



Journal of Applied Sciences

ISSN 1812-5654

science
alert

ANSI*net*
an open access publisher
<http://ansinet.com>

A Hybrid Scheme for Knowledge Representation

Hussein H. Owaied, Mahmoud Malek Abu-A'ra and Monzer Moh'd Qasem
Faculty of Information Technology, Middle East University,
P.O. Box 42, Amman 11610, Jordan

Abstract: This study presents new methodology for knowledge representation called hybrid scheme for knowledge representation. The scheme is based on the integration of two different knowledge representation formats. The integration consists of the Rule-based and the Case-based formats using the Blackboard. This scheme uses both procedural and declarative knowledge representation formalisms through the application of relational database. So the rule base and case base formats have been converted into tables. The convert rule to table assigns the first attribute of table to be a conclusion of a rule and the reset of attributes as the conditions of a rule. Also assign the first attribute of table to be a cause of a case and the reset of attributes as the assertions of a case. In this study, all the algorithms, for creating, indexing and checking the availability of a rule and a case, are present. The scheme facilitates combination of forward and backward chaining reasoning, using many problems solving methodologies and different searching techniques. Therefore, the proposed scheme facilitates the common sense, deduction and analogical reasoning activities in the inference engine, because rule base provides the deduction, case base provides the analogical reasoning and the blackboard provides the common sense. This scheme will be used for developing shell expert system as new environment development for expert systems. The scheme makes the proposed Rule-Case-based shell expert system more flexible, efficient and more powerful for the development of the expert systems in future.

Key words: Artificial intelligence, computer science, expert systems, knowledge engineering, information technology

INTRODUCTION

The integration of (two or more) different knowledge representation methods is a very active research area in Artificial Intelligence. The aim is creates hybrid formalisms benefiting from each of their components. It is generally believed that complex problems can be easier solved with hybrid systems. The effectiveness of the various hybrid approaches has been demonstrated in a number of application areas (Fahmy and Douligeris, 1999). In most of the hybrid approaches, two knowledge representation methods are being integrated. This is due to the fact that the integration of three or more knowledge representation methods is more complicated. One of the most popular types of integration involves the combination of rule-based with case-based reasoning approaches (Aha and Daniels, 1998; Vossos *et al.*, 1991; Prentzas and Hatzilygeroudis, 2002). The efforts to combine symbolic rules and cases have yielded advanced knowledge representation formalisms. The effectiveness of those approaches stems from the fact that rules and cases are alternatives in representing application domains and solving problems (Aamodt and Plaza, 1994;

Sabater *et al.*, 1998; Chi and Kiang, 1991). Rules represent general knowledge of the domain, whereas cases specific knowledge.

Rule-based systems solve problems from scratch, while case-based systems use previously stored situations to deal with similar new instances, therefore, the integration of both approaches turns out to be natural and useful (Chan *et al.*, 2000). There are many implementations of expert systems using various tools and various hardware platforms, from powerful LISP machine workstations to smaller personal computers. Many expert systems are built with products called shell expert systems (Brown, 1995). Expert systems are computer-based software applications which embody some non-algorithmic expertise for solving certain types of problems, such as in the domain of e-learning (Tlili-Guiassa and Tayeb, 2006) and using Artificial Neural Network (Hewahi, 2010).

Also the expert system can be defined as specific type of knowledge based system with the facilities of correctly deduct and making decision, in other words the knowledge based system that can answer the two questions how and why. There continues to be a debate

as to whether it is best to explore the technology and experiment for write expert systems or using shell expert systems (Salim *et al.*, 2002). The shell is a piece of software which contains the user interface, a format for representation of the knowledge base in narrow and specifics domains and an inference engine. The knowledge engineer uses the shell to build an expert system for a particular problem domain (Abu-Naser *et al.*, 2008, 2010). One of the major bottlenecks in building expert systems is the knowledge engineering process. The coding of the expertise into the previously chosen format, such as rule base, frame, semantics nets, case-base, or others, can be a difficult and tedious task.

THE PROPOSED KNOWLEDGE BASE SCHEME

The knowledge base presents the repository of knowledge for specific and narrow domain for the knowledge based system. So, the most important part of knowledge based system is the knowledge base and the power of any knowledge based system and expert system inherently in the adequate and integration of knowledge representation forms used for the particular domain. In this sense, the most important phase, in building knowledge based system and the expert systems, is the building of the knowledge base; this process is part of knowledge engineering which is an important field at present century. Usually, expert systems are designed and implemented for dedicated narrow and specific domain, such as used in the distributed network management (Nehra *et al.*, 2007), while sell expert system can be used for developing expert system in any domain, but shell expert system are also governed by the format used for representation of the knowledge base.

The proposed scheme consists of the Rule-base and the Case-based formats using the Blackboard. The scheme facilitates combination of forward and backward chaining reasoning, using many problems solving methodologies and different searching techniques. The proposed scheme facilitates the common sense, deduction and analogical reasoning activities in the inference engine, because rule base provides the deduction, case base provides the analogical reasoning, and the blackboard provides the common sense, as seen in Fig. 1 (Owaied and Qasem, 2008).

Blackboard: The blackboard is a shared repository of problems, goals, partial solutions, suggestions and contributed information. The blackboard can be viewed as a dynamic library of requests and contributions that have been recently provided through the cooperation mechanism between the rule base knowledge and the case base knowledge.

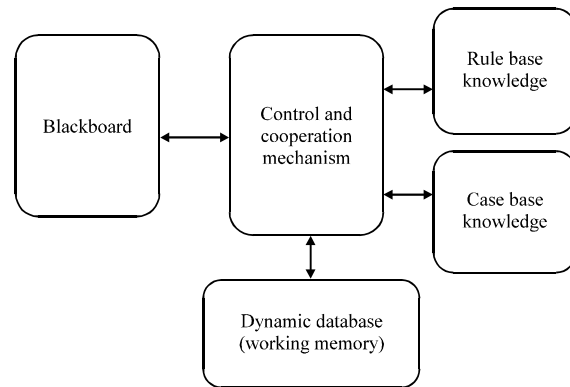


Fig. 1: The architecture of hybrid knowledge scheme

Control and cooperation mechanisms: The control mechanism is to control and reorganize the knowledge bases and used them in the most effective and coherent fashion.

The cooperation mechanism is the activities of passing the appropriate part of knowledge from one knowledge base to another and converting from one representation form into another. Control and cooperation mechanisms make use of the dynamic data base, the knowledge bases, and the blackboard for accomplish task.

Rule based knowledge: Rule base knowledge is the knowledge for certain domain represented in the form of Production rule, sometimes just referred to as rule, which consists of two part statements that embody small pieces of knowledge. The first part of the rule, called the antecedent, expresses a situation or premise while the second part, called the consequent, states a particular action or conclusion that applies if the situation or premise is true. The first or left-hand part of the rule is a statement (sometime called an action or conclusion). The second or right-hand part of the rule is the statements (sometime called the expressions or conditions), this is a form of Horn clause in First Order Predicate Logic.

Case based knowledge: This is the knowledge of a certain domain represents the collections of cases, each case representing as problem and its solution. The method used for representation any case is through the common aspects for all problems, these are:

- Initial state (s)
- Goal state (s)
- **Processes:** The processes of transforming from one state to another state in the problem space
- **Problem space:** Usually in the start consists of Initial state (s), goal state (s) and all the assertions given

There are many methods used for representation of problem space, such as directed graph. The problem solution will be representing as the path from initial state (s) to the goal state (s). Therefore any case can be representing as problem space, which consists of the initial state (s); goal state (s), path solution and the directed graph together with all processes have been used to find the solution.

Dynamic data base (working memory): This part usually is empty at the first but during the execution of the system this will be a collection of the assertions are generated from the processes of the cooperation between the knowledge bases in order to be used in the Blackboard. This can be regarded as working memory.

THE PROPOSED APPLICATION OF THE SCHEME

The proposed application for the hybrid scheme will be best as shell expert system called Rule-Case-Based shell expert system. The shell expert system is a complete development environment for building and maintaining knowledge-Based Applications and Expert Systems. It provides a step-by-step methodology for a knowledge engineer that allows the domain experts themselves to be directly involved in structuring and encoding the knowledge (Bellazzi *et al.*, 1999). Most expert systems are developed via specialized software tools called shell expert systems. These shells come equipped with an inference mechanism (backward chaining, forward chaining, or both) and require knowledge to be entered according to a specified format, user interface, explanation facilities and editing facilities as seen in Fig. 2 (Owaied and Qasem, 2010).

The hybrid knowledge representation scheme makes the proposed Rule-Case-based shell expert system more flexible, efficient and more powerful for the development of the expert systems in future. This view is based on the philosophy of human memory organization and utilizing for solving problems. Usually human uses more than one

form for knowledge representation in his long term memory in order to be more efficient for solving problems, also the knowledge of any domain can't be in one format. In the literature survey, found that many publications have covered the development of knowledge-based systems into expert system, using case-based reasoning in the areas of conceptual design, aircraft conflict resolution, military decision support systems, help-desk operations, customer service management, legal systems, diagnosis, design and planning (Sharma *et al.*, 2003; Aref and Al-Muhtaseb, 1997; Luger, 2005). It is seen that the applications of Case-Based Reasoning in developing knowledge-based systems and the expert systems have been adopted in various industries and other application areas (Smith *et al.*, 1998). The applications have been incorporate other knowledge representation methods besides rule-based and case-based reasoning, such as neural networks and fuzzy logic (Hall, 2001). Combination of forward chaining reasoning and backward chaining reasoning makes expert systems more flexible and efficient and also the use of more than one knowledge representation forms makes the expert system more powerful. Therefore, the mixing of rule-base and the Case-based forms using Blackboard has not been used before for the shell expert systems.

IMPLEMENTATION OF THE PROPOSED SCHEME

In reality, usually human have two types of knowledge which are Procedural and Declarative, so the proposed scheme will use both types of knowledge, which are the Rule base presents as Procedural and Case based presents as declarative. The following subsections present the detail description for the implementation of the proposed scheme as Rule-Case-Based shell expert system. The description present the methodologies used for creating, retrieving and updating of both Rule-base and Case-base.

User interface: The user interface simulates the communications facilities available to be used for interaction with the Rule-Case-Based shell expert system. This means an information processing system of one of (vision, speech, hearing, touching, tasting) or specified protocol many be used to connect the shell expert system to another computerized system. Usually the chosen method or methods to interact with the shell expert system will be based on format used for the representation of knowledge in the knowledge base. Since, the formats used in the proposed system will be a scheme of the integration of two formats, which are rule-base and case-base, so the user interface will be the

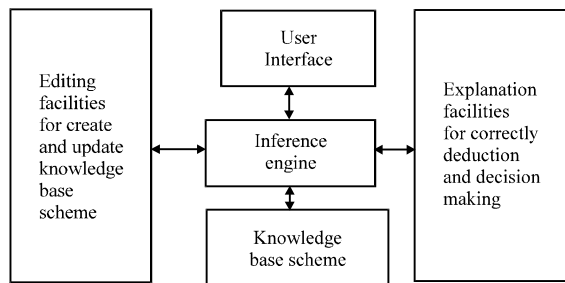


Fig. 2: The rule-case-based shell expert system

appropriate communication facilities between the proposed Rule-Case-Based shell expert system and the domain expert peoples.

These facilities allow the user (the domain expert peoples) to create and update the knowledge-base during the development of the expert system. But if the proposed Rule-Case-Based shell expert system connected to a computerized knowledge acquisition system then the interaction between two computer-based systems will be through the special protocols between them and should be appropriate with the proposed scheme for representation of the knowledge base.

The inference engine: The inference engine was playing the most important role in the construction of functional model of human system as mentioned by Owaied and Mahmoud (2007). But its implementation depends on the representation of knowledge in the knowledge base of the shell expert system. Therefore, the implementation of the inference engine will be regarded as a combination of problem solving method, reasoning agent and search technique. Unfortunately, it is difficult to implement general problem solving method for any field, or a general search technique for any field also. The reasoning agent is responsible to accept sophisticated queries concerning general knowledge to deduct specific knowledge in order to use by the problem solving method and the searching technique. The power of the solver-reasoning agent can be increased by implementing a larger number of solvers and by enhancing their capabilities to solve complex tasks. The use of case base format will be facilitates the analogical reasoning and the use of rule base format will facilitates the deduction during the process of solving a problem. The use of blackboard and dynamic memory together with analogical reasoning will a simulation of the common sense of human beings. Therefore the inference engine is a simulation of human behavior for solving a problem using the activities of deduction, analogical reasoning and common sense.

The rule base: In this project the relational database will be used to represent the rule as table. The rules will be stored in a table format with the maximum of number of column are k, for example if k = 6, then (Col-1, Col-2 ... Col-6), as shown in Table 1. The first column represents the left-hand-side of the rule, which is the conclusion of a rule usually called action (A) and from column-2 to column-6 are used to represent the conditions of the rule (C1, C2... C5), so this rule will be as Horn clause presented as follows:



Table 1: The layout of a rule in the table

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6
A	C1	C2	C3	C4	C5

In this view assume that any rule has maximum conditions are 5, but if a rule has more conditions, then the sixth column will be sub-action which has the reset of the conditions and so on. In this case the representation of knowledge is procedural representation not declarative representation. Some examples will shows how the conditions are going to be stored in the table depend on the number of conditions.

Creating table: The following Fig. 3 presents the flowchart used to create table with 6 columns and to check the availability of a rule in the table or not before saving it, which means checking the Action of a rule and its Conditions.

The following is the pseudo code for creating table.

```

10 Create matrix (number of rows, 6)
20 Initialize I to zero
30 Initialize number of column to zero
40 Input the action name go to availability algorithm
60 If the availability is equal to true
70 Go to 40
80 Else save action name to table
100 Set number of column is equal to number of column plus one
110 Input the number of condition
120 If number of conditions is less than or equal to five
130 Go to condition algorithm
140 Else go to formula algorithm
    set Value to j is equal to number of rows
170 Initialize I to zero
180 Initialize number of column to one
190 Go to condition algorithm
200 If number of column is equal to five
210 Save action name concatenate with number of rows
220 Set number of column is equal to zero
230 Save action name concatenate with number of rows
240 Set number of column is equal to one
250 Numbers of rows is equal to number of rows minus one go to
    condition algorithm
270 Else go to condition algorithm
    
```

The following is the pseudo code for condition algorithm.

```

10 Input the condition value
20 Go to check availability algorithm
30 If availability is equal to true go to 10
50 Else save condition value in table
70 Set I is equal to I plus one set number of column is equal to
    number of column plus one
90 If I is equal to number of column
100 Exit
110 Else go to 10
    
```

Checking availability of a rule: The following Fig. 4 presents the flowchart used to check the availability of a rule in the table or not before saving it which means checking the Action of a rule and its Conditions.

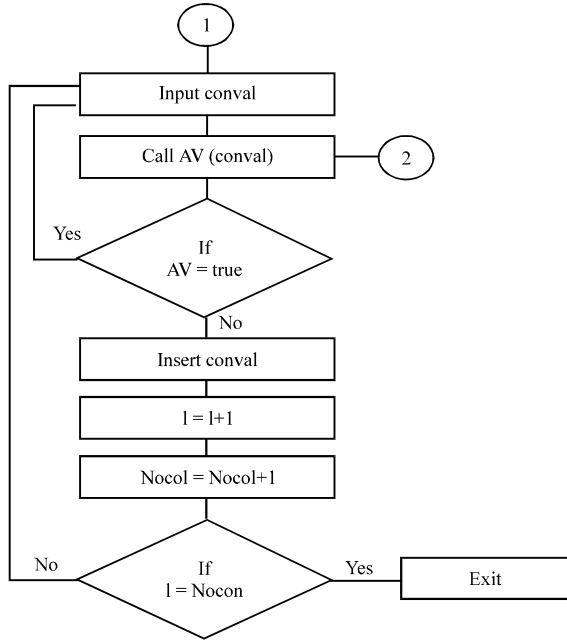


Fig. 3: The flowchart for creating table

The following is the pseudo code used for check the availability of a rule in the table.

```

10  Initialize number of column to
    zero
20  Initialize number of rows to zero
30  Set status is equal to "false"
40  If value is null
50  Set number of rows is equal to
    number of rows plus one
60  If column 1 is equal to action
    name set status to true
80  Else go to 50
100 Else set number of rows is equal
    to number of rows plus one
120 Set number of column is equal
    to number of column plus one
130 Get column concatenate with
    number of columns
140 If column concatenate with
    number of columns is equal to
    value status is equal to true
160 Exit
170 Else
180 If column concatenate with
    Number of columns is equal to null
190 Status is equal to false
200 Exit
210 Else
220 If number of columns is equal
    To Six exit
240 Else
250 If status is equal to true
260 Exit
270 Else
280 Set status to true
    
```

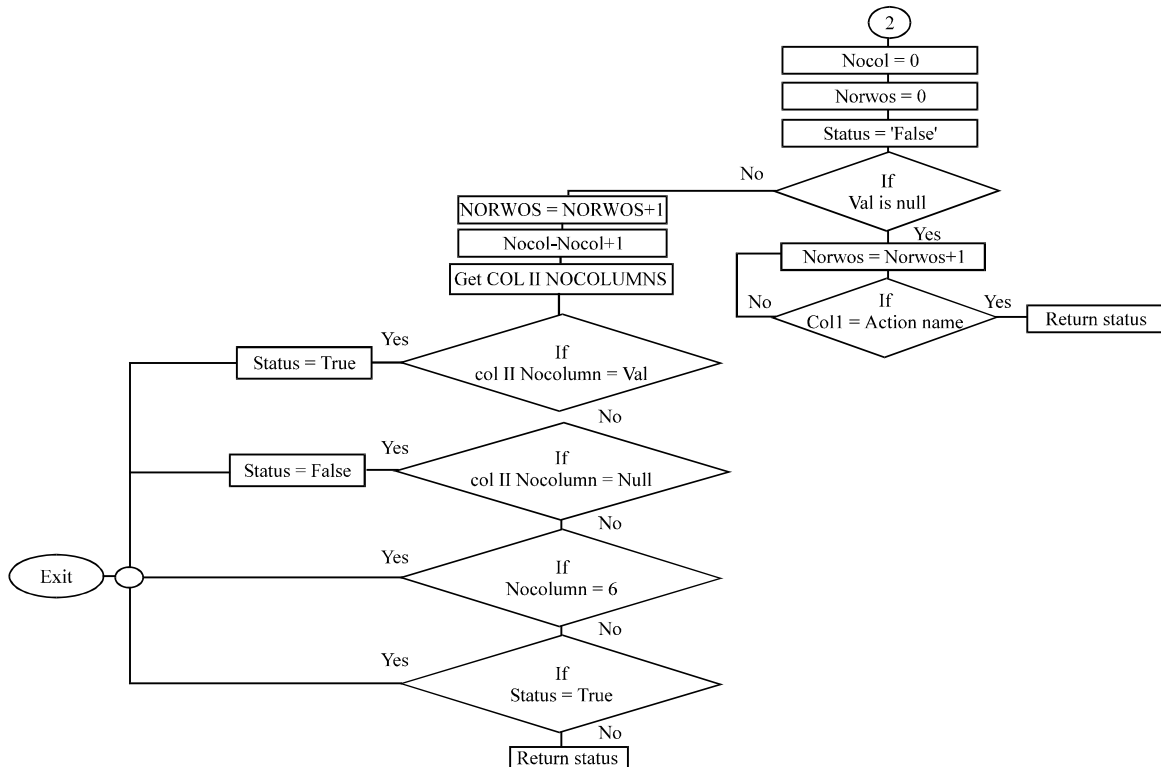


Fig. 4: The flowchart to check the availability of a rule in the table

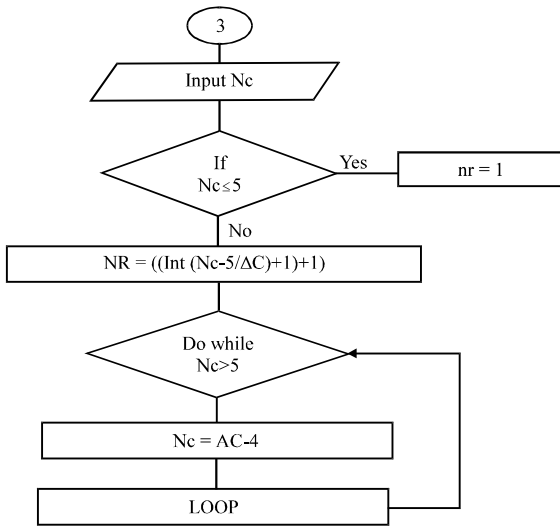


Fig. 5: Flowchart used to calculate the number of rows

Algorithm of calculating number of rows: The following formula is used to determine the number of rows required for a particular rule according to the number of conditions in the rule.

$$\text{No. of rows} = \begin{cases} \left\lceil \frac{NC-5}{\Delta C} + 1 \right\rceil + 1 & : NC > 5 \\ 1 & : NC \leq 5 \end{cases}$$

where, Nc is Number of conditions Δc: S the difference between the conditions to be stored in each row, its value 4 because the table contains 6 columns and the maximum conditions to be stored is 5.

Figure 5 presents the flowchart used to calculate the number of rows, for store a particular rule in the table, according to the number of conditions.

The following is the pseudo code used for calculates the number of rows.

```

10 Input the number of conditions
20 If number of conditions is less than or equal to 5
30 Set number of rows is equal to one
40 Else set number of rows is equal to
   initial (number of columns-5/Δc)+1)+1)
60 While number of columns is greater than five
70 Set number of columns is equal to
   number of columns minus four
  
```

The representation of general form of a rule: The following is the procedure for representing a rule in a table using the algorithm for calculating the number of rows required according to the number of conditions of the rule, can be seen in Appendix 1.

Table 2: The layout of the rule with conditions <= 5

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6
A	C1	C2	C3	C4	C5

Table 3: The layout of the general form of a rule

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6
A	C1	C2	C3	C4	A _{number of rows-1}
A _{number of rows-1}	C5	C6	C7	C8	A _{number of rows-2}
A _{number of rows-2}	C9	C10	C11	C12	A _{number of rows-3}
A _{number of rows-3}	C13	C14	C15	C16	A _{number of rows-4}

- Applying the algorithm for calculate number of rows, n
- If n ≤ 5, then the representation as shown in Table 2
- If n > 5, then the representation as shown in Table 3

Case base: Usually, the human experiences for solving problems in a certain domain present the collections of cases; each case presents a problem and its solution. Organizing the storage of the cases and retrieval of cases is central for effective case-based reasoning method, see the Appendix 2. Cases can be organized by the goal and retrieved when the case has the same goal as the current situation. Another organizing method is to use cases with most important features matched or the most number of features matched. The matching may first look for exactly matched case before looking for a more general case. Using the cases most frequently matched or most recently matched is also used when retrieving cases to match a new situation. Another method is to use the case that matches without much adjusting. Using these heuristics a similar case is retrieved.

The proposed method to organize the cases will be in three tables, each table consists of two columns. The first table: column one presents the case number and column two presents the case name, its Primary key is the case number and has one-to-many relationships with the second and third tables. The second table: Column one presents condition number and column two presents condition name, its Primary key is the condition number and has one-to-many relationship with third table. While the first column of the third table presents case number and the second column presents condition number, its Primary key is the case number and both columns are foreign keys one for second table and the other for third table. Figure 6 presents the relationships between tables; also all types of these relationships.

The following is the pseudo code used to create tables.

```

10 CREATE TABLE NAME CASES WITH TWO FIELDS
10.1 FIELD NAME: CASE_NO AND FIELD TYPE: NUMBER (4)
10.2 FIELD NAME: CASE_NAME AND FIELD TYPE: VARCHAR2 (100) NOT NULL
10.3 CREATE UNIQUE INDEX CASES_BRW_P1 ON CASES TABLE BY FIELD (CASE_NO)
10.4 CREATE PRIMARY KEY ON TABLE CASES USING FIELD CASE_NO ALTER TABLE
CASES ADD (CONSTRAINTCASES_BRW_P1 PRIMARY KEY (CASE_NO));
20 CREATE TABLE NAME CONDITIONS WITH TWO FIELDS
20.1 FIELD NAME: CONDITION_NO AND FIELD TYPE: NUMBER (4)
20.2 FIELD NAME: CONDITION_NAME AND FIELD TYPE: VARCHAR2 (100) NOT NULL
20.3 TO MAKE RELATION FROM TABLE (1) TO TABLE (2) USING FIELD NAME: CASE_NO
NUMBER (4)
20.4 CREATE UNIQUE INDEX CONDITIONS_BRW_P1 ON CONDITIONS
TABLE BY FIELD (CONDITION_NO)
20.5 CREATE PRIMARY KEY ON TABLE CONDITIONS USING FIELD CONDITION_NO
ALTER TABLE CONDITIONS ADD (CONSTRAINT CONDITIONS_BRW_P1
PRIMARY KEY (CONDITION_NO));
20.6 CREATE FOREIGN KEY ON TABLE CONDITIONS USING FIELD CASE_NO
ALTER TABLE CONDITIONS ADD (CONSTRAINT CONDITIONS_BRW_F1
FOREIGN KEY (CASE_NO) REFERENCES CASES (CASE_NO));
30 CREATE TABLE NAME CASES_CONDITIONS WITH TWO FIELDS
30.1 FIELD NAME: CASE_NO AND FIELD TYPE: NUMBER (4) NOT NUL
30.2 FIELD NAME: CONDITIONS_NO AND FIELD TYPE: NUMBER (4) NOT NULL
30.3 CREATE UNIQUE INDEX CASES_CONDITIONS_BRW_P1 ON
CASES_CONDITIONS TABLE BY FIELD (CASE_NO, CONDITIONS_NO)
30.4 CREATE PRIMARY KEY ON TABLE CASES_CONDITIONS USING FIELDS
CASE_NO, CONDITIONS_NO ALTER TABLE CASES_CONDITIONS ADD (CONSTRAINT
CASES_CONDITIONS_BRW_P1 PRIMARY KEY (CASE_NO, CONDITIONS_NO));
30.5 CREATE FOREIGN KEY ON TABLE CASES_CONDITIONS USING FIELD CASE_NO
ALTER TABLE CASES_CONDITIONS ADD (CONSTRAINTCASES_CONDITIONS_BRW_F1
FOREIGN KEY (CASE_NO) REFERENCES CASES (CASE_NO));
30.6 CREATE FOREIGN KEY ON TABLE CASES_CONDITIONS USING FIELD CONDITIONS_NO
ALTER TABLE CASES_CONDITIONS ADD (CONSTRAINT CASES_CONDITIONS_BRW_F2
FOREIGN KEY (CONDITIONS_NO) REFERENCES CONDITIONS (CONDITION_NO));

```

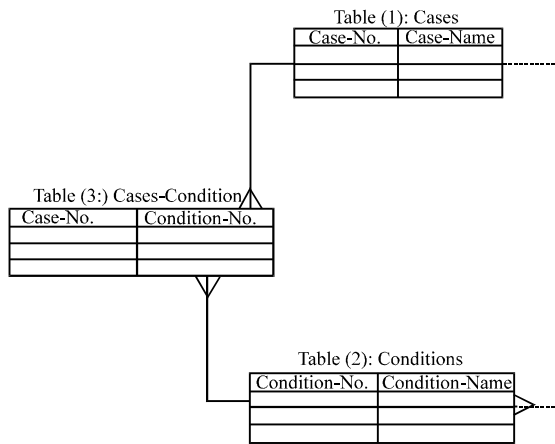


Fig. 6: The relationships between tables

ZThe blackboard: The blackboard is a shared repository of problems, goals, partial solutions, suggestions and contributed information. The blackboard can be viewed as a dynamic library of requests and contributions that have been recently provided through the cooperation mechanism between the rule base knowledge and the case base knowledge. In a case-based, a problem is matched against cases in the case base and one or more similar cases are retrieved. Case indexing involves assigning

indices to cases to facilitate their retrieval. In order to decide whether or not there is a similar case to retrieve for further processes, which means check the availability of case to retrieve as a condition or checking the Rule base? In order to retrieve cases efficiently, it is crucial to decide what the key attributes of a case are and on which attributes the cases should be indexed, see Table 6A. All these processes will be done in the Blackboard. A solution suggested by the matching cases is then reused. Unless the retrieved case is a close match, the solution will probably have to be revised, producing a new case that can be retained. Figure 7 presents the flowchart used for processing user query.

The following is the pseudo code used for processing user query.

```

10 Input number of conditions
20 Initialize I to zero
30 Input the condition
40 Set condition of I is equal to condition
50 Go to check availability algorithm
60 If the availability is equal to true
70 Set condition 1 is equal to condition
80 Set a1 is equal to a1 plus condition of I
90 Else
100 Set condition 0 is equal to condition
110 Set a0 is equal to a0 plus condition of I
120 If I is equal to number of conditions

```


130 If condition 1 is equal to number of
conditions retrieve action
150 Else if condition 1 is greater than or
equal to condition
170 Set similar action is equal to a1
180 Set new conditions is equal to similar
action concatenate with a0 go to check
availability algorithm for new
condition
200 If availability is true print action
220 else go to rule base algorithm
240 Else go to rule base algorithm
260 Else set I is equal to I plus one
280 Go to 30

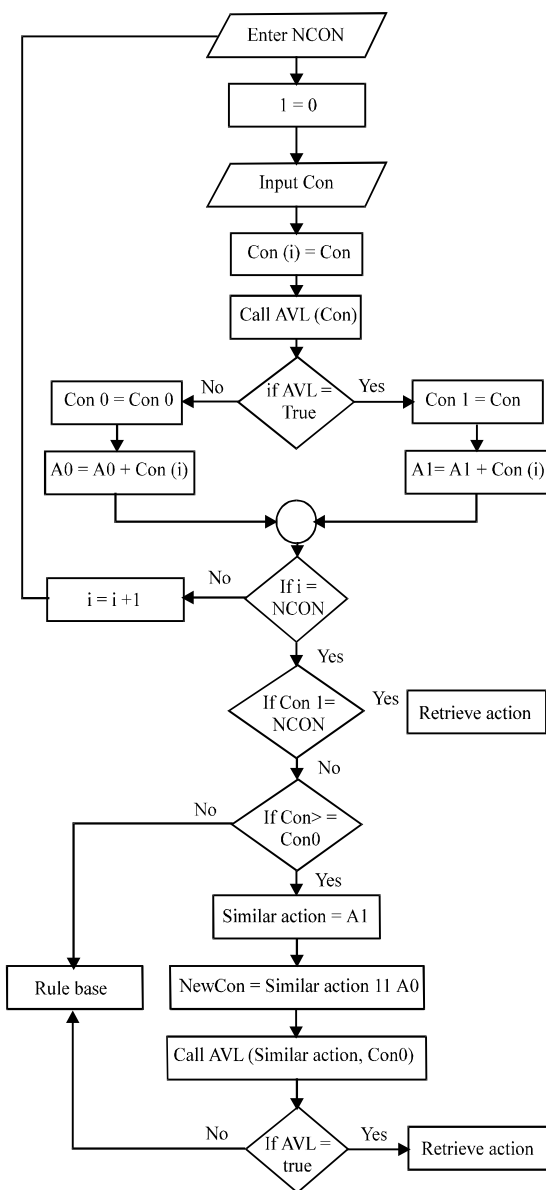


Fig. 7: The flowchart of processing user query

Control and cooperation mechanisms: The control mechanism is to control and reorganize the knowledge bases and used them in the most effective and coherent fashion. The cooperation mechanism is the activities of passing the appropriate part of knowledge from one knowledge base to another and converting from one representation form into another. Control and cooperation mechanisms make use of the dynamic data base, the knowledge bases (Rule base and Case base) and the blackboard in order to decide whether there is a similar case to retrieve for further processes or not. This means check the availability of case to retrieve as a condition in a particular rule or checking the rule base to find a particular rule to apply. All these processes have been done in the Blackboard.

CONCLUSION

This study presents a way of using rules base and case base knowledge representation forms through the use of Blackboard architecture. This view is based on the philosophy of human memory organization and utilizing for solving problems. Since human represent his knowledge in more than one form in order to be more efficient to solving a problem, also it's found that for any domain the knowledge can't be in one form. The proposed Hybrid Knowledge Representation Scheme, utilizes the combination of forward chaining reasoning and backward chaining reasoning, make the knowledge-based systems and expert systems more flexible and efficient and also the use of Blackboard architecture utilizes the flexibility of organization of knowledge bases and the flexibility of problem solving methodologies. The applications of the proposed scheme, as Rule-Case-Based Shell Expert System have been implemented. Therefore, the mixing of rule base and the Case base forms using Blackboard have not been used before for the implementation of shell expert systems.

APPENDIX

Appendix-1 illustration examples for rule base: The following are three examples, which are demonstrating the layout of the rules.

Example (1):
Pregnancy ← Missed one male, Nausea,
Generalized weakness,
Pregnancy test positive

The example in Table 1A, shows that the numbers of conditions are four which are stored in columns from

Col-2 to Col-5 and the Col-6 not used, the action will be stored in Col-1.

Example (2):

Sinusitis ← Headache, Itching in nose,
Sneezing, watering of eyes,
X-Ray show dizziness in the sinus

The example in Table 2A, shows that the numbers of conditions are five which are stored in columns from Col-2 to Col-6 means that all the columns in the table being engaged. The action will be stored in Col-1.

Example (3):

Tonsillitis ← Fever, Generalized fetid,
Pain in throat,
On examination redness,
Follicular, Some time pus

Example in Table 3A shows that the numbers to conditions are six, which are exceeding the numbers of columns, are allocated to store the conditions.

So the four conditions of the rule stored in Col-2, Col-3, Col-4, Col-5 and the head of the rule is stored in Col-1 and pretend that sub-action as a condition and store the sixth condition of the rule in Col-6 which has the same name of the head together with the index, in this example the index is 1. After that store the pretended condition in a new row as a new action and continue store the remaining conditions that have numbers five and six.

That is:

A ← C1, C2, C3, C4, A1
A1 ← C5, C6

Note that:

If we have a number of eleven conditions, the solution is:

A ← C1, C2, C3, C4, A1
A1 ← C5, C6, C7, C8, A2
A2 ← C9, C10, C11

Table 1A: The layout of the rule in example (1)

Cl-1	Cl-2	Cl-3	Cl-4	Cl-5
Pregnancy	Missed one male	Nausea	Generalized weakness	Pregnancy test positive

Table 2A: The layout of the rule in example (2)

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6
Sinusitis	Headache	Itching in nose	Sneezing	Watering of eyes	x-ray show dizziness in the sinus

Table 3A: The layout of the rule in example (3)

Col-1	Col-2	Col-3	Col-4	Col-5	Col-6
Tonsillitis	Fever	Generalized fetid	Pain in throat	On examine redness	Tonsilitis1
Tonsilitis1	Follicular	Some time pus			

APPENDEX-2

Illustration examples for case base: In the following examples, there are ten cases which are stored in the column two of the Table 4A, while the first column stores the index numbers for the cases as shown below, each case contains a number of conditions depend on the case

Table 4A: CASES

Case Number	Case Name
1	Pregnancy
2	Sinusitis
3	Otitismedia
4	HEMERROID
5	ACUTE COJUCTIVITIS
6	EPIDIDEMOORCHITIS
7	INTESTINSL OBSTRUCTION
8	UPER RESPIRATORY TRACT INFECTION
9	TONSILITIS
10	ABSCCESS

Table 5A: Conditions

Condition number	Condition name
1	MISSED ONE MENS
2	NEUSEA
3	GENERALIZED WEEKNESS
4	PREG. TEST POSITIVE
5	HEADECH
6	VOMITTING
7	ITCHING IN NOSE
8	SNEEZING
9	WATERING OF EYES
10	X-RAY SHOW HIZZINESS IN THE SINUS
11	FEVER
12	PAIN IN EAR
13	DISCHARGE FROM EAR
14	REDNESS IN TEMPANIC MEMBRAN AND CANAL
15	PAIN IN ANAL AREA
16	CONSTIPATION
17	ITCHING
18	DISCHARGE
19	BLEEDING
20	REDNESS
21	TENDERNESS
22	DISCHARGE FROM EYES
23	PAIN IN TESTIS
24	HOTNESS
25	SWELLING
26	SOME TIME REDNESS
27	ABDOMENAL PAIN
28	EXAMINATION TENDERNESS ALL OVER ABDOMEN
29	DECREASED OR ABSENT BOWAL SOUND
30	X-RAY SHOW MULTIPLE FLOUD LEVEL
31	BLOOD TEST ELECTROLIT ABNORMALATIES
32	COGHT
33	DEFICULTY OF BREATH
34	CHEST X-RAY SHOW ABNORMALATY
35	PAIN
36	GENERALIZED FETIQ
37	PAIN IN THROT
38	ON EXAMINATION REDNESS
39	FOLICULAR
40	SOME TIME PUS
41	COLLECTION OF PUS
42	HARDNESS
43	EXAMINATION OF THE AREA THERE IS FLUCTUATION

Table 6A: Cases condition

Case number	Condition number
1	1
1	2
1	3
1	4
1	5
1	6
2	5
2	7
2	8
2	9
2	10
3	11
3	12
3	13
3	14
4	15
4	16
4	17
4	18
4	19
5	20
5	17
5	21
5	9
5	22
5	5
6	23
6	24
6	25
6	26
7	6
7	27
7	28
7	29
7	30
7	31
8	32
8	33
8	11
8	34
8	35
9	11
9	36
9	37
9	38
9	39
9	40
10	35
10	41
10	20
10	43
10	21
10	24

given which stored in the second column of the Table 5A also the first column of the Table 5A shows the index number for the conditions, so the total cases are ten and the total conditions are 54 given. By using the relation called One-To-Many between the tables; it will be produce a new table contains two columns: The first column called Case_Number which is refers to the index number for the cases and the second column called Condition_Number which is refers to the index number for the conditions as shows in Table 6A. From Table 6A, it's

observed that the number of cases remains as it is while the number for conditions reduced to 43 instead of 54 without deleting any condition, i.e., 9 conditions are repeated in several cases which is not included in Table 6A. Thus the main advantages of this methodology is to avoid duplication in conditions, flexibility of marinating tables and easy for searching.

REFERENCES

Aamodt, A. and E. Plaza, 1994. Case-based reasoning: Foundational issues, methodological variations and system approaches. *AI Commun.*, 7: 39-59.

Abu-Naser, S.S., H. El-Hissi, M. Abu-Rass and N. El-Khozondar, 2010. An expert system for endocrine diagnosis and treatments using JESS. *J. Artificial Intell.*, 3: 239-251.

Abu-Naser, S.S., K.A. Kashkash and M. Fayyad, 2008. Developing an expert system for plant disease diagnosis. *J. Artif. Intell.*, 1: 78-85.

Aha, D. and J.J. Daniels, 1998. Case-based reasoning integrations: Papers from the 1998 AAAI workshop. Technical Report WS-98-15, AAAI Press, <http://www.aaai.org/Library/Workshops/ws98-15.php>.

Aref, M.M. and H.A. Al-Muhtaseb, 1997. Khabeer: An object-oriented arabic expert system shell. *Arabian J. Sci. Eng.*, 22: 275-293.

Bellazzi, R., S. Montani, L. Portinale and A. Riva, 1999. Integrating rule based and case-based decision making in diabetic patient management. *Lecture Notes Comput. Sci.*, 1650: 386-400.

Brown, C.E., 1995. Introduction to Artificial Intelligence and Expert Systems. Chapman and Hall, New York.

Chan, C.W., L.L. Chen and L. Geng, 2000. Knowledge engineering of an intelligent case-based system for help-desk operations. *Exp. Syst. Appl.*, 18: 125-132.

Chi, R.T.H. and M. Y. Kiang, 1991. An integrated approach of rule-based and case-based reasoning for decision support. *Proceedings of the 19th Annual Conference on Computer Science, (ACCS'91)*, New York, USA., pp: 255-267.

Fahmy, H.I. and C. Douligeris, 1999. Applications of hybrid fuzzy expert systems in computer networks design. Department of Electrical and Computer Engineering, University of Miami, <ftp://ftp.eng.shirazu.ac.ir/Documents/Expert%20System/Applications%20of%20Hybrid%20Fuzzy%20Expert%20Systems%20in%20Computer%20Networks%20Design.pdf>.

Hall, L.O., 2001. Rule chaining in fuzzy expert systems. *IEEE Trans. Fuzzy Syst.*, 9: 822-828.

- Hewahi, N.M., 2009. A hybrid architecture for a decision making system. *Int. Artif. Intell.*, 2: 73-80.
- Luger, G.F., 2005. *Artificial Intelligence Structures and Strategies for Complex Problem Solving*. 5th Edn., Addison Wesley, New Jersey.
- Nehra, N., N.K. Gondhi, D. Pant and R.B. Patel, 2007. Mobile agent based expert system for distributed network management. *Inform. Technol. J.*, 6: 891-896.
- Owaied, H.H. and M.A.A. Mahmoud, 2007. Functional model of human system as knowledge base system. *Proceedings of International Conference on Information and Knowledge Engineering*, June 25-28, CSREA Press, Las Vegas, Nevada, USA., pp: 158-161.
- Owaied, H.H. and M.M. Qasem, 2008. Hybrid Knowledge representation scheme for expert system. *Proceedings of International Conference on Artificial Intelligence*, July 14-17, Las Vegas Nevada, USA., pp: 159-163.
- Owaied, H.H. and M.M. Qasem, 2010. Developing rule-case-based shell expert system. *Proceedings of the International Multi Conference of Engineers and Computer Scientists*, March 17-19, Hong Kong, pp: 81-91.
- Prentzas, J. and I. Hatzilygeroudis, 2002. Integrations of Rule-Based and Case-Based Reasoning. In: *Advances in Case-Based Reasoning*, Craw, S. and A. Preece (Eds.). Springer-Verlag, Berlin, Heidelberg, pp: 336-349.
- Sabater, J., J.L. Arcos and R.L. De-Mantaras, 1998. Using rules to support case-based reasoning for harmonizing melodies. *AAAI Technical Report SS-98-04*, pp. 147-151, <http://www.aaai.org/Papers/Symposia/Spring/1998/SS-98-04/SS98-04-027.pdf>.
- Salim, M.D., A. Villavicencio and M.A. Timmerman, 2002. A method for evaluating expert system shells for classroom instruction. *J. Ind. Technol.*, 19: 1-11.
- Sharma, A.K., C. Kumar, K. Mustafa and A. Kumar, 2003. A fuzzy frame based expert shell. *Proceedings of National Workshop on IT Services and Applications*, Feb. 27-28, New Delhi, pp: 1-6.
- Smith, P., O. Taylor, J. MacIntyre and J.I. Tait, 1998. *Expert System Applications: Power Industry*. In: *Handbook of Expert Systems*, Lebowitz, J., (Ed.). CRC Press, New York.
- Tlili-Guiassa, Y. and L.M. Tayeb, 2006. Tagging by combining rules-based and memory-based learning. *Inform. Technol. J.*, 5: 679-684.
- Vossos, G., J. Zeleznikow, T. Dillon and V. Vossos, 1991. An example of integrating legal case-based reasoning with object-oriented rule based systems: IKBALS II. *Proceedings of International Conference on Artificial Intelligence and Law, (ICAIL '91)*, New York, USA., pp: 31-41.