Sci. J. Fac. Sci. Menoufia Univ. Vol. IX (1995). 315 - 334

PROCESSOR ASSIGNMENT:
EXPERT DATABASE SYSTEM

By

Hany M. Harb

Computers and Systems Dept. Faculty of Engineering

Al - Azhar University

ABSTRACT

This paper discusses a processor assignment expert database system. The processor stands for any server like an employee in personnel, physical processor (cpu) in task scheduling. In other words, this generalized model has a wide variety of applications. This system is an integration of a processor database and an expert system to select candidates for a certain function. Attached with each candidate, there is a score expressing its priority to have the function. The score is calculated from the function attributes weights. Each attribute has an attached weight and these weights may be variant for the same function and from function to another. There are two interface units associated with the expert system, one to the user and the other to the database. The user has still a direct access to the database. A processor assignment expert database system application has been built and experimented.

Keywords : Database, Expert Systems, Processor Assignment.

INTRODUCTION

The combination of the expert system and database system called Expert - Database system (EDS). Exper system and AI technology in general contribute to the database system in areas such as providing a useful reasoning ability in query optimization tasks. DBMS technology contributes to the expert systems in giving them the ability to access large collection of facts and also to apply features such as concurrency control, data security and optimized access to knowledge base. There are different ways of integration between the database systems, and the expert systems. Different ways of integration lead to different types of EDS such as an enhanced database system, an enhanced expert system, interdependent expert system and database as depicted in Figure 1 [4].

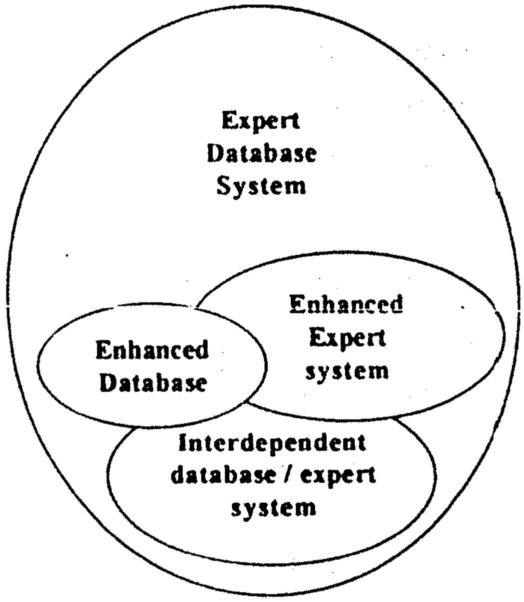

Figure (1.a) : EDS Different Types

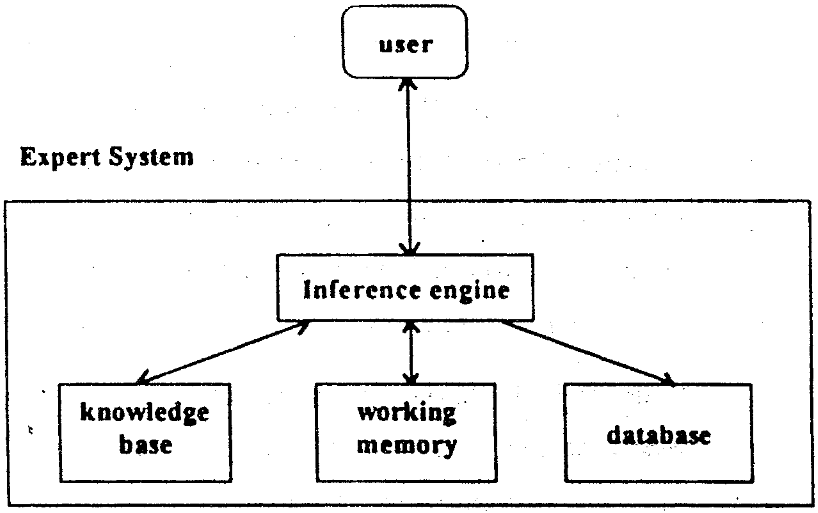

Figure (1.b) : Database and Expert systems integration Architecture

The techniques and tools needed to build the expert database systems come from AI, database management and logic programming aspects [1,2].

The expert database system can improve performance by providing intelligent answers using database semantic integrity constrains for query optimization and combining knowledge and data given search techniques in efficient inference schemes.

The expert database seeks techniques and tools that make databases "active" reasoning agents and that allow database systems to support AI applications managing and accessing large knowledge and databases.

The architecture of EDS is shown in figure (1. b). Different kinds of system components exist between the users and the information. The components are knowledge base for knowledge handling, database for shared information management, and inference processors for linking between database and knowledge base. The number and type of inference processors will depend on the application.

From the architecture of EDS, we can say that an EDS is essentially the composition of two powerful search engines. One engine is searching knowledge rules to solve an application problem and, in so doing, generating queries over shared information. The other engine is searching the shared information to answer the queries.

Here is another way to classify the integration ways of EDS which is based on the cooperation of the two engines. These ways or approaches are deductive database, heterogeneous approaches.

Deductive database approach (or the homogeneous approach) integrates data manipulation functions and deductive functions into a single system. This approach does not provide an intrinsic capability to distinguish data and knowledge. A typical framework is represented by logic programming [2, 3, 5]. The heterogeneous approach refers to the cooperation of the expert system and the database management system [2, 11]. In this approach the key issue is the interface that allows the two systems to communicate. There are two kinds of coupling in designing the interface: loosely or tightly coupling [2].

The loosely coupled approach tends to maintain for each component its identity: the ES essentially devoted to deductive functions, the DBMS manages the database. It may be considered as enhanced database or enhanced expert systems. The DBMS acts as a server to the ES, supplying on demand the data that ES requirs. This approach has been also referred to as the compiled approach [6], since it is based on two distinct phases. The first phase is a computation on the side of the ES, which, using its knowledge, generates the queries for the DBMS, then the second phase commences as the execution of the queries on the side of the DBMS and the delivery of the result to the former.

One of the major advantages of the approach is represented by the possibility of using existing databases, to which ES can be connected as one of the application programs. In this way the database continues to serve all its users without the necessity of any data replication. The principle drawback of this approach is represented by the necessity of separating in a precise way the deductive phase and the data retrieval phase. Also, a problem of consistency may arise if the data collection extracted from database is used while the original version is updated.

In a tightly coupled approach the interactions between ES and DB can take place at any level, the data of the database represents indeed a natural extension of the knowledge encompassed by the ES. In this approach, the main problems seen in the loosely coupled approach are avoided. The problem of the tightly coupled approach is a server slowdown for the ES operations [7, 8, 9].

This paper discusses an expert database called the processor assignment expert database. This system is based on ES -DBMS loosely coupled approach. The system is a combination of a processor database and an expert system to select the most efficient processor candidates for a certain function.

THE SYSTEM MODEL

As already mentioned, one of database and expert systems (ES) integration approaches is to build an intelligent layer (expert system) on the database system as a top layer (ES-DBMS Loosely coupled approach). This is a natural way if the database has already been implemented as in our case and we want to extend its capabilities by writing compatible expert system making use of the stored data. The user has still a direct access to the database and more than one ES can be implemented above the databse. A database dependent ES shell is provided as an expert system layer developing tool.

1) The Model organization :

A processor databse is available and an expert system to select a set of processors for a certain function is to be built as a database intelligent application program. Each candidate has a score expressing its eligibility to have the function. It may be more than one candidate for a certain function. Figure 3 shows the detailed structure of the model, where there are two interface units asociated with the system, one to the user (UI) and the other to the database (Front End Processor). The user can interact with the database either directly of through the ES layers. The ES has a reasoning capability to give an explanation how certain processor gets its score.

2) The database system :

The processor database can be presented as two entities : The processor entity and the capability entity. Processor entity identifies the unique processor attributes such as processor ID, processor title where there is only one occurrence of each attribute for each processor. The capability entity identifies a certain capability for a processor. There may be more than one capability for each processor and there may be one occurrence of a capability for a processor. In other words, there is one to many relationship between the processor acapability entities as in figure 2.

Processor

Processor ID	Processor one-to-one attribute values

Capability

Capability ID	Validity	Occurrence #	One-to-many attributes

Figure 2 : Processor and its Capability relationships

3) ES shell :

It is not necessary for ES to work on the schema (whole database) but it can work on subschema which is defined in the knowledge base. The subschema which is mainly supported for security issues, should be verified by the databse engine.

The shell is mainly composed of two parts : schema editor, and rule editor. Through schema editor, the ES developer can define the database subschema on which the ES works in terms of SQL - like statements.

select attribute (s)
from relation (s)
[where predicate]
[grouping by attribute (s) [having predicate]]

Through the rule editor, the user can insert, delete, modigy a rule. A rule has the following format

if condition
    [and condition] .....
then
    conclusion

4) Front End Processor (FEP) :

This unit prepares SQL statements to be submitted to DBMS engine as shown in figure (3). It is composed of construct, prepare, and iterate modules. The construct module takes input from a rule coming from the inference engine and creates a boolean expression that can be used in the where - clause of a select statement. The resultant attributes (fields) are derived from the rules and the relations to which the attributes belong are derived or deduced from the subschema on which ES works. The prepare module is intelligent enough to resolve any conflict resulting if the attribute name belongs to more than one table. The SQL statement results from the concatenation of the select part derived by the prepare module and the where - clause that constructed by the construct module. The iterate module forms the loop for the previously prepared select statement as it follows.

for each pointer into output - array

end foreach

The output - array is filled with the records resulting from the loop execution. There is also a boolean variable (existent variable) which is automatically assigned a TRUE if the output - array gains any output from the last previously executed SQL loop, otherwise FALSE is assigned. The existent variable may be checked by the inference engine. FEP acts as the interface unit between the inference engine and the database subschema, and it is responsible to transfer any selective stored database item mentioned in any rule to the available database query language retrieval statement. This statement is submitted to DBMS to retrieve the required data item value which in turn submitted back to the inference engine.

IMPLEMENTATION

The expert system is implemented as an inttelligent layer above the database and it is allowed for the ES to call the database through FEP and not vice versa. The ES is built as production system (rule based expert system).

The rules work on two data sources : the processor database and submitted function attributes.

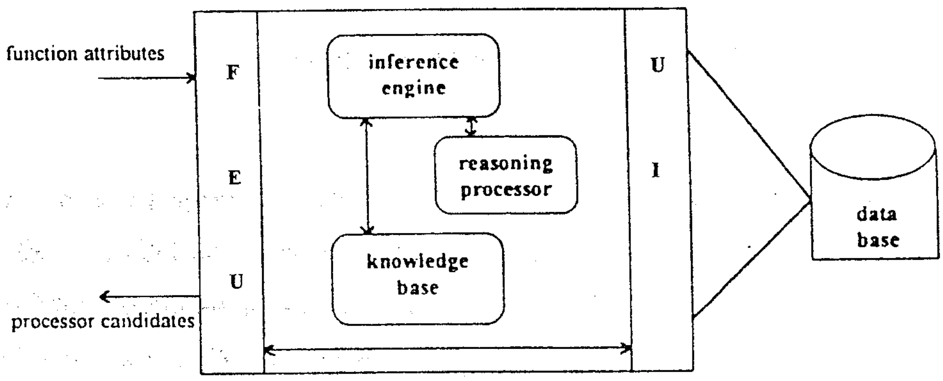

Figure 3 : Expert Database Processor Assignment model Architecure

1) The Function Attributes :

A function has different attributes according to which the best suitable processor (or set of processor) is selected. A function attribute may have boolean value (a must attribute) or integer value with defined minimum and maximum (range attribute). This function specification may be stored for later use and so, it is identified by code and specification ID to differentiate different specifications of the same function or of different functions.

Function Specification

Function Code	Title	Specification ID	Attribute 1	.....	Attribute n

A processor is rejected if it does not satisfy a must attribute. Some of range attributes have a higher weight than others, so each range attribute has a weight and the attribute weight may be variant for the same function and from function to another. AS a result of assignment, each processor has a score depending on how much it satisfies the attributes of the function. The system user may consult how this score is gained (reasoning capability).

In function specification, a range attribute may have an associated numeric value n, an assigned processor is given a score depending on how must it satisfies this associated value n. If this value is positive (+n), it means greater than n, negative value means less than n, and n (without sign) means exact n. For example, in task scheduling problem, if task memory requirements is + 120 kb, then an assigned cpu must have an available memory greater than this requirement. The greater processor memory is available, the greater score is given to this processor.

2) Inference Engine :

As already mentioned, a function has a set of attributes, either must or range attributes. This must attribute either exist or does not exist in the processor. If a processor does not have the capability to satisfy this function attribute, then this processor is discarded. For range attributes (not must attributes), non - discarded processors are checked to see how much each processor satisfies each of the range attributes. This satisfaction is expressed as a satisfaction score added to the processor total weight (total score). After checking all the function attributes, the list of candidate processors are sorted in descending order according to their weights. The first processor (maximum weight processor) is the most eligible processor to execute the function.

```
if a processor does not satisfy any must attribute
then discarded the processor

for each range attribute
do
    check how much processor - attribute satisfaction
    weight = weight + satisfaction score
done
```

The engine is augmented by the data dictionary expressing for each function attribute, its corresponding column in the database relations, its minimum and maximum values if exist. There also exists a synonym dictionary describing related terms. A dictionary sample is presented in the following example.

The system asks the user for the function code and specification ID, if the function is already stored, the attributes and their values are displayed. The user has the ability to change the attributes and / or their values. The modified values are stored. If the function is not already stored the system starts displaying each attribute and the user has the ability to reject the attribute, or enter its value. The system saves the function attributes, applies the rules seeking the candidates. The user may enter a limit for the number of the required candidates.

EXAMPLE

The model is applied as the researcher - project database where the processor stands for the research and the project stands for the function. The researcher database includes capability entities such as research history, travel, vacations, training, punishment, languages, financial resources, and annual evaluation.

The researcher database is assumed to be found before ES estabilshment. The researcher attributes are emp - number, emp - name, birth - date, hire - date, current - position, specialization degree, social - state, military - state, address, religion, phone, dept - code, and current salary. The research relations are listed as it follows.

qualification history (degree - code, major, graduation - date, school).
research history (project - code, title, starting - date, ending - date, degree).
vacation (vacation - code, starting - date,period).
travel (starting - date, period, country, purpose).
training (train - number, title, date, period, evaluation).

punishment (punishment - code, date, reason).
language (language, reading - rate, writing - rate).
financial resource (resource, annual - income).
annual evaluation reports (year, rate, comments).

A list of project attributes is listed in table 1. A project attribute value n may be positive (+n) which means greater than n, negative (-n) which means less than n, or without sign which means exact n.

Toble 1: Probect attributes

Probect Attribute	Value	Type
Project code		Alphanumeric
project title		Alphanumeric
qualification	n	Integer
current degree	n	Integer
age	n	Integer
sex		Boolean
religion		Integer
experience years	n	Integer
specialization		Alphanumeric
military exemption		Boolean
social status		Integer
languages	n	Integer
attitude attributes		
other attributes		
other attributes		
other attributes (as have a car)		

Data dictionary is presented in table2.

Toble 2: Data dictionary

Probect Attribute	relation	field	Min.	Max
language	language	code		
ability - to - read	language	reading rate	-10	+10
ability- to - write	language	writing rate	-10	+10
sex	emp	sex	-100	0
qualification	qualification	degree code		
major	qualification	major	-50	+50
experience years	project - history	starting date ending date	30	+

Toble 1: Probect attributes

specialization	Related Fields
Computer	Computer engineering, computer science, software engineering, AI, ES,..

Amale engineer is required for system adminidtration.Aminimum of 10 years experience is required. It is preferably for candidates to have master of science in computer engineering or computer science field. Fluency in English is preferable.

The system consults attribute by atrribute. We give a score range from-100 (least preferable) to 100 (most preferable).

A sample of inference rules is listed as it follows.

For all researcher
weight = 0

if sex is female
then weight - = 100
/* decrease the weight by 100 * i.e. weight = weight - 100
or the processor is discarded */

if qualification = +2 and major = "computer" then weight + = 50
/* if qualification degree greater than B.Sc. in computer as a major (master or Ph. D. degree), then increase the weight by 50.

Note the qualification is coded as 2 B. Sc., 3 MSc, 4 Ph. D.
*/

if qualification = 2 and major "computer" then weight + = 30

if qualification = -2 and major ◊ "computer" then weight - = 50

if years - of experience = +10 and special = "system administration" then weight + = 30

/* if the researcher has years of experience greater than 10 years with specialization in system administration, then increase the weight by 30 */

if year - of - experience = +5 and special = " system adminstration"
then weight + = 20
if years - of - experience = -5 and special = "system adminstration"
then weight - = 20
if language = " english" and ability - to - read = 5 and ability - to - write = 5
then weight + = 10
if language = " english" and ability - to - read = 5 and ability - to - write = -5
then weight + = 7
if language = " english" and ability - to - read = -5 and ability - to - write = 5
then weight + = 7
if language = " english" and ability - to - read = -5 and ability - to - write = -5 then weight - = 7

The user does not have to memorize any variable names since for any selected attribute the system displays its associated variable names and asks the user to enter the corresponding values and weight. The correspondence between the variable names and attribute names in the relational database is reserved in the data dictionary.

The system can be trerminated if specific rule is satisfied, for example, the following rule can be used to terminate the system.

if number of candidates = 10 and their scores > = 95%
then termination condition.

CONCLUSIONS

In this paper we discussed features of an expert database system called a processor assignment expert database. Thissystem has been built based on ES - DBMS loosely coupled approach. This means that an intelligent layer (expert system) has been built on the database system as a top layer. The personnel expert database system used to select candidates for a certain job.

The database system has been found before the expert system establishment. The expert system shell has been built. It is mainly composed of two parts : schema editor, and male editor.

Through schema editor, the ES developer can difine the database subschema on which the ES works on. Through rule editor the user can insert, delete, and modify a rule. The database interface unit has also been built. It is responsible to transfer any selective stored database item mentioned in any rule to the available database query language retrieval statement. This statment is submitted to DBMS to retrieve the required data item value which in turn submitted back to rule. The personnel expert database system has reasoning capability to give an explanation how certain candidate gets it score.

REFERENCES

1- J.M. Smith "expert database system : A Database Perspective", The Benjamin Cammings 1986.

2- M. Missikoff, and G. Wiederhold "towards Unified Approach for expert system and database system" 1986.

3- G. Werderhold : "Knowledge and database Management", IEEE Software, Jan. 1984.

4- P. Beynon - Davies : "Expert Database System", 1991.

5- J.W. Lloyd : "An Introduction to Deductive Database system"; The Australian Comp., May 1983.

6- R. Kowalski : "Logic for Problem Solving"; North Holland, 1979.

7- U.S. Chatravarthy, J. Minker, and C. Tran : "Interfacing Predicate Logic Lnaguages and Relational Database"; Proc. of First Int. Logic Programming Conference, Maraseille, 1982.

8- D.K. Hsiao (Ed) : "Advanced Database Machine Architecture"; Prentice Hall, 1983.

9- D. Kogan : "The Manager's Assistant, An Application of Knowledge Management"; IEEE Data Engineering Conference, Los Angeles, Apr. 1984.

10- H.O. Leilich and M. Issikoff (Eds) : "Database Machines"; Springer Verlag, 1983.

11- L.M. Delcambre : "RPL : An Expert System Language With Query Power", IEEE Expert, 1988.

12- Y.Y. Leung and D.L. Lee : "Logic Approaches for Deductive Database", IEEE Expert, 1989.

نظام قواعد بيانات خبير

هانى حرب - قسم الكمبيوتر والأنظمة - كلية الهندسة - جامعة الأزهر .

هذا البحث يقدم نظام قواعد بيانات خبير يفترض وجود قاعدة البيانات مسبقاً ونظام خبرة بينى كتطبيق ذكى فوق هذه القاعدة من البيانات. قاعدة البيانات تتضمن بيانات المشغل ونظام الخبرة يقوم بإختبار أنسب مجموعة من المشغلات لاداء وظيفة معينه. كل وظيفة لها مجموعة من المتطلبات وكل متطلب له وزن. ومدى تحقيق مشغل لمتطلبات وظيفة معينه يقيم برقم وأنسب مشغل هو المشغل الذى يحصل على أعلى رقم مع ملاحظة أن بعض هذه المتطلبات ضرورى ولازم التحقيق بمعنى أن المشغل الذى لا يستطيع تحقيق هذه النوعية من المتطلبات يستبعد من الترشيح وقد تم تطبيق هذا النموذج كقاعدة بيانات للباحثين المناسبين لمشروع بحث معين.