Zeitschrift: IABSE reports = Rapports AIPC = IVBH Berichte

Band: 68 (1993)

Artikel: Expert system for estimating bill of quantities

Autor: Lu, Youjie / Lu, Qian / Wang, Shouquing

DOI: https://doi.org/10.5169/seals-51854

Expert System for Estimating Bill of Quantities

Système expert de détermination des métrés

Expertensystem für die Mengenkostenermittlung

Youjie LU
Associate Professor
Tsinghua University
Beijing, China

Lu Youjie, born 1945, got his MS in structural analysis at Tsinghua Univ. MS in structural Eng. at Polytechnic Inst. of New York. He now is active in design, research and teaching.

Qian LU
Professor
Tsinghua Univertsity
Beijing, China

Lu Qian, born 1926, got his Civil Eng. degree at Tsinghua Univ. Involved in design, construction, teaching and research for 44 years. He now is a professor emeritus of Tsinghua Univ.

Shouquing WANG
Lecturer
Tsinghua University
Beijing, China

Wang Shouquing, born 1963, he got his BA in Struct. Eng. MS in constr. management at Tsinghua Univ. For 5 years, he was teaching and researched on constr. management.

SUMMARY

This paper presents an expert system for estimating civil engineering works and its autoadaptive databases. The representation of a portion of knowledge possessed by experienced estimators, the inference mechanism, autoadaptation of the databases to the changes in knowledge base and other components are discussed in sufficient detail. The proposed system has adequate explanatory facilities and hence can help novice estimators or students in the determination of the bill of quantities on an international scale.

RÉSUMÉ

Le système expert développé par les auteurs est destiné à la détermination des métrés en génie civil et à l'auto-adaptation de la banque de données correspondante. Il expose en détail l'apport de connaissances d'éminents experts en estimation des coûts, le mécanisme de déduction, l'adaptation automatique de la banque de données en fonction des modifications intervenant dans la base de connaissances, ainsi que d'autres composantes du système expert. Ce dernier peut, grâce à l'incorporation de moyens auxiliaires explicatifs, apporter une aide appréciable à des techniciens inexpérimentés dans la détermination des métrés pour des maîtres d'ouvrage à l'échelle internationale.

ZUSAMMENFASSUNG

Der Beitrag präsentiert ein von den Autoren entwickeltes Expertensystem für die Kostenermittlung von Tiefbauvorhaben und dessen sich selbsttätig erweiternde Datenbank. Die Darstellung des Teilwissens erfahrener Kostenschätzer, die Uebertragungsregeln, die Anpassung der Datenbank an Aenderungen in der Wissensbasis und andere Programmkomponenten werden angesprochen. Dank eingebauter Hilfestellungen kann das Programm Neulingen oder Studenten bei der Mengenkostenermittlung für internationale Auftraggeber behilflich sein.

1. INTRODUCTION

Chinese contractors are becoming more and more involved in overseas operations and they need computer softwares to aid them in pricing bill of quantities provided by international employers.

There are several foreign estimating softwares available on the international market. But they are not quite suitable for Chinese contractors because of their being expensive and written in English.

When the authors conceived an estimating software for the Chinese contractors three years ago, they had two options to choose, a conventional algorithmic program package and an expert system.

As a matter of fact, in China there are several dozens of computer packages in use by construction companies to estimate domestic construction projects. The estimating packages are based on the State or regional authorities stipulated norms for detailed estimates of the normal amount of labour, material, or construction equipment required per unit measure of various building or civil engineering works elements. All of the packages are custom built for local regions and difficult to adapt to other parts of China, let alone of the world, Their inflexibility is due to the steps in which human estimators estimate a construction projects having been coded into them and their users being unable to modify the steps as the case may require.

Expert systems have several advantages over the algorithmic packages because of being able to handle structured symbolic data, complete or incomplete, certain or uncertain, which represents a model of the relationships between data elements and the uses to be made of them, and to communicate with their users in natural Chinese language.

In order to make the conceived estimating system to hold appeal to Chinese contractors, the authors decided to try an expert system even though it would be risky to make such a try. The expert system built by the authors is termed ESBOP and its architecture is given in Fig.1.

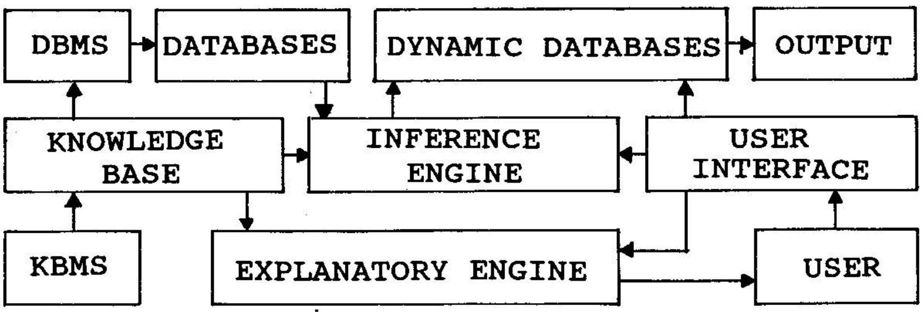


Fig.1 Architecture of ESBOP

2. KNOWLEDGE REPRESENTATION

As well known, a prerequisite for applying the expert system technology to the domain of estimating or pricing bill of quantities is the existence of human experts in estimating. An expert estimator combines his common sense, awareness of fluctuations of resources suppliers' quotations and other potential risks which will influence and affect his tender price, knowledge of the correctness and sufficiency of his tender and the rates and prices entered by him into the bill of quantities and skills of defining and manipulating formulas when estimating a works and prices bill of quantities provided by the employers.

Again as well known, it is very difficult to elicit all the estimating specific knowledge out of the experienced estimators and represent them in a form that lends itself to implementation on a computer.

However, it is equally true that a portion of the estimating knowledge body can be adequately captured by several sets of production rules.

The rules can be otherwise coded into conventional programs, which will sacrifice the flexibility achieved and deprive the user of control over the system as below described.

2.1 Estimating Formulas Used By Chinese Contractors

The example formulas the Chinese contractors use to calculate the unit rates and prices of the work items in the bill of quantities are as follows.

1. Unit rate = Labour and material cost + Overheads + Profit, or
   Unit rate = Direct cost + Indirect cost
2. Price of a work item = Unit rate x Quantity
3. Overheads = Labour and material cost x Overhead rate, or
   Overheads = Direct cost x Overhead rate, or
   Overheads = Annual overhead expenses
4. Profit = Labour and material cost x Profit rate, or
   Profit = Direct cost x Profit rate
5. Overhead rate = Annual overhead expenses
   / Annual output of a construction company

and etc.

Obviously, the formulas constitute a vital part of the knowledge body of an experienced estimator.

2.2 Representation of Formulas in Production Rules

It is a key step to represent the estimating knowledge as above mentioned in a way appropriate to implement on computer. A close look at the formulas and careful consideration of the relationships in the given portion of the knowledge body will convince the reader that they can be represented in rule form as follows:

```
Rule 1: IF Labour and material cost AND Overheads AND Profit KNOWN
        THEN Unit rate CAN BE OBTAINED
        OPERATORS wnn+n+=
        CONDITIONS used for Method 1

Rule 2: IF Direct cost AND Indirect cost KNOWN
        THEN Unit rate CAN BE OBTAINED
        OPERATORS wnn+=
        CONDITIONS used for Method 2

Rule 3: IF Unit rate AND Quantity KNOWN
        THEN Price of a work item CAN BE OBTAINED
        OPERATORS wnnx=
        CONDITIONS unconditional
```

Rule 4: IF Labour and material cost AND Overhead rate KNOWN
THEN Overheads CAN BE OBTAINED
OPERATORS wnnx=
CONDITIONS unconditional

and etc.

The OPERATORS in a rule indicates to the ESBOP inference engine what operations it should make on the known antecedent variables to derive the value of the consequent variable, while the CONDITIONS indicates under what conditions the rule shall be fired. The rules may be coded into knowledge base in an arbitrary order, not necessarily in a logical sequence.

The full set of production rules will be chained into inference network, as shown in Fig.2(a) when the inference engine reasons.

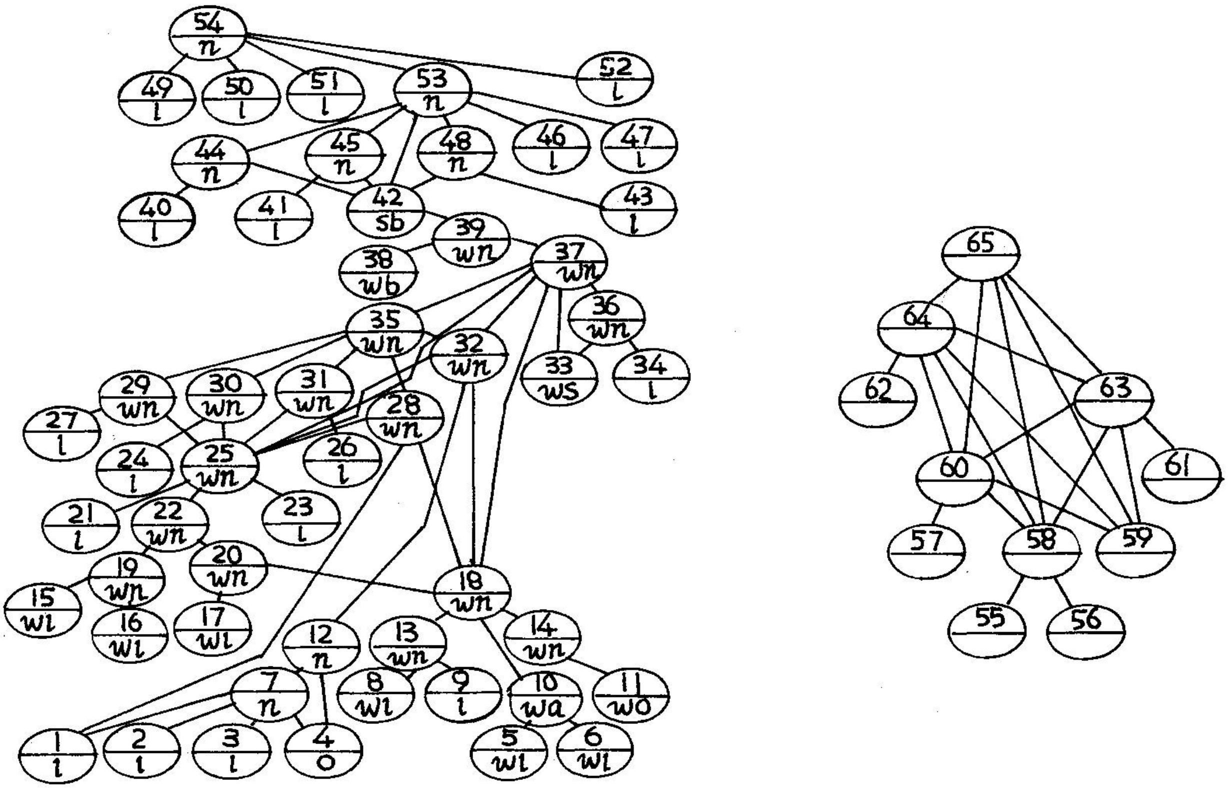
The representation of the formulas in a set of production rules rather than coding them into sequential algorithmic steps of a conventional programme makes it possible for the users, instead of the system developer, to change the estimating formulas as the case may require. For example, if the user wants to add freightage, leakage loss and purchase and storage costs to building material supplier's quotations to produce his estimates of building material costs, he can give this task to ESBOP. One thing ESBOP needs in order to perform the task is the estimating formulas. It is easy for the user to put them into the knowledge base. The formulas are represented in a set of production rules which are chained into the inference network as shown in Fig.2(b).

The inference network in Fig.2(a) represent the model of pricing bill of quantities by an experienced estimator. It is a hierarchical structure of cost or data items, linked through rule and logical arcs, as depicted in the figure. The oval circled cost or data items which can be identified in the pricing process are referred to as nodes. The lines or arcs represent the dependency of one cost item upon the other cost or data items.

Any node in the network together with the nodes, if any, linked to it through arcs from below represent a rule. This node in question represents the consequent of the rule and the nodes linked to it from below represent the antecedents of the rule. For example, the node 12, node 4 and node 7 represent the rule for calculating "Overhead rate". In this case, "Overhead rate" plays the role of the consequent, while "Annual overhead expenses" and "Annual output of a construction company" play the role of antecedents, because the value of "Overhead rate" can not be derived until those of "Annual overhead expenses" and "Annual output of a construction company" are evaluated.

It can be seen from Fig.2(a) that the nodes can be classified into three categories. The top level node in the inference network represents the Tender Price and therefore is the model's goal node, which terminates but does not originate arcs. The terminal or "leaf" nodes correspond to the data asked of the user or retrieved from the databases and therefore are referred to as data nodes which only originate arcs. The intermediate nodes are linked to each other through arcs from both above and below. They represent well-defined cost items forming the Tender Price and therefore are referred to as subgoal nodes.

The rules as shown in Fig.2 are used by the Chinese contractors to estimate building construction works. If there is a need of estimating installation work or other civil engineering works such as roads, bridges and tunnels the user is provided with a facility for coding their formulas applicable to the same into the knowledge base and then ESBOP will manipulate them as intelligently as a human estimator.



(a) (b)

1. Profit rate
2. Employee productivity
3. Number of employees
4. Annual overhead expenses
5. Supplier's quotations of materials
6. Material consumption per unit of elementary work
7. Annual output of a construction company
8. Workhours per unit of elementary work
9. Worker daily payment
10. Material cost
11. Minor material percent
12. Overhead rate
13. Labour cost
14. Minor material cost
18. Labour & material cost
32. Overheads
54. Tender Price
55. Hauling distance
56. Freight rate
57. Leakage loss percent
59. Supplier's quotations of materials
61. Storage waste percent
62. Purchasing cost percent
65. Estimate of material cost

(some items omitted)

(a) Pricing Bill of Quantities
(b) Adding freightage, leakage loss and etc. to supplier's quotations

Fig.2 Inference Networks

3. INFERENCE

The ESBOP employs two forms of inference in its reasoning, that is, backward and mixed chaining. There is a user interface in ESBOP for switching the inference between the two forms of chaining in response to the user's request or to the current status of the system.

3.1 Backward Chaining (Goal driven reasoning)

Backward chaining proceeds in two stages, the first stage chains rules together to form an inference network or subnetwork. The second evaluates the antecedents and then the consequent, rule by rule.

ESBOP begins backward chaining by offering to the user menu options which are actually the nodes in the inference network depicted in Fig.2. The user chooses an option, or sets a subgoal or goal. In response to the users option, the inference engine searches the knowledge base and chains all the rules needed to reach the subgoal or goal. In this stage, all the activated rules are examined to ensure that none of them makes more than one appearance in the set of active rules. The activated rules form a complete subset of the knowledge base, which means that the inference subnetwork or network formed of the chained rules is not broken. It can be verified that in this stage the backward chaining uses the depth-first search strategy.

Backward chaining enters the second stage thereupon to evaluate the nodes in the inference subnetwork. This is a bottom to top process, or in a forward fashion because the antecedent nodes are evaluated before the consequent node. The data nodes are assigned the values retrieved from the databases or asked of the user. The values are put into a dynamically allocated memory area referred to as dynamic database for subsequent evaluation of the intermediate or subgoal nodes. When evaluating an intermediate node, the inference engine searches the dynamic database for the values of the antecedents of the rule of which the intermediate node is the consequent. The evaluating process continues until the user chosen node is evaluated. Once the goal or subgoal set by the user is reached, the inference engine comes back to the interfacing menu and returns the control to the user.

The backward chaining as above explained can be illustrated using the case of "Overhead rate". If the user chooses "Overhead rate" at the interfacing menu, the backward chaining will proceed in the two stages as shown in Fig.3. The arrows and the circled arabic numbers indicates the directions and the sequential order of chaining the rules and evaluating the nodes. The chaining and evaluating indicated by the dashed arrow will be avoided by the inference engine.

In this case, Rule 12,7,1,2,3 and 4 are chained in that order. Then the inference engine reverses the order as follows.

Rule 1: IF database1 KNOWN THEN Profit rate CAN BE OBTAINED OPERATORS 1 CONDITIONS unconditional

Rule 2: IF database1 KNOWN THEN Employee productivity CAN BE OBTAINED OPERATORS 1 CONDITIONS unconditional

Rule 3: IF database1 KNOWN THEN Number of employees CAN BE OBTAINED OPERATORS 1 CONDITIONS unconditional

Rule 4: IF database2 KNOWN THEN Annual overhead expenses CAN BE OBTAINED OPERATORS o CONDITIONS unconditional

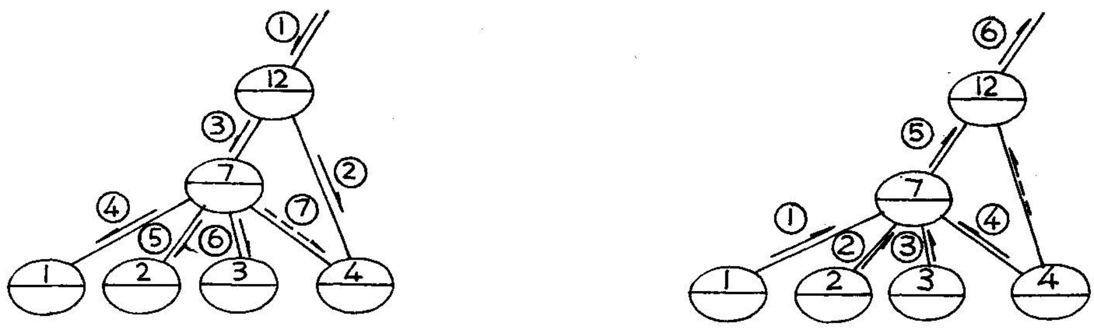
Rule 7: IF Employee productivity AND Number of employees AND Profit rate AND Annual overhead expenses KNOWN THEN Annual output of a construction company CAN BE OBTAINED
OPERATORS nn*n1+/n-= CONDITIONS unconditional

Rule 12: IF Annual overhead expenses AND Annual output of a construction company KNOWN THEN Overhead rate CAN BE OBTAINED
OPERATORS nn/= CONDITIONS unconditional

Having activated the rules, following the '1' in the OPERATORS, the inference engine searches database1 for the values of the consequents of Rule 1 - 3 and retrieves, for example, 0.15, $24,000 and 750 persons, while for that of Rule 4's consequent, it , summing all the overhead expenses in database2 according to the 'o' in the OPERATORS, finds $2,119,230. When it is Rule 7's turn, the engine, making the math as the 'nn*n1+/n-=' indicates, evaluates the consequent as 24,000 x 750/(1+0.15) - 2,119,023 = $13,533,150.91. "Overhead rate", the consequent of Rule 12, is then evaluated to be 2,119,023/13,533,150.91=0.1566 as the 'nn/=' tells.

3.2 Mixed Chaining (Data driven chaining)

When the user regains hold of control at the interfacing menu, he can choose to set another subgoal or goal. If the inference network or subnetwork to be formed by the inference engine in order to reach the newly set subgoal or goal does not contain the nodes which have been evaluated and the values of which have been put into the dynamic database, then another backward chaining will take place. However, if the inference network or subnetwork does contain the evaluated nodes, then the inference will be in a mixed fashion, or a combination of forward and backward chaining. In order to reach the user's new subgoal or goal, the inference engine will search the dynamic database for the last entered item and then pick out of the knowledge base the rule the antecedents of which contain the last item in the dynamic database. Then the inference engine attempts to evaluate the consequent node. If the last item is the only antecedent of the activated rule, then the attempt will succeed. Otherwise, the antecedent(s) of the rule other than the last item in the dynamic database will act as trigger(s) making the inference engine to do backward chaining(s) so that they are evaluated and the attempt to evaluate the consequent node can succeed. When the consequent node has been evaluated, its value will be added to the dynamic database and become the new last item in the same.



(a) Activating rules (b) Evaluating nodes

Fig.3 Backward Chaining

The reason for the inference engine to combine forward and backward chaining instead of using simply the backward chaining to reach the new subgoal or goal is to save reevaluating the previously evaluated nodes, and hence economize the use of the dynamic database.

If the consequent node happens to be the new subgoal or goal, then the mixed chaining will be ended. Otherwise, the evaluated node will trigger a new round mixed chaining in a way similar to the aforesaid until the new subgoal or goal is reached.

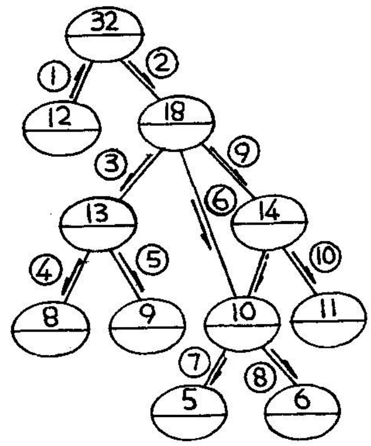


Fig.4 Mixed Chaining

Fig.4 gives an example of the mixed chaining. In this example, the user, after "Overhead rate" has been evaluated, chooses "Overheads" option at the interfacing menu as his new subgoal.

4. EXPLANATORY FACILITIES

ESBOP is provided with two explanatory features. It answers "How" questions from the user. When the user asks "How is "Overhead rate" evaluated?", the system answers by displaying all the rules needed to evaluate "Overhead rate" in a backward chain on the monitor's screen and thus the user will be satisfied by tracing the sequence of rules. "Why" questions are interpreted as "Why is a cost or data item needed to evaluate the subgoal or goal?" The answer is given by displaying the rule chain in a forward fashion which is originated by the node representing the cost or data item in question and top ended by the subgoal or goal node.

The explanatory facilities makes the system very friendly and transparent to the user and can help students in learning estimating and pricing.

5. ACTIVE RULES STACK AND DYNAMIC DATABASE

The rules activated in the first stage of backward chaining are sorted in such a manner as to enable the antecedents of every rule to be evaluated before the consequent and then the pointers to the sorted rules are pushed into a stack named "active". The evaluating begins from the stack top. The pointer to the stack moves down as the evaluating proceeds. When the pointer is down to the stack bottom, the user's subgoal or goal is reached.

All the data and values the inference engine has sought from the databases or derived during chaining are added to the dynamic database which is managed dynamically during the reasoning. The dynamic database does not come into existence until the inference is put into operation and grows in size as the inference goes on. At the same time it keeps track of the reasoning history. The last entered item represents the current status of the system indicating to the inference engine where it arrives when it runs along the inference subnetwork in a backward or forward fashion.

6. AUTOADAPTIVE DATABASES

ESBOP may be different from the expert systems for the domains other than estimating and pricing in that it manipulates much more numeric data than the other systems. The unique characteristic requires the provision of a powerful database management. The DBMS of ESBOP manages building materials, the norms for detailed estimating, bill of quantities and various charge rates.

The serious problem facing the DBMS is the variability of the data structure of the databases. As claimed above in the paper, the system can be adapted by the user to the tasks other than estimating building construction projects, out of which arises necessity of the DBMS's adapting itself to the specific task data structures.

Certainly, there are several approaches to this problem. The common method is to provide a feature (as with dBASE III or IV) by which the user defines the data structures for their particular tasks using CREATE or MODIFY STRUCTURE commands. However, the user will feel it an additional burden to do this. Having studied the inference networks given in Fig.2, the authors find a cue to propose an alternative approach to this problem.

6.1 Dependency of Data Structure of Databases on Data Nodes of Inference Networks

For convenience and clarity of exposition, let us consider the inference network as shown in Fig.2(b). The goal node is the estimate of building materials and the six data nodes represent the building material suppliers quotations, hauling distance, freight rate, leakage, storage waste and purchasing cost, respectively. On the other hand, each record in the building material database shall at least have six fields which are nothing else but the six data nodes in the inference network. It is this coincidence that reveals the dependency of data structure of databases on the rules coded into the knowledge base and leads the authors to a solution to making the databases respond automatically to the changes in the knowledge base.

6.2 A Dictionary Supplementing Knowledge Base

It is certainly true that the records in the building material database shall have another three fields for the purpose of identifying individual materials, namely, description, specification and measure unit. The three fields are the basic attributes of building materials, but do not appear in the inference network and the knowledge base. A dictionary added to the DBMS makes good the insufficiency of the knowledge base. The words and phrases representing the attributes of the data nodes in the inference network are elicited out of the user by the inference engine built in the DBMS and put into the dictionary.

The inference engine of the DBMS can adapt the data format of its database to the tasks implied in the knowledge base without need of the user making changes to them, which explains why the databases are referred to as "Autoadaptive Databases".

When it comes to a dictionary, there is always a fear that it may not be able to cover all possibilities. The sufficiency of the dictionary is guaranteed by the DBMS's capability of picking out the new words or phrases which appear in the new rules coded by the user but have never made appearance in the dictionary before. If the DBMS does make such a discovery, it will prompt the user to put all the synonyms and attributes of the new data nodes into the dictionary.

Having incorporated the dictionary and the inference engine, the DBMS has become intelligent and much more convenient to the user. The enhanced DBMS is shown in Fig.5.

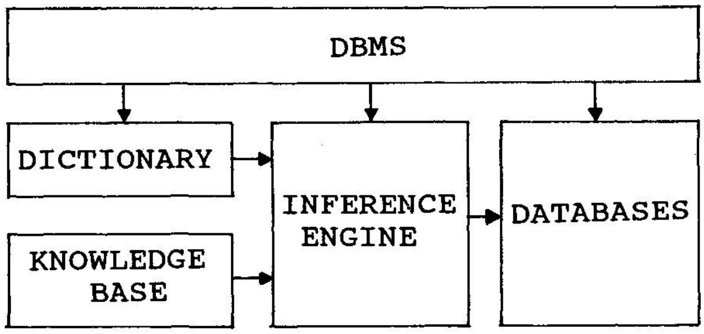


Fig.5 Enhanced Database Management System

7. IMPLEMENTATION DETAILS

ESBOP is coded in Microsoft C. C is both a symbol and a numeric processing language and best meets the need of building expert systems. Its recursive function makes it easier to code the inference engine than several other programming languages such as BASIC and FORTRAN. The good portability of C also makes it popular with the world wide programmers. ESBOP is run under DOS and on GW286, GW386 and other compatible microcomputers with Chinese character system.

REFERENCES

1. P.H. WINSTON, Artificial Intelligence, 2nd Edition. Addison-Wesley Publishing Company, 1984.

2. L. JOHNSON, E.T. KERAVNOV, Expert Systems Technology - A Guide. Abacus Press, 1985.

3. Y.J. LU, S.Q. WANG, Q. LU, Expert System ESBOP for Bidding of Overseas Construction Projects. Proceedings of 1989 Annual Conference on Intelligent Decision Support Systems for Civil Engineering Works sponsored by the National Natural Science Foundation of China, Electronic Industry Press, Beijing, 1990.

4. Y.J. LU, L.M. SUN, Q. LU, Generation of Database Management Systems by Using Production Rules. Journal of Tsinghua University, Vol.31, No.5, 1991.