

Expert System for Single Objective Criteria for Selection of Manufacturing Method

Girish R. Naik^{1,3}, Dr. V.A.Raikar², Dr. Poornima G. Naik³

¹Associate Professor, Production Department, KIT's College of Engineering, Kolhapur, India

²Principal Govt. Engineering College, Karwar, India

³Assistant Professor, Department of Computer Studies, Chh Shahu Institute of Business Education and Research, Kolhapur, India

Abstract: While material selection methods are becoming mature, the selection of manufacturing processes has been neglected particularly after the preliminary design stages. The manufacturing process selected must be an economical balance of materials, manpower, product design, tooling equipment, plant space, and many other factors influencing cost and practicality. The process must be selected in such a way that the produced product will be acceptable to the consumer functionally, economically and appearance wise. The process must assure a product that meets all design requirements of quality, function and reliability. The process should be able to meet daily production requirement, which at the same time should utilize full capacity of the machine and its tooling and should reduce to a minimum idle operator and idle machine time and must provide the maximum utilization of minimum amount of material. The process should be flexible enough to accommodate reasonable changes in design. This poses a great challenge to a manager in selection of effective and economical manufacturing process. Different organizations have different objectives and based on their specific requirement they deploy suitable process conforming to their objective. Based on their needs, the weights assigned to the objectives vary. In this paper we have made an attempt to enable the end user a selection of appropriate manufacturing method based on a single objective. In the present work, we have developed an expert system in Visual Basic (VB) for selection of a manufacturing method based on a single objective. Method and class rules are automatically generated by retrieving the data from the access database. For the problem under consideration 680 method rules and 110 class rules are generated and stored in a prolog database, methodrules.pl. The expert system shell is implemented in VB which interfaces with SWI Prolog with the help of java application using JPL library. The prolog query is generated on the fly based on the objective and the classes selected by the user in expert system shell which is evaluated using SWI prolog and the solutions are redirected to the text file, methods.txt. The expert system in Visual Basic reads the contents of methods.txt as computed by SWI prolog and presents the results to an end user. The methodology and application architecture is subject to future enhancements. Our current work focuses on a single objective. It is an idealistic scenario where a single objective defines the section of manufacturing method. In real situations multi objective criteria is required for the purpose. Nevertheless this is our first attempt towards development of such tool and our future work involves modification of the expert system to take account of multiple objectives and functions.

Keywords: class method, expert system, knowledge base, manufacturing objective, SWI prolog, JPL

1. Introduction

Manufacturing methods include methods of many different types. Some of the methods are of a technological nature, while others are organizational and architectural, and yet others focus on information technology. Some are of a practical nature while others are of a philosophical nature. To assist managers in selecting the best method to achieve certain criteria, two mapping methods are available, one based on the objectives of the method and the other based on the functions that the methods may serve. Based on the maturity of the manufacturing company, a particular manufacturing method may focus on manufacturing hardware, auxiliary software support, production planning and control, next generation production management, processing manufacturing methods, commercial aspects, organization, advanced organizational manufacturing methods, design methods, human factors in manufacturing, environmental manufacturing methods, or cost and quality manufacturing methods. In this paper we consider the following objectives in selection of a particular manufacturing method as proposed by Gideon Halevi et. Al [1].

1. Meeting delivery dates
2. Reduce production costs.

3. Rapid response to market demands
4. Reduce lead time
5. Progress towards zero defects
6. Progress towards zero inventory
7. Improve management knowledge and information
8. Improve and increase team work collaboration
9. Improve customer and supplier relationships
10. Improve procurement management and control
11. Management strategic planning
12. Improve human resources management
13. Improve enterprise integration
14. Continuous improvement
15. Environmental production
16. Marketing – market share.

The suitability of each method to a specific objective is graded according to the following grades.

- a – Excellent for specific dedicated objective
- b – Very good
- c – Good
- d - Fair

This paper focuses on assisting managers to evaluate and select the most appropriate manufacturing method or methods for their needs. Several techniques may be proposed, allowing the user to decide which one is more

suitable under the circumstances. The user can select the method according to its type. The decision depends on the objectives and the functions considered, and on the grading given to each method. The objectives and grades can be manipulated by the end user.

1) Objective Grading Table

The commonly adopted procedure for selecting a manufacturing method(s) based on a single objective is by constructing an objective grading table. The structure of the objective grading table is as follows. The objective grading table consists of 110 rows and 19 columns. The first column contains the method number. The second column contains the method initial for verification purposes. The third column contains the method classification. The following 16 columns refer to the 16 objectives. The blank cell indicates that the method in the corresponding row has nothing to do with the objective in the corresponding column.

2) Selecting the method using a single objective

The procedure for selecting a manufacturing method using a single objective is as follows:

- Select the column that represents the objective in objective grading table.
- Scan the rows in this column for grades a or b.
- Make an objective table that contains only the methods filtered in step 2.
- Decide which class of method to use
- Narrow down the table constructed in step 3 to those that correspond to the desired class.
- Decide which of the proposed methods is preferred.

In this paper we propose an alternative method for selecting manufacturing method based on a single objective using declarative programming approach. An expert system is developed in VB for that purpose.

3) Introduction to Prolog

Prolog, Programming in Logic, is a special type of declarative type programming in which the various program elements and constructs are expressed in predicate logic. A program consists of mainly a number of declarations representing relevant facts and rules concerning the problem domain. The solution to be discovered is also expressed as a question to be answered or to be more precise the goal to be achieved based on the resolution method suggested by Robinson consisting of matching goals with facts and rules. A prolog program consists of a finite sequence of facts, rules and a query or goal statement. Prolog database or knowledge base consists of facts and rules. Prolog inferencing system mainly consists of three mechanisms viz.,

- Backtracking
- Unification and
- Resolution.

Two interesting features of logic programming are non-determinism and backtracking. A non-deterministic program may find a number of solutions, rather than just one, to a given problem. Backtracking mechanism allows exploration of potentially alternative directions for solutions, when some direction currently being investigated, fails to find an appropriate solution.

2. Literature Survey

There exists a vast amount of literature on manufacturing process monitoring using both crisp and fuzzy logic approach [2, 9] which focuses mainly on software selection, technology selection and system project selection. ChenhuiShao et.al [10] has developed a novel algorithm for parameter tuning and feature selection. Quality monitoring is used for monitoring a quality of a manufacturing process. Multiple criteria decision making method is employed by R. V. Rao, T. S. Rajesh [11]. The authors have presented a decision making framework using a multiple criteria decision making method viz., Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE) which has been integrated with analytic hierarchy process (AHP) and the fuzzy logic. The framework enables the manager a software selection in manufacturing industries. Mohammad Akhshabi [12] has developed a Fuzzy Multi Criteria Model for Maintenance Policy which is used for the optimized decision making.

3. Proposed Algorithm

A. Pseudo Code

```
Function CreateMethodClassRules()
{
/* Create 680 method rules and 110 class rules by reading
the corresponding tables from the backend manufacturing
database */
con.Open
"Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\manufacturing.mdb"
rs.Open "SELECT * FROM ObjectiveGrading ORDER BY
methodnumber", con
Open "c:\methodrules.pl" For Output As #1
cnt=0
while (rs.EOF = false)
begin
cnt=cnt+1;
/* construct the method rule */
str = "objective_for_method(method"
str = str + CStr(rs.Fields(0)) + ", objective"
str = str + CStr(rs.Fields(1)) + ", "
str = str + rs.Fields(2) + ")."
print #1, str
rs.movenext
end if;
rs.close;
close #1

rs.Open "SELECT * FROM Method", con
Open "c:\methodrules.pl" For Append As #1
cnt=0
while (rs.EOF = false)
begin
cnt = cnt + 1
/* construct the class rule */
str = "method_in_class(method"
/* rs.Fields(0) is the methodnumber column in the
Method table */
str = str + CStr(rs.Fields(0)) + ", "
/* rs.Fields(2) is the classificationcode column in
the Method table */
```

```
str = str + LCase(rs.Fields(2)) + ")."
Print #1, str
rs.MoveNext
end if;
closerecordset object;
close connection object;
close file;
}
```

Function GenerateOutput()

```
{
/*Populates the grid with the selected method names based
on the single objective and classes by reading methods.txt
file */
```

```
Open "c:\run.bat" For Output As #1
Print #1, "set path=%path%;C:\Program Files\pl\bin"
Print #1, "set classpath=C:\Program Files\pl\lib\jpl.jar;.;c:\"
Print #1, "javac.exe c:\Methods.java"
str = "java Methods " & Combo1.Text
If ChkM.Value = 1 Then
str = str& " m"
End If
If ChkP.Value = 1 Then
str = str& " p"
End If
If ChkS.Value = 1 Then
str = str& " s"
End If
If ChkT.Value = 1 Then
str = str& " t"
End If
If ChkX.Value = 1 Then
str = str& " x"
End If
Print #1, str
Print #1, "Pause"
Close #1
```

```
Shell ("c:\run.bat")
Open "c:\query.txt" For Input As #1
```

```
While (Not EOF(1))
Input #1, str
str1 = str1 &str
Wend
Text2.Text = str1
Close #1
```

```
con.Open "Provider=Microsoft.Jet.OLEDB.4.0;Data
Source=C:\manufacturing.mdb;Persist Security Info=False"
Open "c:\methods.txt" For Input As #1
cnt = 1
While (Not EOF(1))
Input #1, str
cnt = cnt + 1
Wend
Close #1
Open "c:\methods.txt" For Input As #1
cnt = 1
While (Not EOF(1))
Input #1, str
str1 = Right(str, Len(str) - InStr(1, str, "d"))
displayInGrid(cnt, 0) = str1
rs.Open "SELECT methodname FROM method WHERE
methodnumber = " & str1, con
displayInGrid(cnt, 1) = rs.Fields(0)
rs.Close
cnt = cnt + 1
Wend
Close #1
con.Close

End If
}
```

4. Results and Analysis

The results presented above are implemented in VB with MS-Access as backend for storing method and objective details. The structure of the database is shown in the following Figure 1.

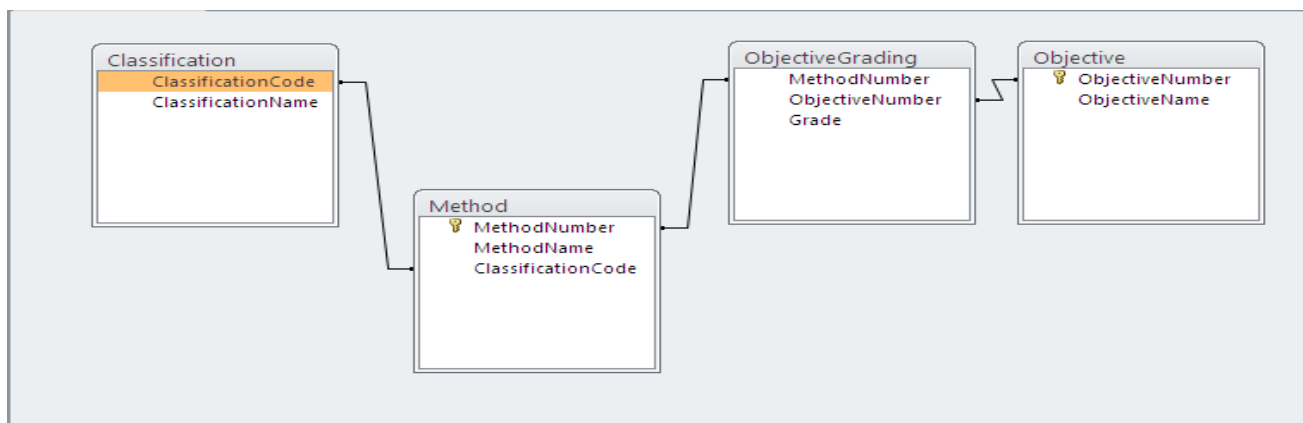
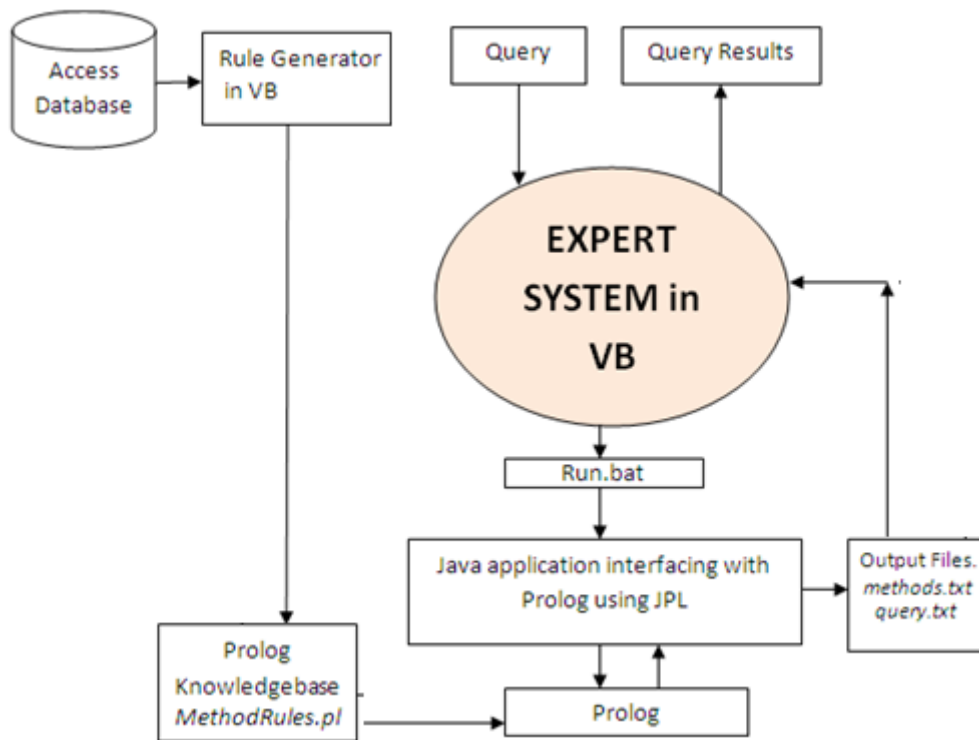


Figure 1: Structure of Manufacturing Database

We have developed an expert system in VB for selection of a manufacturing method based on a single objective. Figure 2 (a) and 2 (b) show the architecture of Expert System.

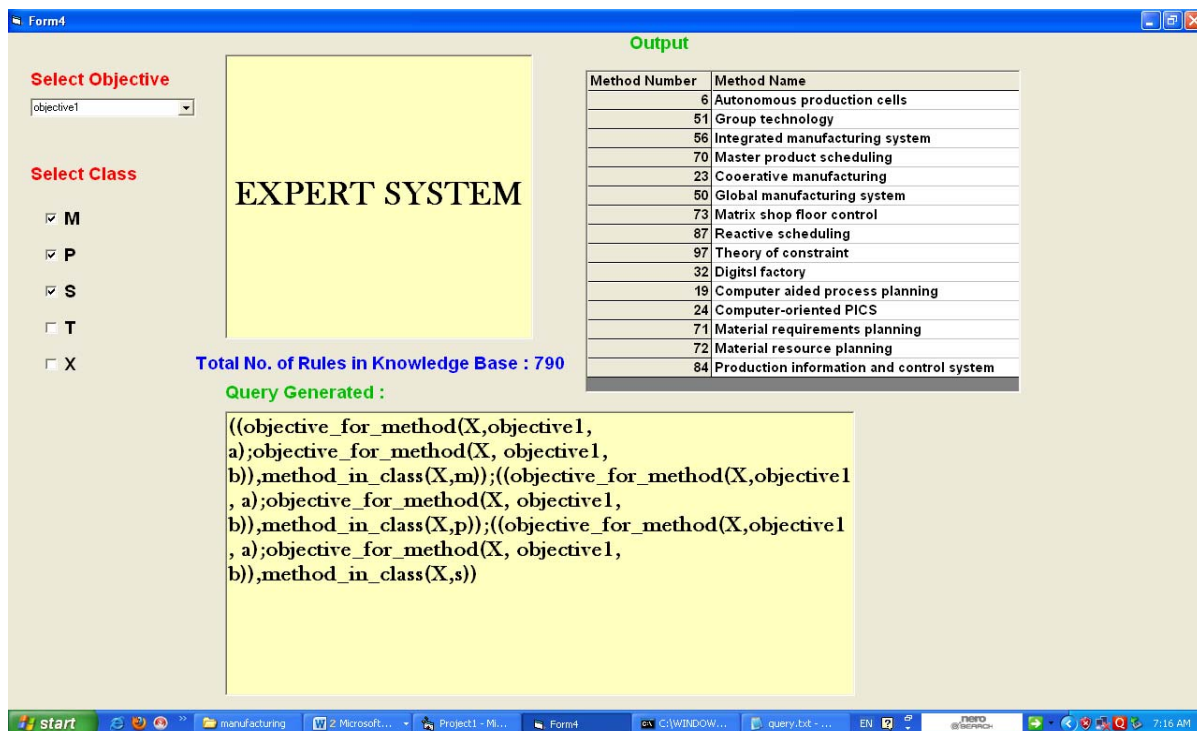


User Interface in Visual Basic
Java application interfacing with Prolog using JPL
Prolog
Prolog Database or Knowledgebase

Figure 2(a) – 2(b): Architecture of Expert System

An expert system shell with Graphical User Interface (GUI) is presented to the end user to select one of the manufacturing methods from the available alternatives using simple drag and drop operations. All the 16 methods and 5 classes are listed on the left of the grid as shown in the Figure 3 (a) and 3(b). The user can drag and drop any

required objective on to an expert system to filter the methods aiming at that objective. Further, the user can select one or more of the five class methods to widen table showing the filtered list of methods meeting objective1, i.e. Meeting Delivery Dates and belonging to the classes M, P, S and M, P, S, T, respectively. Out of 110 rows 15 rows are selected which belong to the classes M, P, and S and 17 rows are selected which belong to the classes M, P, S and T as shown in Figure 3(a) and 3(b), respectively.



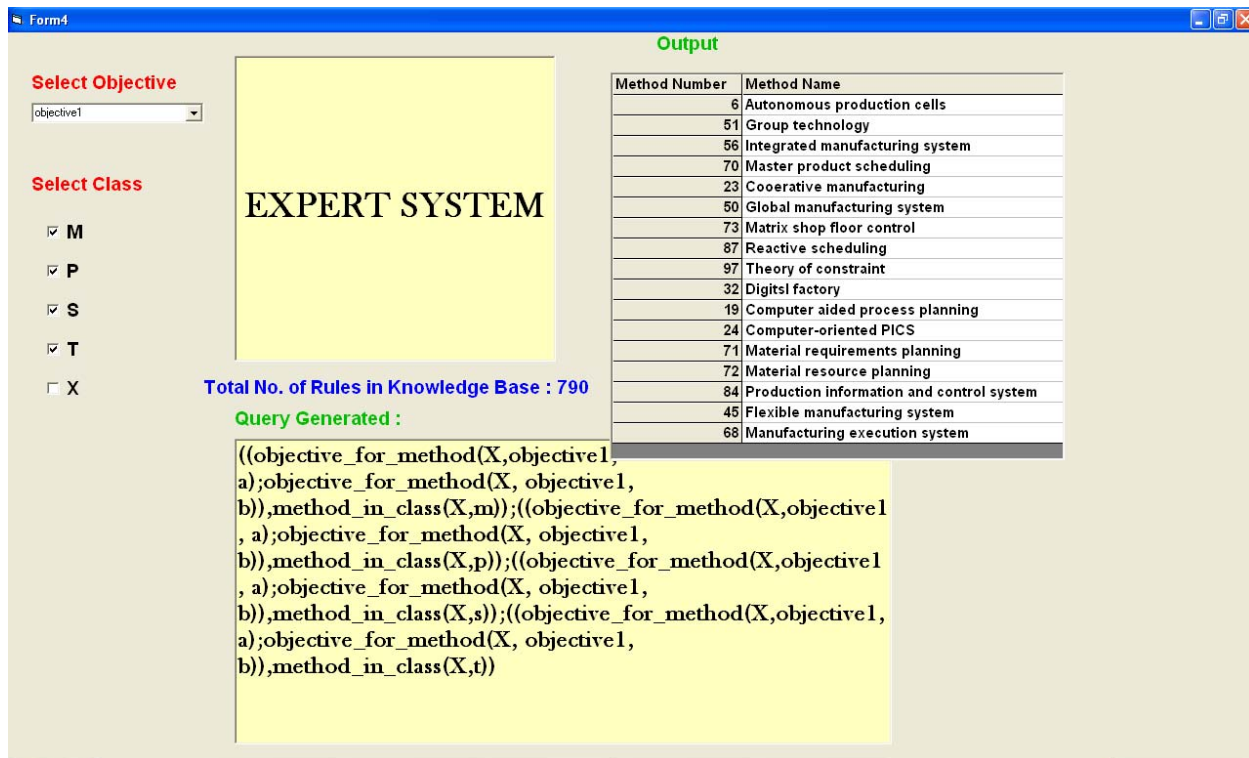


Figure 3(a)-(b): Expert System shell

Method and class rules are automatically generated by reading the data from the access database as shown in Figure 4.

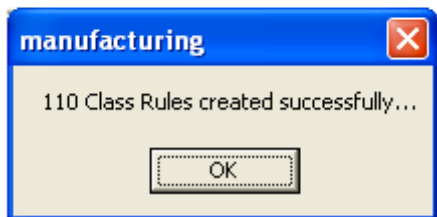
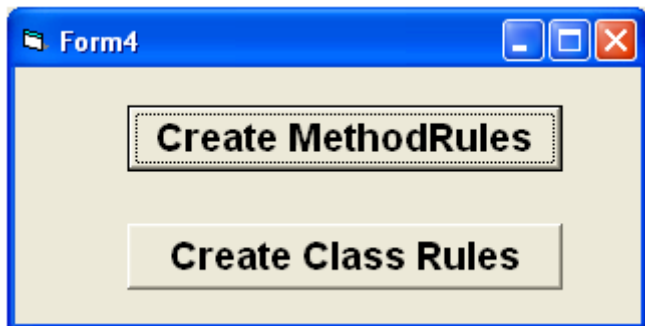


Figure 4: Automatic Rule Generation.

For the current domain and the problem under consideration 680 method rules and 110 class rules are generated and stored in a prolog database methodrules.pl. The format of the prolog facts are depicted below:

```
objective_for_method(method1, objective14, c).
objective_for_method(method1, objective2, c).
objective_for_method(method1, objective11, d).
objective_for_method(method1, objective7, c).
method_in_class(method1, s).
method_in_class(method2, m).
method_in_class(method3, m).
method_in_class(method4, x).
method_in_class(method5, p).
```

The format of the query for selection of manufacturing methods conforming to objective1 is as follows. If (objective1.grade=a OR objective1.grade=b) then select method.

Where, objective1.grade refers to the grade of objective1.

The equivalent prolog query is:

```
Selected_methods:=objective_for_method(X,objective1,a);
objective_for_method(X,objective1,b).
```

The format of the query for selection of manufacturing method conforming to objective2 and belonging to class M or S is as follows:

If ((objective1.grade=a OR objective1.grade=b) AND class=M) OR

((objective1.grade=a OR objective1.grade=b) AND class=S) THEN select method.

The equivalent prolog query is:

```
Selected_methods:=((objective_for_method(X,objective1,a);
objective_for_method(X,objective1,b)),method_in_class(X,
m));((objective_for_method(X,objective1,a);
objective_for_method(X,objective1,b)),method_in_class(X,s
)).
```

The expert system dynamically generates a batch file, run.bat, for setting the required environment variables, compiling the java application using jpl for interfacing with SWI prolog and passing the necessary command line arguments based on the objective and the classes selected.

The prolog database is queried for one or more solutions which are redirected to a text file, methods.txt. The format of methods.txt is shown below:

```
method26
method51
method56
method101
method44
method50
method53
method39
method68
```

The query is also generated on the fly based on the command-line arguments and stored in a text file, query.txt. The content of run.bat file is shown below:

```
set path=%path%;C:\Program Files\pl\bin
setclasspath=C:\Program Files\pl\lib\jpl.jar,.;c:\
javac.exe c:\Methods.java
java Methods objective2 m p s t
Pause
```

pl denotes the folder where SWI prolog is installed and bin folder contains the required Windows libraries. jpl.jar file contains the necessary java classes for interfacing with prolog. The structure of java program for executing prolog query is shown below:

```
String t1 = "consult('methodrules.pl')";
Query q1 = new Query(t1);
FileOutputStream fos=new
FileOutputStream("c:\\methods.txt");
FileOutputStream fos1=new
FileOutputStream("c:\\query.txt");
byte[] arr=new byte[20];
byte[] q=new byte[100];
```

```
String str;
System.out.println( t1 + " " +
(q1.hasSolution() ? "succeeded" : "failed"));
String str1 = "((objective_for_method(X,"
+ args[0] + ", a);objective_for_method(X, " + args[0] +
", b)),method_in_class(X, ";String str2 = "));";
String t2="";
for (int i=1;i<args.length-1;i++)
{
t2=t2+str1+args[i]+str2+";";
}
t2=t2+str1+args[args.length-1]+str2;
Query q2 = new Query(t2);
System.out.println( "first solution of " + t2
+ ": X = " + q2.oneSolution().get("X"));
q=t2.getBytes();
fos1.write(q);
//-----
java.util.Hashtable[] ss4 =
q2.allSolutions();
System.out.println( "all solutions of " +
t2);
for ( int i=0 ; i<ss4.length ; i++ ) {
System.out.println( "X = " +
ss4[i].get("X"));
str=ss4[i].get("X").toString()+"\r\n";
arr=str.getBytes();
fos.write(arr);
}
fos.close();
fos1.close();
```

The execution of run.bat file is shown in the Figure 5(a)-(b).

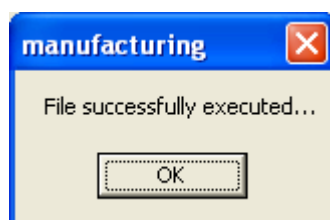
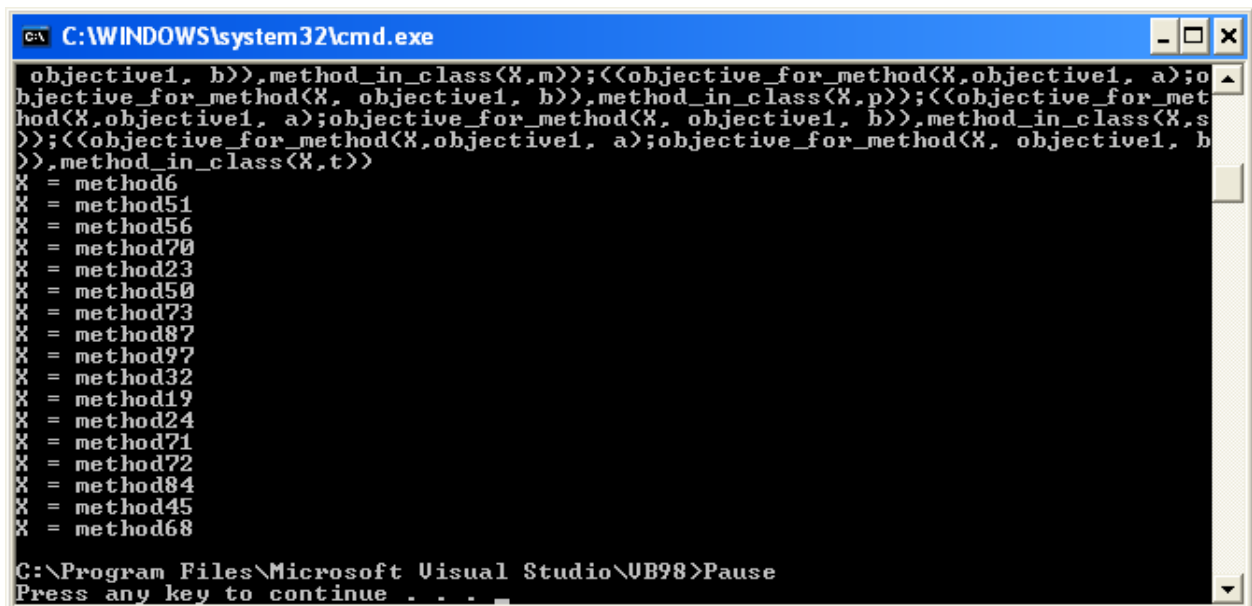


Figure 5 (a)-(b): Execution of run.bat.

The expert system developed in Visual Basic reads the contents of mehods.txt as computed by prolog and presents the results to an end user. The format of the auto generated prolog query generated is shown below:

```
((objective_for_method(X,objective2,
a);objective_for_method(X, objective2,
b)),method_in_class(X,m));((objective_for_method(X,object
ive2, a);objective_for_method(X, objective2,
b)),method_in_class(X,p));((objective_for_method(X,objecti
ve2, a);objective_for_method(X, objective2,
b)),method_in_class(X,s));((objective_for_method(X,objecti
ve2, a);objective_for_method(X, objective2,
b)),method_in_class(X,t))
```

5. Conclusion and Future Work

This paper presents the design of an expert system in Visual Basic (VB) which assists the manager in selection of a manufacturing method based on a single objective. Method and class rules are automatically generated by retrieving the data from the access database which are stored in a prolog database. The expert system shell is implemented in VB which interfaces with SWI Prolog with the help of java application using JPL library. The prolog query is generated on the fly based on the objective and the classes selected by the user in expert system shell which is evaluated using SWI prolog and the solutions are redirected to the text file. The expert system in Visual Basic reads the contents of these text files as computed by SWI prolog and presents the results to an end user. The methodology and application architecture is general and can be applied to similar problems. Our current work focuses on a single objective. It is an idealistic scenario where a single objective defines the section of manufacturing method. In real situations multi objective criteria is required for the purpose. Our future work involves modification of the expert system to take account of multiple objectives and functions.

References

- [1] Gideon Halevi, Handbook of Production Management Methods, Butterworth Heinemann publications, ISBN 0 7506 5088 5.
- [2] L. Mikhailov and M. G. Singh, "Fuzzy analytic network process and its application to the development of decision support systems," IEEE Transactions on Systems, Man, and Cybernetics, Part C. Applications and Reviews, Vol. 33, No. 1, pp. 33–41, 2003.
- [3] R. Santhanam and G. J. Kyparisis, "A multiple criteria decision model for information system project selection," Computers & Operations Research, Vol. 22, No. 8, pp. 807–818, 1995.
- [4] V. S. Lai, K. W. Bo, and W. Cheung, "Group decision making in a multiple criteria environment: A case using the AHP in software selection," European Journal of Operational Research, Vol. 137, No. 1, pp. 34–144, 2002. C. C. Wei, C. F. Chien, and M. J. J. Wang, "An AHP- based approach to ERP system selection," International Journal of Production Economics, Vol. 96, No. 1, pp. 47–62, 2005.

- [5] J. P. Brans, B. Mareschal, and P. Vincke, "PROMETHEE: A new family of outranking methods in multicriteria analysis," Operational Research, Vol. 3, pp. 477–490. 1984.
- [6] R. V. Rao, "Decision making in the manufacturing environment using graph theory and fuzzy multiple attribute decision making methods," Springer-Verlag, London, 2007.
- [7] R. Santhanam and G. J. Kyparisis, "A multiple criteria decision model for information system project selection," Computers & Operations Research, Vol. 22, No. 8, pp. 807–818, 1995.
- [8] Dhananjay R. Kalbandeand G.T.Thampi, Multi-attribute and Multi-criteria Decision MakingModel for technology selection using fuzzy logic, International Journal of Computing Science and Communication Technologies, VOL. 2, NO. 1, July 2009. (ISSN 0974-3375)
- [9] Journal of Micromechanics and Microengineering, Xuan F Zha and H Du, Manufacturing process and materialselection in concurrent collaborativedesign of MEMS devices, 13, 509–522, 2003.
- [10] ChenhuiShaoa, , Kamran Paynabarb, Tae HyungKima, Jionghua (Judy) Jinc, S. Jack Hua, J. Patrick Spicerd, HuiWangd, Jeffrey A. Abelld, Feature selection for manufacturing process monitoring using cross-validation, Journal of Manufacturing Systems, Volume 32, Issue 4, October 2013, Pages 550–555
- [11] R. V. RAO, T. S. RAJESH, Software Selection in Manufacturing Industries Using a Fuzzy Multiple Criteria Decision Making Method, PROMETHEE, Intelligent Information Management, 2009, 1, 159-165, December 2009
- [12] Mohammad Akhshabi, A New Fuzzy Multi Criteria Model for Maintenance Policy, Middle-East Journal of Scientific Research 10 (1): 33-38, 2011