Framework Model for Shell Expert System

Hussein H. Owaied †, Monzer Moh'd Qasem ††, and Hazim A. Farhan †††

† Faculty of I.T, CS Dept. Middle East University for graduate studies, P.O.Box 42, Amman 11610, Jordan
†† Faculty of Information Technology Computer Science Department princess Nour University, Al-Ryad, KSA
†††Faculty of I.T, CIS Dept. Middle East University for graduate studies, P.O.Box 42, Amman 11610, Jordan

Summary
In the design and implementation of any expert system, there are many problems should be considered and study carefully. In reality, human experts have common sense, deduction and analogical reasoning facilities. The proposed framework model for shell expert system is based on the integration of two different knowledge representation formats (scheme) in the knowledge base. The proposed scheme is the mixing of the Rule-base and the Case-based forms using Blackboard in order to include the three facilities in one scheme. This Scheme will be facilitate applying more than one problem solving methods and search techniques in inference engine of the shell expert system. The rule base and case base formats have been converted into tables. This paper presents the implementation of the proposed scheme as a Rule-Case-based shell expert system. Also, presents illustrated examples of using the framework model together with the evaluation of the proposed framework model. The evaluation is based on the methodologies used for developing shell expert system such as knowledge representations, searching technique, problem solving, and user interface design. The benefits of the propose scheme in the application of Rule-Case-based shell expert system to be more flexible, efficient, and more powerful for the development of the expert systems in future.

Key words:
Artificial Intelligence, Expert Systems, Shell expert Systems, Problem Solving Methodologies, Search Techniques and Relational Database**.**

1. Introduction

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, while shell 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 [1]. The proposed scheme consists of the Rule-base and the Case-based formats using the Blackboard [2]. The scheme facilitates combination of forward and backward chaining reasoning, using many problem solving methodologies, and different searching techniques. 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 [3]. This view is based on the philosophy of human memory organization and utilizing for solving problems [4]-[5]. Therefore, the mixing of rule-base and the Case-based forms using Blackboard has not been used before for the shell expert systems. Adapting the proposed scheme facilitates the common sense, deduction and analogical reasoning activities in the inference engine [6]. So this scheme will be used for the development of shell expert system as Rule-Case-based Shell Expert System.

2. Structure of 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 [7]. 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. 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. 1 [8].

Manuscript received November 5, 2009
Manuscript revised November 20, 2009

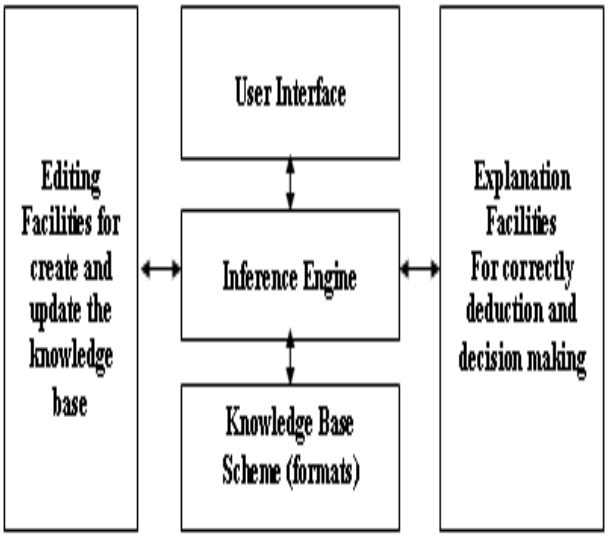


Fig. 1 Architecture of Rule-Case-Based Shell Expert System

2.1 Knowledge Base Scheme (formats)

Databases were built in relational database systems that allowed data to be extracted using Structured Data Query (SQL) tools [9]. Therefore, relational database systems will be used for the implementation of Knowledge Base Scheme (formats) for the Rule-Case-based Shell Expert System. The implementation consists of three tables; as table 1, table 2 and table 3 are labeled, Cases table, Conditions table, and relations between Case_table & Condition_table respectively. Table 1 consists of two columns, the first column labels by Case_No while the second columns labeled by Case_Name, a set of cases were stored in the Case_Name column. The Column Case_Number is assigned as primary key and the relation type used between the tables 1 and table3 are one-to-many and the relation used between the table 1 and table 2 in one-to-many also. Table 2 consists of two columns also, the first column labeled by Condition_No while the second column labeled by Condition_Name, a set of conditions were stored in the Condition_Name column. The Column Condition_Number is assigned as primary key and the relation type used between the table 2 and table 3 is one-to-many. Table 3 consists of two columns, the first column labeled by Case_No while the second column labeled by Conditions_No. The columns (Case_Number and Condition_Number) is assigned as a primary key in table 1 and table 2 while the same columns are assigned as a foreign keys in table 3.

Table: 1 Cases table

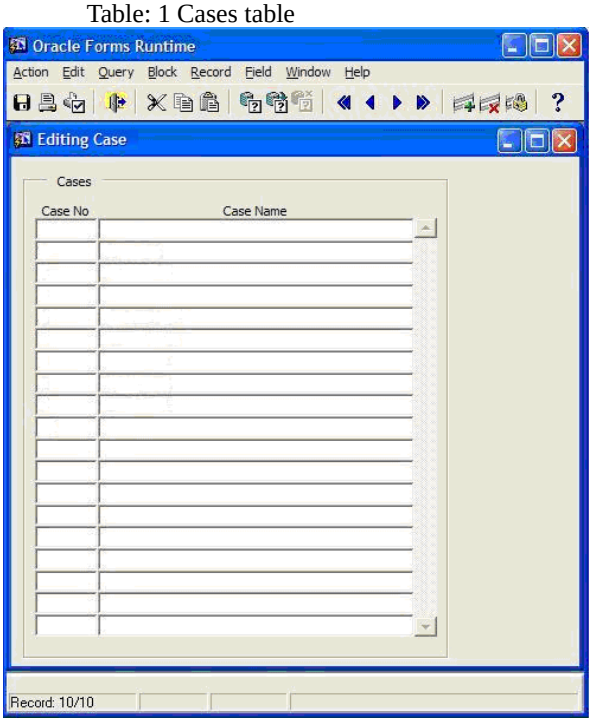


Table: 2 Conditions table

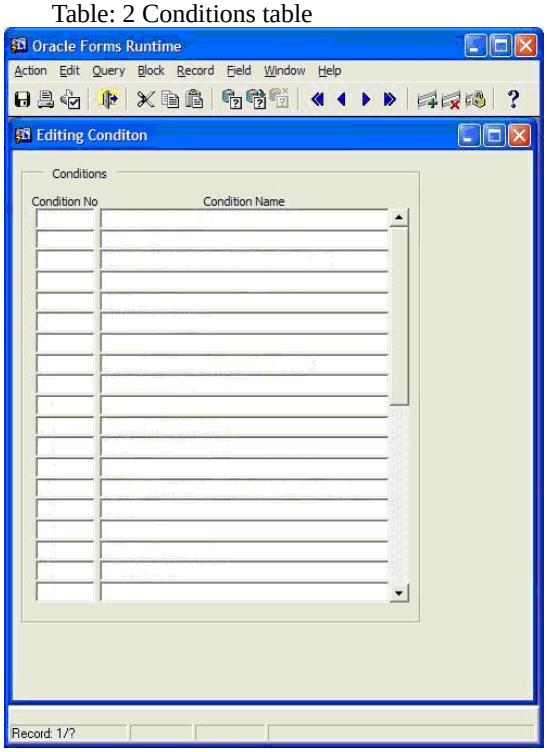
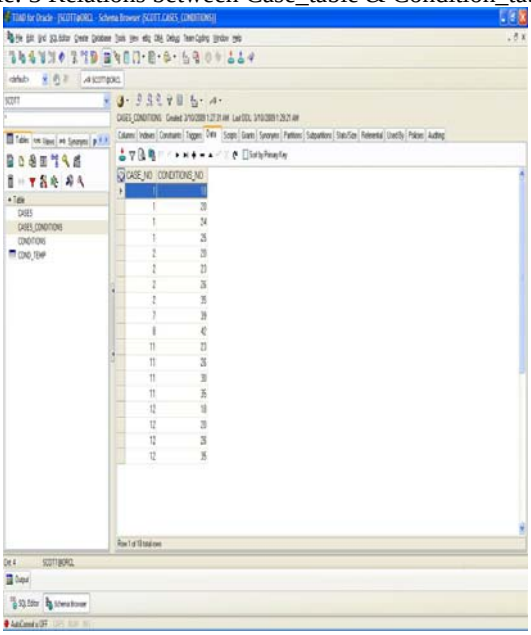


Table: 3 Relations between Case_table & Condition_table



2.1.1 User Interface

The implementation of user interface consists of many Oracle forms, such as the main menu, which consists of four phases as seen in Fig. 2, when the system started, so this menu will be displayed in order to allow the user to select one of the following phases.

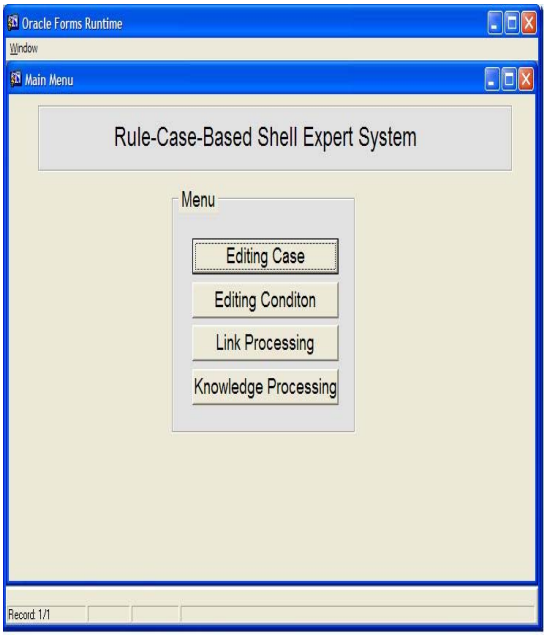


Fig.2: User Interface

A. Editing case base knowledge: This form is used to editing the case base knowledge, which means the user can insert a new case to be added to the case base knowledge, delete an existing case, or amending existing case. For example, there are ten cases were stored in the case base knowledge as seen in Fig.3, then the user have a capability of using editing facilities (insert, delete, update) to the case base knowledge.

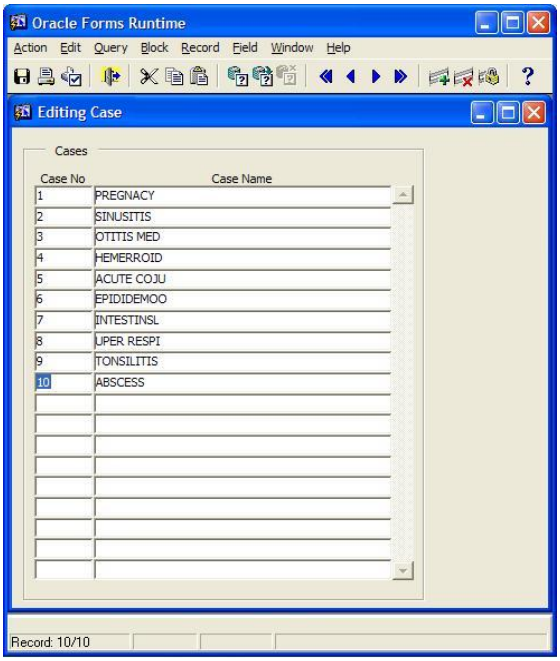


Fig. 3: Editing case base knowledge

B. Editing rule base knowledge: This form is used to editing the rule base knowledge, which means the user can insert a new rule to be added to the rule base, delete an existing rule, or amending existing rule. Since the rules are stored in tabular form, so the editing rule will be done through the processing of relational database using SQL statements. For example, a new condition(s) can be added to a specific rule as seen in Fig. 4, in which that specific rule will be updated through the link processing.

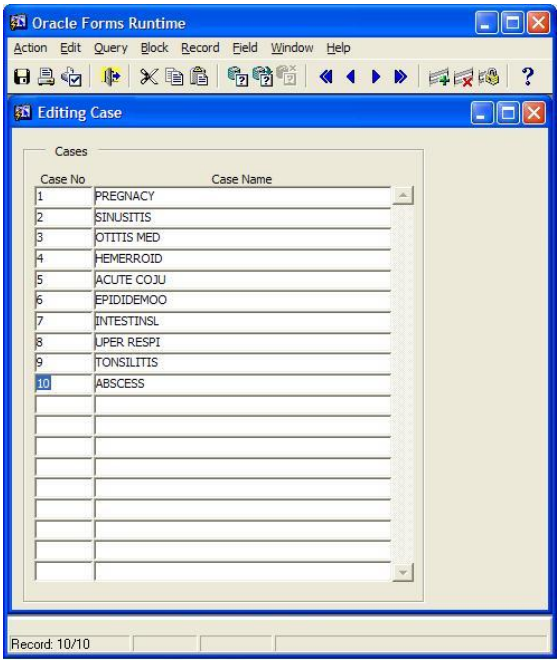


Fig. 4: Editing rule base knowledge

C. Link Processing: to make relation between the action (in the implementation is called case name for standardization of both case base knowledge and rule base knowledge) of a rule or case and its conditions as shown in Fig. 5.

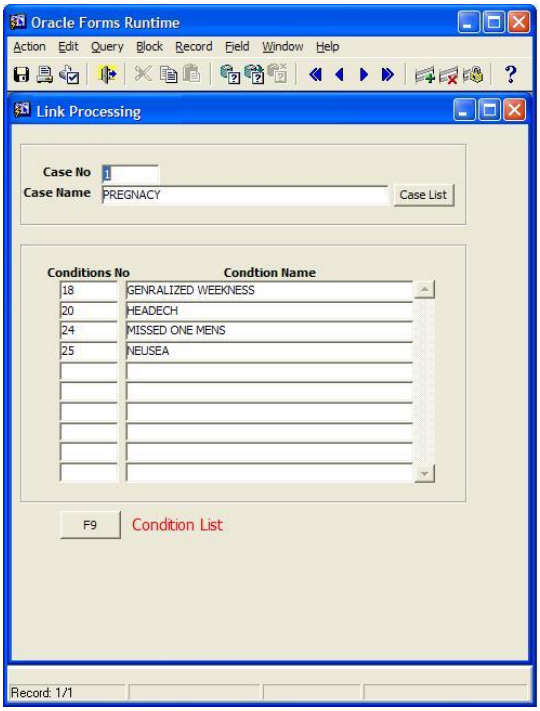


Fig. 5: Link Processing

In Fig. 5 the user can make the link between the cases and conditions such that by entering the case name in the field case name or press no. to Case List bottom directly a menu will appear which contains all cases were stored before, by selecting the case desired after that the user should insert the conditions related to that case in the field Condition Name or by pressing on F9 bottom directly a menu will appear which contains a set of conditions were stored before, by selection the conditions desired the link process will determine.

D. Knowledge Processing: to enter the conditions and the system will retrieve the case related to the conditions entered as shown in Fig. 6. The form consists of several fields, the user should insert the number of conditions will be use for query about the case and should insert the condition number for all conditions entered or by pressing on Conditions List, a menu will appear which contains all conditions were stored before as shown in Fig.7, by selecting the conditions required the result will present in conclusion box and will be shown later by using examples, the Temp Date shows the process done and some details for the conditions entered to reach to the conclusion.

- Case No: will show the case number from table 2.
- Case name: will show the case name from table 2.
- Condition No: will show the condition number from table 3.
- Condition name: will show the condition name from table 3.
- Status: If the condition available the status value will be 1 otherwise will be 0.

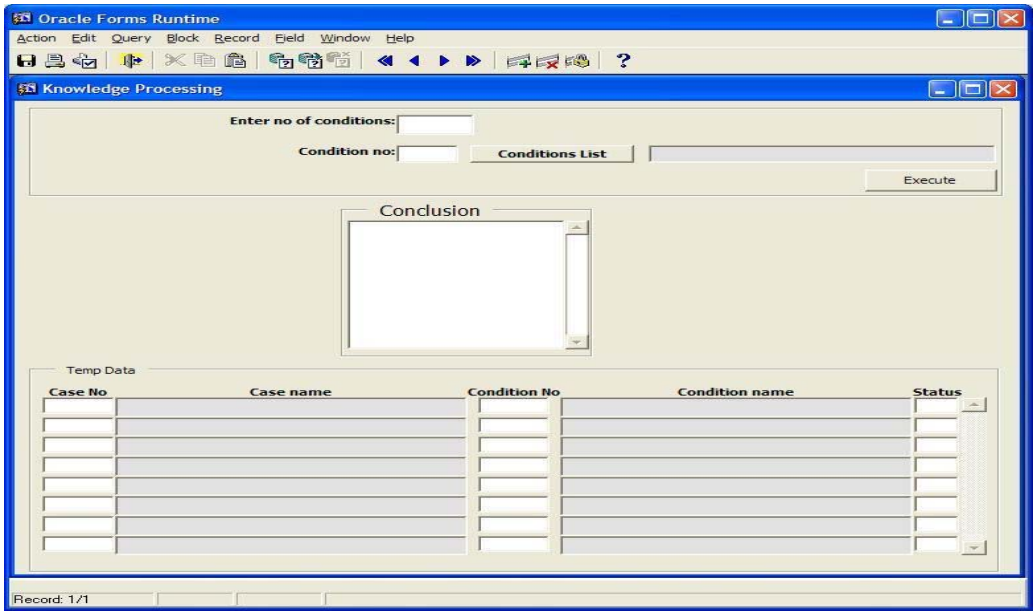


Fig. 6: Knowledge Processing

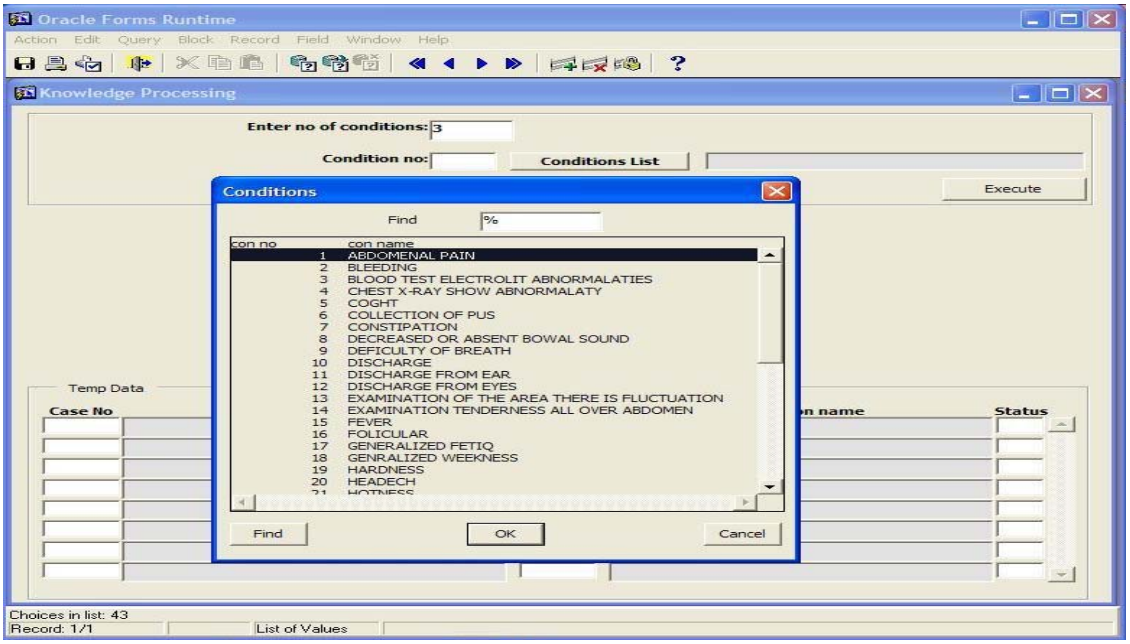


Fig. 7: Conditions List

2.1.2 The illustration Examples

Example (1):-

Number of conditions entered: 3			
Condition Number	23	26	35
Condition Name	ITCHING IN NOSE	ON EXAMINATION REDNESS	SNEEZING
Status	1	1	1
Conclusion	Exact Case as shown in Fig. 8		
Note	all the conditions entered which available in the same case		

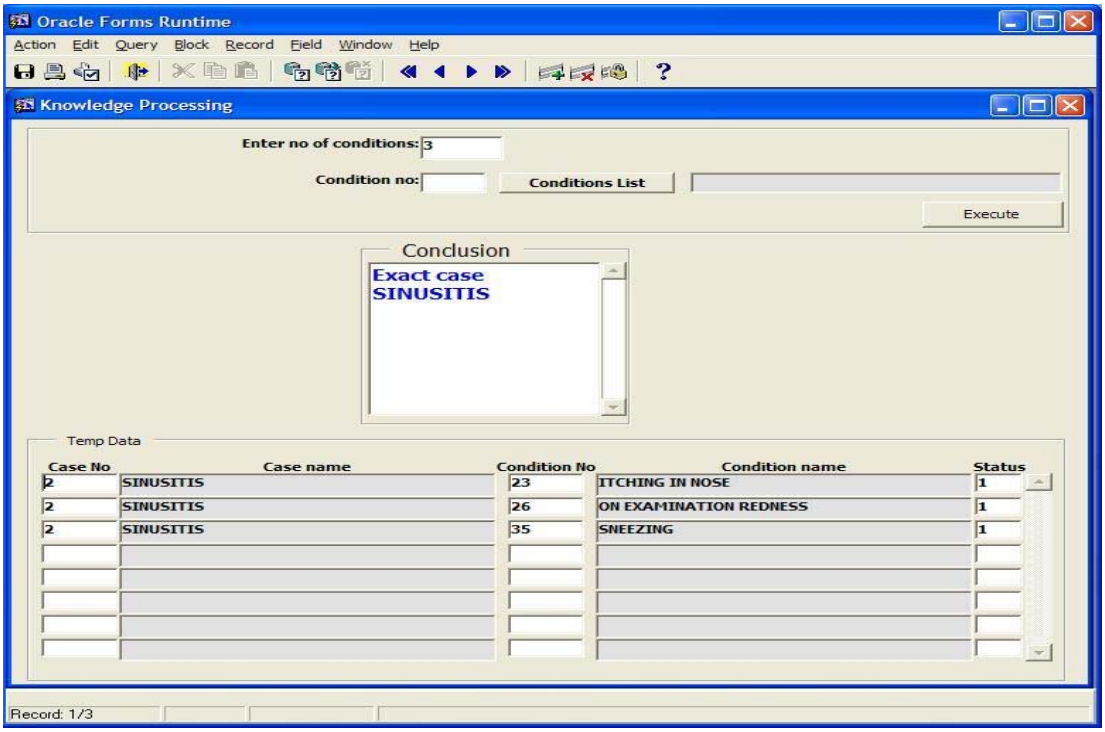


Fig. 8: Exact case

Example (2):-

Number of conditions entered: 4				
Condition Number	18	20	24	25
Condition Name	GENRALIZED WEEKNESS	HEADECH	MISSED ONE MENS	NEUSEA
Status	1	1	1	1
Conclusion	Match Case as shown in Fig. 9			
Note	all the conditions entered which available in the same case and the condition number 20 also available in other case.			

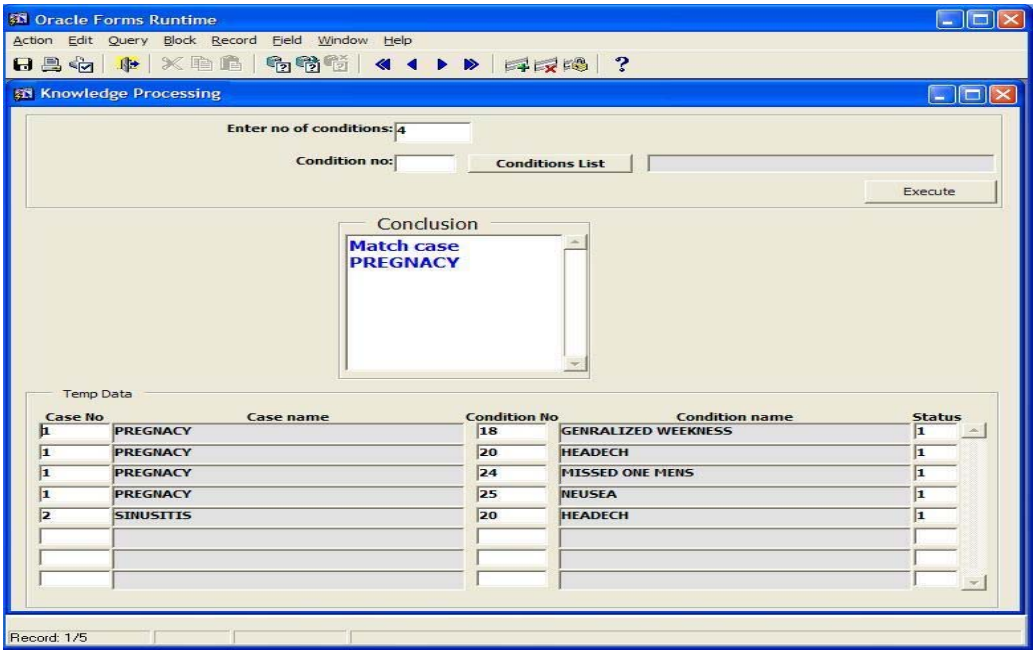


Fig. 9: Match case

Example (3):-

Number of conditions entered: 4				
Condition Number	23	26	35	30
Condition Name	ITCHING IN NOSE	ON EXAMINATION REDNESS	SNEEZING	PAIN IN TESTIS
Status	1	1	1	0
Conclusion	**Similar Case : SINUSITIS** **New Case : SINUSITIS + PAIN IN TESTIS**			
Note	conditions 23, 26, 35 are available in case named SINUSITIS which means similar case, but the condition number 30 not available in the same case, so the process will create a new case carrying the similar case with the condition having the status 0, so the total cases were stored in the cases table will be 11 as shown in Fig. 10 and Fig. 11.			

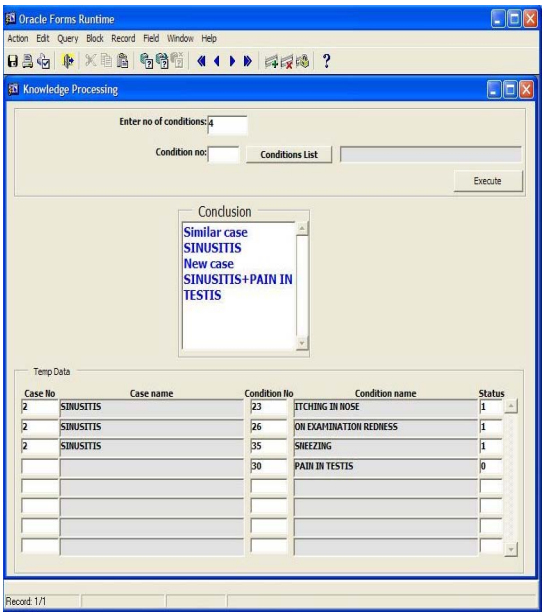


Fig. 10: New case

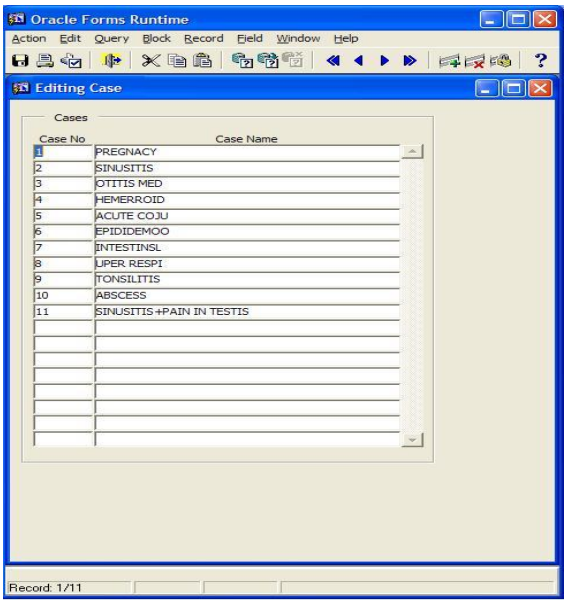


Fig. 11: Adding new case to cases_table

Example (4):-

Number of conditions entered: 4				
Condition Number	18	20	26	35
Condition Name	GENRALIZED WEEKNESS	HEADECH	ON EXAMINATION REDNESS	SNEEZING
Status	1	1	1	1
Conclusion	**Similar Case : SINUSITIS** **New Case : SINUSITIS + GENRALIZED WEEKNESS**			
Note	Conditions number 18 and 20 are available in case pregnancy. Conditions number 20, 26 and 35 are available in case sinusitis. Conditions number 26 and 35 are available in case sinusitis+ pain in testis The Case SINUSITIS has maximum number of accepted conditions comparing to other cases, so the process will present this case as a similar case and create a new case carrying the similar case with the condition number 18, so the total cases were stored in the cases table will be 12 as shown in Fig. 12 and Fig.13.			

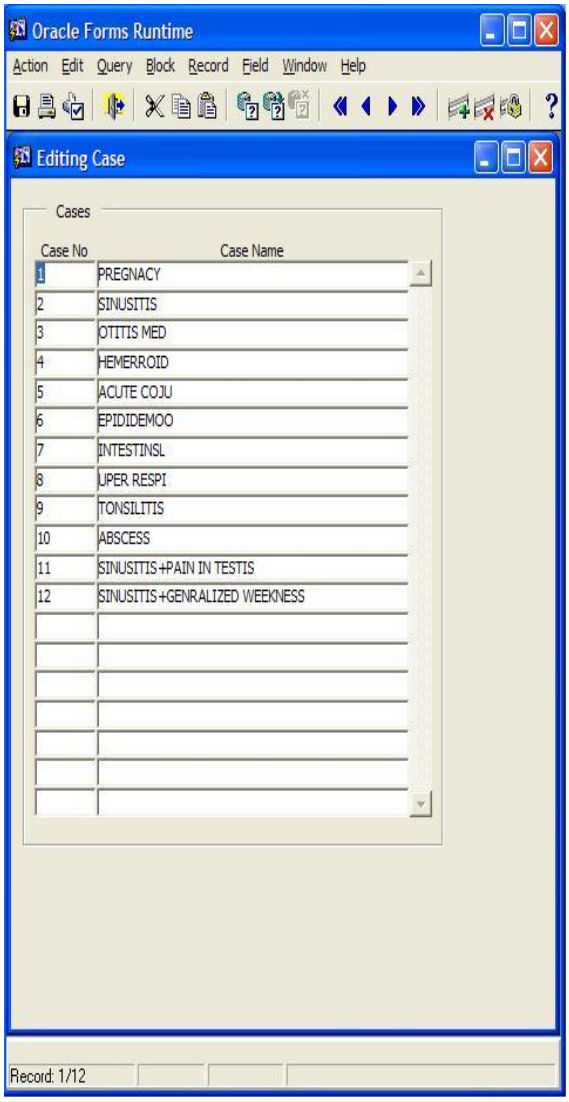


Fig. 12: Similar case

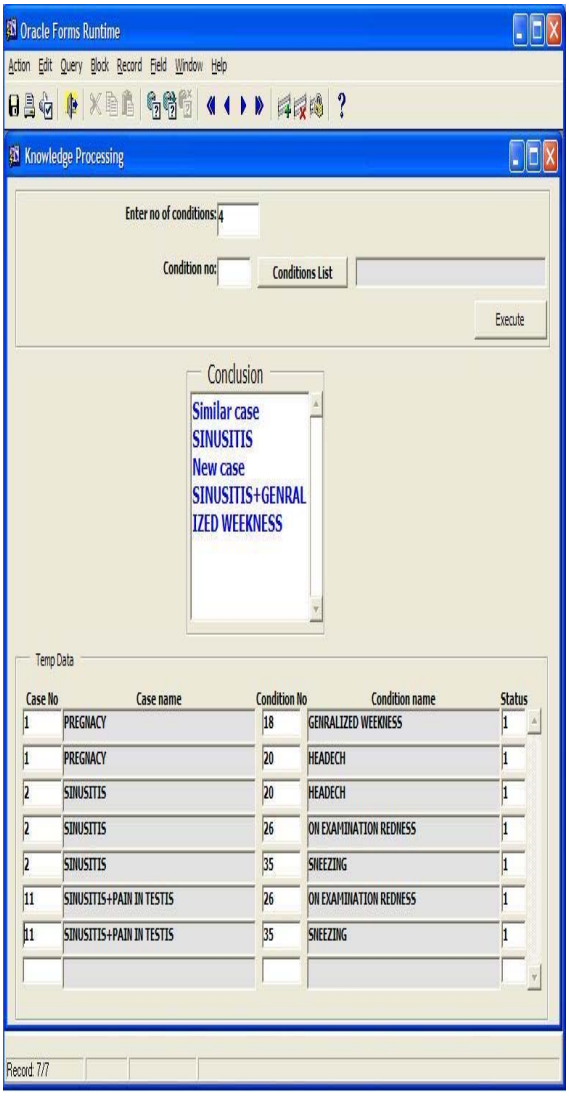


Fig. 13: Adding another new case to cases_table

Example (5):-

Number of conditions entered: 2		
Condition Number	39	42
Condition Name	TENDERNESS	X-RAY SHOW HIZZINESS IN THE SINUS
Status	1	1
Conclusion	Similar Case : INTESTINSL New Case : INTESTINSL + X-RAY SHOW HIZZINESS IN THE SINUS	
Note	Condition number 39 is available in case INTESTINSL. Condition number 42 is available in case UPER RESPI. Each condition is available in different case, so the process will select the first case appear and pretended as a similar case and create a new case carrying the similar case with the condition number 42 as shown in Fig. 14.	

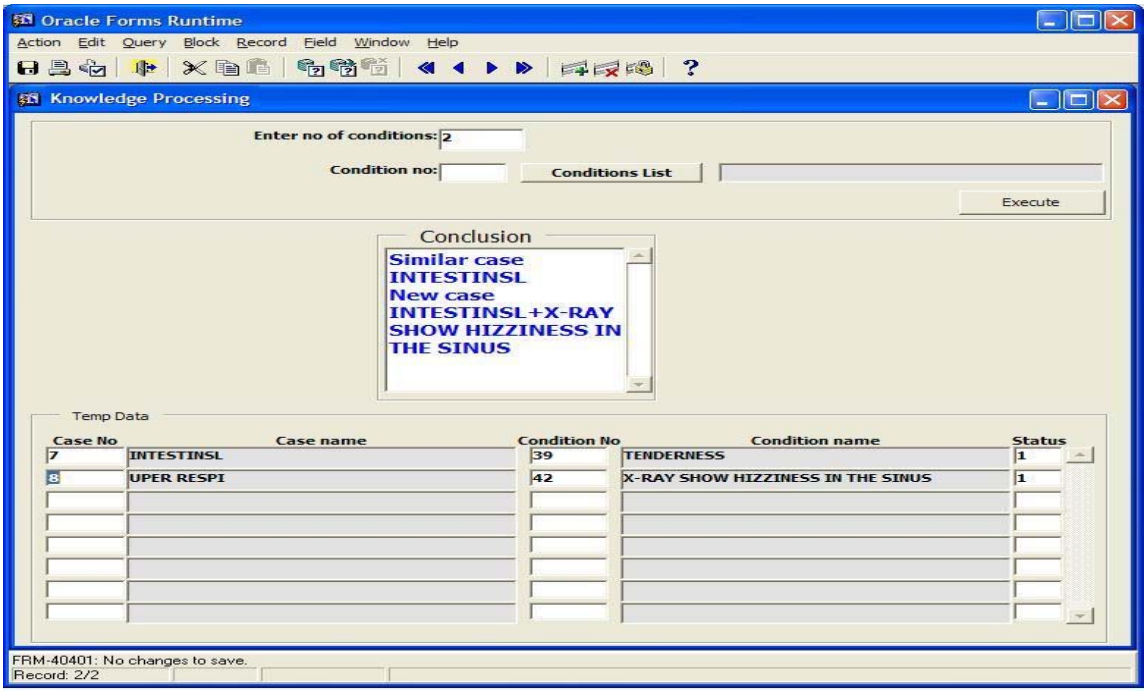


Fig. 14: Equivalent case

Example (6):-

Number of conditions entered: 5					
Condition Number	1	2	36	38	40
Condition Name	ABDOMENAL PAIN	BLEEDING	SOME TIME PUS	SWEELING	VOMITING
Status	0	0	0	0	0
Conclusion	No match case as shown in Fig. 15				
Note	There is no case having any one of the entered conditions.				

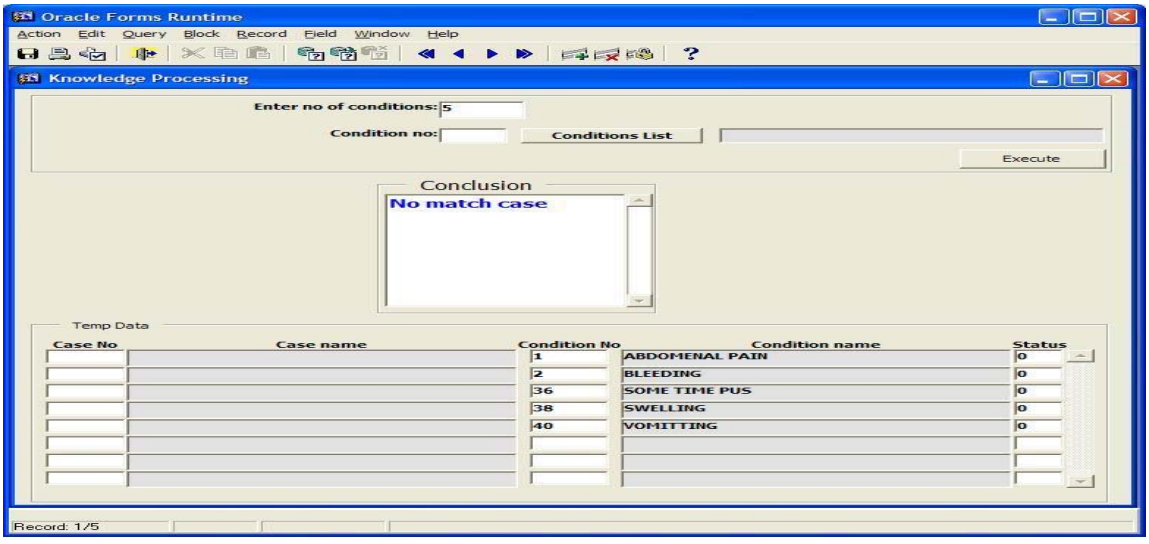


Fig. 15: No match case

3. Materials and Methods for Evaluation Shell Expert Systems

The evaluation is based on three other shell expert systems together with the proposed Rule-Case-Based Shell Expert System (Al-NADA EXPSYS) using an evaluation methodology proposed by MD Salim, Alvaro Villavicencio, and Marc A. Timmerman [10]. The methodology uses two types of evaluation, either direct method or indirect method. The direct method generates data for "Satisfaction Level", which is a direct measure of the overall user satisfaction with the shell. The indirect method is based on estimating the resources needed to write a program using general computer language to implement an Expert System. This type of calculation is based on finding answers to the following questions," If a user wrote a program in some computer language to solve this problem instead of using a ready-made Expert System shell, how long would this program be, how many person-months would it take to write this program, and so on. In this paper the direct method is used and in the next subsection present the evaluation processes together with the results of the evaluation.

3.1 Evaluation Using Direct Method

This method is based on the factors which are related to the most important criteria's in the design and implementation of expert systems, which are; knowledge representation, search technique, problem solving method, and user interface. The method consists of many steps, these steps has been called instrument, and these steps are presented in table 4. The application of this method is presented in table 5, table 6, table 7, and table8. This instrument is completed by the evaluator as follows:

1. The evaluator obtains demonstration or sample copies of the software packages to be evaluated.
2. The evaluator selects a benchmark problem.
3. After running the bench-mark problem, the evaluator responds to the 19 questions in the instrument and estimates a quantitative answer to each question on a 0 to 5 scale with 5 being very true and 0 being very false.
4. Each numerical result is multiplied by a weight factor as given in the weight column.
5. The weighted values are summed and then divided by 26 (as for 19 questions, 7 questions have weight factor of 2 and 12 questions have weight factor of 1, i.e., 7X2 = 12 X1 = 26) to give a result in the numerical range of 0 to 5.

Table 4 Direct measurement test instrument

Category	Question	Assessment	Weight	Value x Weight
Correctness of Answer	Is there enough information to evaluate the software?		2	
	Does the software give the same answers as other methods of calculation?		2	
	Does the software give the same answer that a human expert would give?		2	
	Does the software provide the right answer for the right reasons?		2	
Accuracy of Answer	Is the software accurate in its answer(s)?		2	
	Is the answer complete? Does the user need to do additional work to get a usable result?		2	
	Is the procedure of getting the answer simple and dear?		2	
Correctness of Reasoning Technique	Does the answer change if new but irrelevant data is entered into the software?		1	
	Can the system clearly explain its reasoning technique to the user?		1	
	Does the system require a lot of irrelevant questions to reach the answer?		1	
Sensitivity	Does the answer change if irrelevant changes are made to the system's rules?		1	
Reliability	Does the software crashes or hang-ups in its host computer?		1	
	Does the system give warnings for cases involving incomplete data or rules?		1	
Cost Effectiveness	Is the cost of the system justified by its performance?		1	
	Does the shell have all the features listed in the vendor's literature?		1	
	Does the software still provide answers with incomplete knowledge?		1	
Limitations	Can limitations of the shell be detected at this point in time?		1	
	Does the shell allow the user to expand a program if needed?		1	
	Can the system learn from increased data or experience?		1	
Results	Add Weight x Value Divide by 26			
		5-4-3-2-1-0 True...False		5 = Satisfied User 0=Unsatisfied User

Table 5 MP2 Results. Direct measurement technique.

Category	Question	Assessment	Weight	Value x Weight
Correctness of Answer	Is there enough information to evaluate the software?	*5*	2	*10*
	Does the software give the same answers as other methods of calculation?	*5*	2	*10*
	Does the software give the same answer that a human expert would give?	*5*	2	*10*
	Does the software provide the right answer for the right reasons?	*5*	2	*10*
Accuracy of Answer	Is the software accurate in its answer(s)?	*5*	2	*10*
	Is the answer complete? Does the user need to do additional work to get a usable result?	*5*	2	*10*
	Is the procedure of getting the answer simple and dear?	*5*	2	*10*
Correctness of Reasoning Technique	Does the answer change if new but irrelevant data is entered into the software?	*5*	1	*5*
	Can the system clearly explain its reasoning technique to the user?	*3*	1	*3*
	Does the system require a lot of irrelevant questions to reach the answer?	*5*	1	*5*
Sensitivity	Does the answer change if irrelevant changes are made to the system's rules?	*5*	1	*5*
Reliability	Does the software crashes or hang-ups in its host computer?	*5*	1	*5*
	Does the system give warnings for cases involving incomplete data or rules?	*5*	1	*5*
Cost Effectiveness	Is the cost of the system justified by its performance?	*5*	1	*5*
	Does the shell have all the features listed in the vendor's literature?	*5*	1	*5*
	Does the software still provide answers with incomplete knowledge?	*1*	1	*1*
Limitations	Can limitations of the shell be detected at this point in time?	*5*	1	*5*
	Does the shell allow the user to expand a program if needed?	*4*	1	*4*
	Can the system learn from increased data or experience?	*0*	1	*0*
Results	Add Weight x Value Divide by 26			118 4.54
		5-4-3-2-1-0 True...False		5 = Satisfied User 0=Unsatisfied User

Table 6 EXSYS CORVID Results.
Direct measurement technique.

Category	Question	Assessment	Weight	Value x Weight
Correctness of Answer	Is there enough information to evaluate the software?	4	2	8
	Does the software give the same answers as other methods of calculation?	5	2	10
	Does the software give the same answer that a human expert would give?	5	2	10
	Does the software provide the right answer for the right reasons?	5	2	10
Accuracy of Answer	Is the software accurate in its answer(s)?	5	2	10
	Is the answer complete? Does the user need to do additional work to get a usable result?	5	2	10
	Is the procedure of getting the answer simple and dear?	4	2	8
Correctness of Reasoning Technique	Does the answer change if new but irrelevant data is entered into the software?	5	1	5
	Can the system clearly explain its reasoning technique to the user?	2	1	2
	Does the system require a lot of irrelevant questions to reach the answer?	2	1	2
Sensitivity	Does the answer change if irrelevant changes are made to the system's rules?	4	1	4
Reliability	Does the software crashes or hang-ups in its host computer?	5	1	5
	Does the system give warnings for cases involving incomplete data or rules?	3	1	3
Cost Effectiveness	Is the cost of the system justified by its performance?	3	1	3
	Does the shell have all the features listed in the vendor's literature?	4	1	4
	Does the software still provide answers with incomplete knowledge?	5	1	5
Limitations	Can limitations of the shell be detected at this point in time?	4	1	4
	Does the shell allow the user to expand a program if needed?	5	1	5
	Can the system learn from increased data or experience?	5	1	5
Results	Add Weight x Value			113
	Divide by 26			4.35
		5-4-3-2-1-0 True...False		5 = Satisfied User 0=Unsatisfied User

Table 7 ART * Enterprise Results.
Direct measurement technique.

Category	Question	Assessment	Weight	Value x Weight
Correctness of Answer	Is there enough information to evaluate the software?	5	2	10
	Does the software give the same answers as other methods of calculation?	5	2	10
	Does the software give the same answer that a human expert would give?	5	2	10
	Does the software provide the right answer for the right reasons?	5	2	10
Accuracy of Answer	Is the software accurate in its answer(s)?	4	2	8
	Is the answer complete? Does the user need to do additional work to get a usable result?	4	2	8
	Is the procedure of getting the answer simple and dear?	4	2	8
Correctness of Reasoning Technique	Does the answer change if new but irrelevant data is entered into the software?	4	1	4
	Can the system clearly explain its reasoning technique to the user?	3	1	3
	Does the system require a lot of irrelevant questions to reach the answer?	5	1	5
Sensitivity	Does the answer change if irrelevant changes are made to the system's rules?	4	1	4
Reliability	Does the software crashes or hang-ups in its host computer?	4	1	4
	Does the system give warnings for cases involving incomplete data or rules?	3	1	3
Cost Effectiveness	Is the cost of the system justified by its performance?	5	1	5
	Does the shell have all the features listed in the vendor's literature?	5	1	5
	Does the software still provide answers with incomplete knowledge?	4	1	4
Limitations	Can limitations of the shell be detected at this point in time?	4	1	4
	Does the shell allow the user to expand a program if needed?	4	1	4
	Can the system learn from increased data or experience?	3	1	3
Results	Add Weight x Value			112
	Divide by 26			4.31
		5-4-3-2-1-0 True...False		5 = Satisfied User 0=Unsatisfied User

Table 8 Al-NADA EXPSYS Results.
Direct measurement technique.

Category	Question	Assessment	Weight	Value x Weight
Correctness of Answer	Is there enough information to evaluate the software?	5	2	10
	Does the software give the same answers as other methods of calculation?	5	2	10
	Does the software give the same answer that a human expert would give?	5	2	10
	Does the software provide the right answer for the right reasons?	5	2	10
Accuracy of Answer	Is the software accurate in its answer(s)?	5	2	10
	Is the answer complete? Does the user need to do additional work to get a usable result?	5	2	10
	Is the procedure of getting the answer simple and dear?	5	2	10
Correctness of Reasoning Technique	Does the answer change if new but irrelevant data is entered into the software?	5	1	5
	Can the system clearly explain its reasoning technique to the user?	5	1	5
	Does the system require a lot of irrelevant questions to reach the answer?	3	1	3
Sensitivity	Does the answer change if irrelevant changes are made to the system's rules?	5	1	5
Reliability	Does the software crashes or hang-ups in its host computer?	0	1	0
	Does the system give warnings for cases involving incomplete data or rules?	5	1	5
Cost Effectiveness	Is the cost of the system justified by its performance?	5	1	5
	Does the shell have all the features listed in the vendor's literature?	5	1	5
	Does the software still provide answers with incomplete knowledge?	5	1	5
Limitations	Can limitations of the shell be detected at this point in time?	5	1	5
	Does the shell allow the user to expand a program if needed?	5	1	3
	Can the system learn from increased data or experience?	5	1	5
Results	Add Weight x Value			121
	Divide by 26			4.65
		5-4-3-2-1-0 True...False		5 = Satisfied User 0=Unsatisfied User

3.2 The Results of Evaluation

In table 9 presents the final results for the evaluation of the four shell expert system. Notice that the best result was the first one which is the proposed Rule-Case-Based Shell Expert System and gives 4.65. This evaluation was based on the methodologies used for developing shell expert system such as knowledge representations, searching technique, problem solving, and user interfacing design.

Table 9 Results of evaluation

Expert System Shell	Direct Method Results
Al-NADA EXPSYS	4.65
MP2	4.54
EXSYS CORVID	4.35
ART* Enterprise	4.31

4. Conclusion

The framework will be used for the development of shell expert system as Rule-Case-based Shell Expert System. The framework uses both procedural and declarative knowledge representation formalisms through the application of relational data base. So the rule base and case base formats have been converted into tables. Adapting the proposed framework facilitates the common sense, deduction and analogical reasoning activities in the

inference engine of the shell expert system. The framework makes the proposed Rule-Case-based shell expert system more flexible, efficient, and more powerful for the development of the expert systems in future. The following are the advantages of the proposed shell expert system:

- The system can learn from increased cases
- The system can provide answers with incomplete knowledge
- The system gives the same answer that a human expert would give
- The system is accurate in its answers
- The system can explain clearly its reasoning technique to the user using the TempData dialog box.

The proposed Rule-Case-Based shell expert system can be 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. Also, incorporate other knowledge representation methods besides rule-based and case-based reasoning, such as neural networks and fuzzy logic.

5. References:-

[1] Peter Jackson, Introduction to Expert Systems, 3rd edition, Addison-Wesley, 1999.
[2] Hussein H. Owaied , Monzer Moh'd Qasem, "Hybrid Knowledge representation scheme for expert system", WORLDCOMP'08, International Conference on Artificial Intelligence (ICAI 2008)Volume II, July 14-17,2008. Las Vegas Nevada, USA.
[3] A.J. Gonzalez and D.D. Dankel, The Engineering of Knowledge-Based Systems: Theory and Practice. Englewood Cliffs, NJ: Prentice-Hall, 1993.
[4] Harmon, S. Y., Hoffman, C. W. D., Gonzalez, A. J., Knauf, R., and Barr, V. B., "Validation of Human Behavior Representations," a B3 paper in Proc. Workshop on Foundations for Modeling and Simulation (M&S) Verification and Validation (V&V) in the 21st Century, SCS, San Diego, CA, 2002.
[5] M. Ross Quillian. "Semantic Memories", In M. M. Minsky, editor, Semantic Information Processing, pages 216-270. Cambridge, MA: MIT Press, 1968
[6] John E. Laird, the Consumer Reports Guide to Cognitive Architectures, Term: Winter, 1994.
[7] Ovidiu S. Noran, The Evolution of Expert Systems, Domain: Expert System Design and Development, School of Computing and Information Technology, Griffith University, http://www.cit.gu.edu.au/~noran
[8] Owaied H. Hussein, Abu-Arr'a M. Mahmoud: "Functional Model of Human System as knowledge Base System", The 2007 International Conference on Information & Knowledge Engineering, June 25-28, 2007,158-161.
[9] Alexander Gehre, Peter Katranuschkov, Human Centered Knowledge-Based Model Access Service for Engineers International Council for Research Innovation in CIB w78 Conference 2002.
[10] MD Salim, Alvaro Villavicencio and Marc A. Timmerman, Journal of Industrial Technology, A Method for Evaluating Expert System Shells for Classroom Instruction. Volume 19, Number 1 – November 2002 to January 2003, keyword search administration computer science information technology management, refereed article the official electronic publication of the national association of industrial technology, www.nait.org