, Framingham, MA 01701.

Ross, Timothy J.
The University of New Mexico, Albuquerque, NM.

Rossman, Lewis A.
Department of Civil Engineering, U.S. Environmental Protection Agency Water Eng. Research Laboratory, Cincinnati, OH 45268.

Slater, John H.
Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, MA.

Stone, John R.
Department of Civil Engineering, North Carolina State University, Raleigh, NC.

Strzepek, Kenneth M.
Civil, Environmental and Architectural Engineering, University of Colorado, Boulder, CO.

Wilson, John
Department of Civil Engineering, LeHigh University, Bethlehem, PA.

Wong, Felix S.
Weidlinger Associates, 620 Hansen Way, Suite 100, Palo Alto, CA 93404.

USACERL DISTRIBUTION

Chief of Engineers
ATTN: CEIM-SL
ATTN: CECW
ATTN: CEEC
ATTN: CEEC-C
ATTN: CERD
ATTN: CERD-C
ATTN: CERD-M
ATTN: CERM

EHSC, ATTN: Library 22060

US Army Engineer Districts
ATTN: Library (39)

US Army Engineer Divisions
ATTN: Library (13)

8th USA, Korea (19)

USA Japan (USARJ)
ATTN: DCSEN 96343

416th Engineer Command 60623
ATTN: Facilities Engr

US Military Academy 10966
ATTN: Facilities Engineer

FORSCOM
FORSCOM Engr, ATTN: Construction

TRADOC
HQ, TRADOC, ATTN: ATEN-DEH

WESTCOM
ATTN: DEH, Ft. Shafter 96858

CECRL, ATTN: Library 03755

CEWES, ATTN: Library 39180

AFESC, Tyndall AFB, FL 32403

NCEL
ATTN: Library, Code L08A 93043

US Army Env. Hygiene Agency
ATTN: HSHB-E 21010

Engr Societies Library, NY 10017

National Bureau of Standards 20899

US Govt Print Office 20401
Receiv. Sect/Depository Copies (2)

Defense Technical Info. Ctr. 22304
ATTN: DDA (2)

96
+64
01/89