A Self-Developing Fuzzy Expert System, Designed for Optimization of Machining Process

Asif Iqbal, Iqbal Khan, Naeem Ullah Dar, and Ning He

Abstract—**The paper presents a fuzzy expert system that possesses the ability to self-learn, self-correct and self-expand. These features embellish the expert system with the capability to cope effectively with the ever-changing industrial environment. The system primarily provides the optimization of the predictor parameters of a process, with respect to the desired objectives, and prediction of the process's performance measures based upon settings of the predictor parameters. The self-development mode of the system consists of modules possessing following characteristics: automatic storage/retrieval of data; automatic development of fuzzy sets; automatic generation of fuzzy rules for optimization and prediction rule-bases; conflict resolution among contradictory rules; and automatic updating of expert system interface. The presented expert system finds high degree of applicability in optimization of manufacturing processes in general, and machining in particular.**

Index Terms— **AI application, Automated reasoning, Fuzzy sets, Optimization**

I. Introduction

The foremost shortcoming of expert system technology is the lack of dynamism. The price paid by this inadequacy is its failure in coping with fast changing environments. This describes the main reason of inability of expert system technology to find its full fledge application at industrial levels.

The knowledge-base is considered as the heart of an expert system and it is predominantly important to keep it updated and corrected all the time in order to maintain the efficacy of the expert system. Obviously, it is not prudent to engage the services of a knowledge engineer for this purpose, as the efficiency calls for rapid upgrading of the knowledge-base. The dire need is to find the means for automatic accomplishment of this requirement.

Most of the time required for the development of expert system can be attributed to the knowledge acquisition [1]. Researchers have put forward variety of knowledge acquisition methods for automatic learning of the expert systems. In [2] authors suggested that two basic anomalies of expert system are incompleteness and incorrectness. They suggested that by integrating machine learning (ML) techniques in the validation step of the expert systems' evolutionary life cycle model, the knowledge-base can be refined and corrected throughout a refinement cycle. Researchers in [3] presented an adaptive fuzzy learning algorithm, used for nonlinear system identification that provided a new way for representing the consequent part of the production rule. In [4], a knowledge factory has been proposed that allows the domain expert to collaborate directly with ML system without needing assistance of a knowledge engineer. In [1] the authors have presented an inductive learning algorithm that generates a minimal set of fuzzy IF-THEN rules from a set of examples. In [5], the authors presented a self-testing and self-learning expert system, which is based upon fuzzy certainty factor and it checks the rule-base for correctness and completeness.

Few papers can be found that describe the application of machine learning in manufacturing domain. In [6] researchers presented a fuzzy expert system for design of machining conditions dedicated to the turning process. The learning module contained by the system used to correct the empirical relationships by changing the fuzzy membership functions. The researchers in [7] presented an approach for building the knowledge-base from the numerical data, which proved to be useful for classification purposes. In [8] the authors utilized Support Vector Regression, a statistical learning technique, to diagnose the condition of tool during a milling process. In [9], the authors presented the comparison of three ML algorithms for the purpose of selection of the right dispatching rule for each particular situation arising in flexible manufacturing system.

In the current research work a fuzzy expert system has been presented that not only self-learns and self-corrects but also self-expands. Following are the distinguishing features of the self-developing expert system:

1. Predicts the values of output variables based upon values of input variables.
2. Suggests the best values of input variables that maximize and/or minimize the values of selected set of output variables.
3. Automatically adjusts newly entered variable at any stage of development.
4. Learns and corrects itself according to new set of data provided.
5. Automatically generates fuzzy sets for newly entered variables and regenerates sets for other variables according to newly added data.
6. Automatically generates rules for optimization and prediction rule-bases.

Manuscript received January 11, 2007.

A. Iqbal is with the Department of Mechanical Engineering, University of Engineering & Technology, Taxila, Pakistan (Phone: +92-302-8548307; fax: +92-51-4830532; e-mail: asif.asifiqbal@gmail.com).

I. Khan is with the Department of Mechanical Engineering, University of Engineering & Technology, Taxila, Pakistan (Phone: +92-300-8540976; fax: +92-51-4830532; e-mail: iqbalkhan1950@yahoo.com).

N.U. Dar is with the Department of Mechanical Engineering, University of Engineering & Technology, Taxila, Pakistan (Phone: +92-300-8544016; fax: +92-51-4830532; e-mail: nudar@hotmail.com).

N. He is with the College of Mechanical & Electrical Engineering, Nanjing University of Aeronautics & Astronautics, Nanjing, China (Phone: +86-25-84892551; fax: +86-25-84892551; e-mail: drnhe@nuaa.edu.cn).

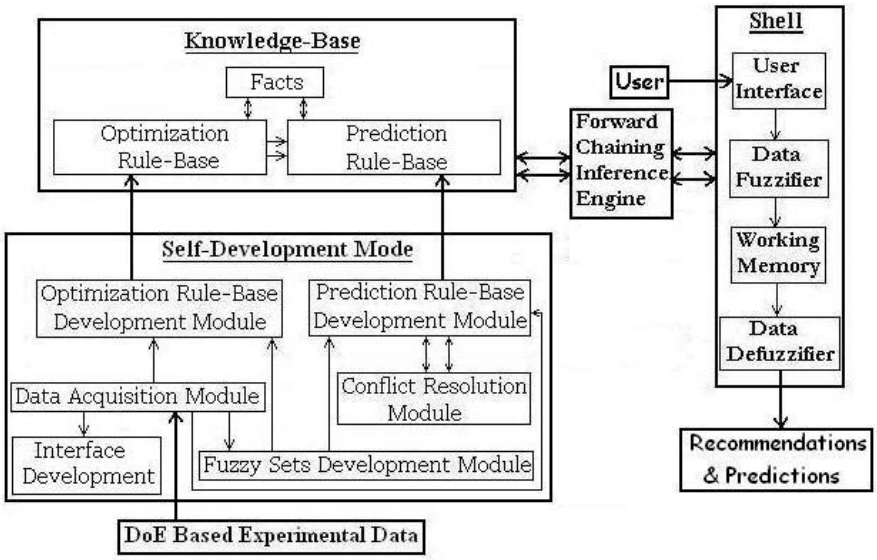

Fig. 1. The configuration of self-developing expert system

7. Provides conflict resolution facility among contradictory rules.
8. Updates interface of the expert system according to newly entered variables.

II. The Configuration

The configuration of the self-developing expert system has been presented in the Fig. 1. The main purpose of the system is to self-generate the rule-bases, from the experimental data, that could be used to: (1) optimize the settings of predictor variables (of any physical process) for the purpose of maximization and/or minimization of set of selected response variables; (2) predict the values of response variables according to the finalized settings of the predictor variables.

Pattern of the system consists of four parts: the knowledge-base, the shell, the inference engine, and the self-development mode. The knowledge-base is the combination of facts, the optimization rule-base, and the prediction rule-base. The functional details of optimization and prediction rule-bases can be read from [10]. The shell consists of the user interface through which the input is taken from the user. The data fuzzifier fuzzifies the values of numeric predictor variables according to the relevant fuzzy sets. The user-interface and the fuzzy sets are also automatically developed by the system itself.

For development of the knowledge-base consisting of two rule-bases, the inference mechanism of forward chaining shell, Fuzzy CLIPS (C Language Integrated Production Systems) was used. See details of the shell in [11].

III. The Self-Development Mode

The most important and distinguishing constituent of the system is the self-development mode. This mode consists of following four modules: data acquisition module; fuzzy sets development module; optimization rule-base development module; and prediction rule-base development module integrated with the conflict resolution sub-module. The detail is as follows.

A. Data Acquisition

This module facilitates the automation of intake, storage, and retrieval of data.

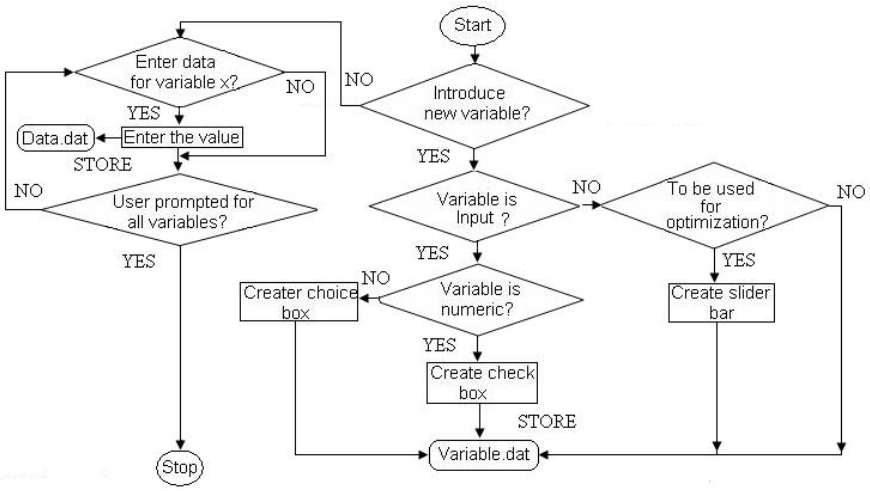

Fig. 2. The flow chart of data acquisition module

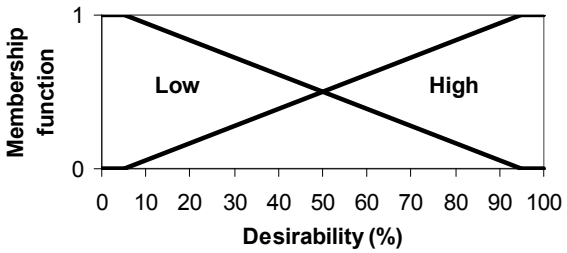

Fig. 3. Fuzzy sets for maximizing/minimizing output variable

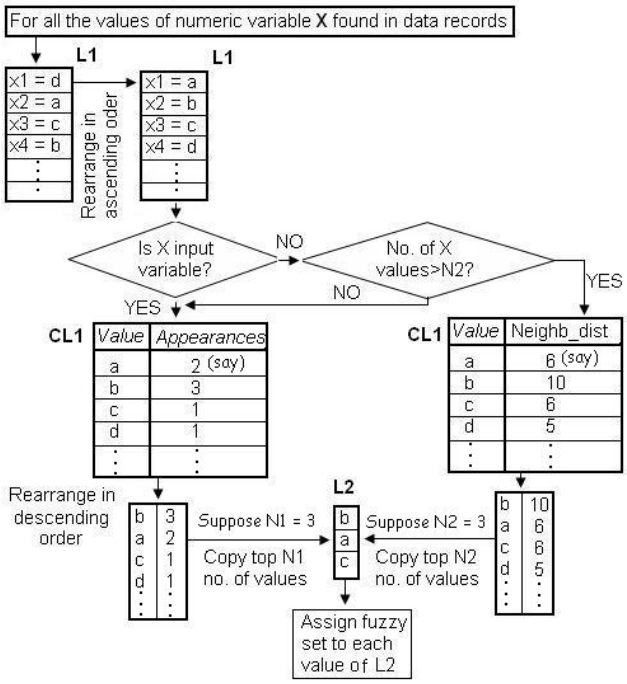

Fig. 4. Customized flow chart for self-development of fuzzy sets

The data could be the specifications of a new variable or the values of input and output variables obtained from experiments. Data related to specifications of a new variable are stored in file Variable.dat, while that related to the new values/records are stored in Data.dat on hard disk. Fig. 2 shows the flow chart of data acquisition algorithm.

B. Self-Development of Fuzzy Sets

This module covers three areas: (1) Rearranging the fuzzy sets for already entered variables according to the newly entered data records; (2) Development of fuzzy sets for newly entered numeric variables; and (3) Development of two fuzzy sets (*low* & *high*) for each output variable that is included for optimization purpose. The set *low* represents the minimization requirement and the other one represents maximization. The design of sets for category 3 is fixed and is shown in Fig. 3, while design of first two categories is dynamic and based upon the data values of respective variables. The desirability values shown in Fig. 3 are set by the user using the slider bar available on interface of the expert system. Any value below 5% means desirability is of totally minimizing the output variable (performance measure), and total desirability of maximization is meant if the value is above 95%. The desirability of 50% means optimization of that output variable makes no difference.

Fig. 4 shows the customized flow chart for the methodology used for self-development of fuzzy sets. The user has to decide the maximum allowable number of fuzzy sets for input as well as for output variables.

The logic involved in methodology is that a value of input variable, which has higher frequency of appearance in the data records, has more right to be picked up for allocation of a fuzzy set, while for output variable any value having greater difference from its previous and next values in the list – termed as Neighbor Distance (*Neighb_dist* in Fig. 4) – possesses more right for allocation of a fuzzy set. Neighbor Distance can mathematically be represented as follows:

$$Neighbor\_Distance = \begin{cases} Value[i+1] - Value[i]; & if\ (i = first) \\ Value[i] - Value[i-1]; & if\ (i = last) \\ \frac{1}{2}\left(Value[i+1] - Value[i-1]\right); & otherwise \end{cases} \quad (1)$$

C. Self-Development of Prediction Rule-Base

This step consists of following two parts: (1) automatic development of rules, for prediction of the manufacturing process's performance measures, based upon the data records provided by the users and (2) conflict resolution among self-developed contradictory rules.

Fig. 5 provides the graphical description of the first part. The objective is to convert each node of 2-D linked list *Data_output* (including list of related values of input variables *Data_input)* into a rule. 2-D linked list is a list that expands in two directions as shown in Fig. 5. The objective is achieved by finding and assigning the most suitable fuzzy sets for all of the values involved per node of *Data_output*. The list *Data_output* is navigated from first node to last and for all of its values the closest values in fuzzy sets of respective variables are matched. If the match is perfect then certainty factor (*CF*) of 1 is assigned to the match of the data value and the fuzzy set. If the suitable match of any fuzzy set for a given data value is not found then the data value is assigned the intersection of two closest fuzzy sets. All the rules are stored in a 2-D linked list named *Rule_Consequent*, each node of whose represents a rule. Each node contains the assigned fuzzy set of output variable and also a linked list (*Rule_antecedent*) containing assigned fuzzy sets of all the relevant input variables.

*Conflict Resolution among Contradictory Rules***.** There is always a possibility that some anomalous data might be entered by the user that could lead to self-development of some opposing rules. So it is necessary to develop a mechanism that would detect such possible conflict and provide a way for its resolution.

The mechanism of conflict resolution can be described as follows: Compare each and every rule of the prediction rule-base to all the other rules of the same rule-base. If, in the consequent parts of any two rules, following two conditions satisfy: (1) output variables are same; and (2) assigned fuzzy sets are different, then check whether the antecedent parts of both the rules are same (i.e., same input variables with same

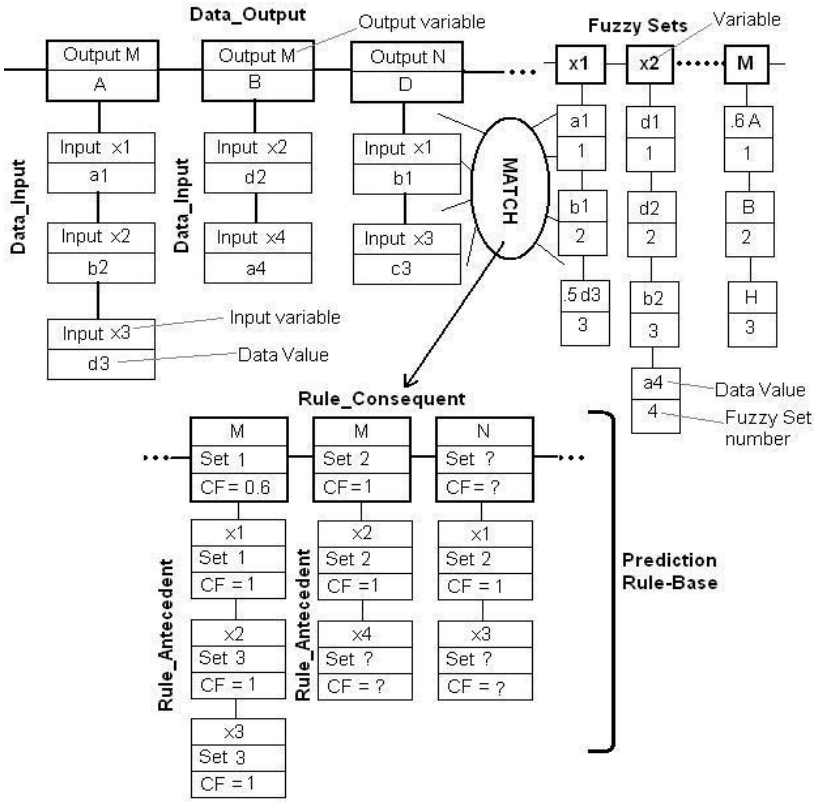

Fig. 5. Framework for self-development of prediction rule-base

fuzzy sets assigned). If yes, then these two rules form a pair of contradictory rules. Next the user is inquired which one of the two contradictory rules needed to be abandoned. The *CF* value of the rule to be abandoned is set to zero. The same procedure is continued for whole of the rule-base. At the completion of the process, all the rules possessing the *CF* values greater than zero are printed to the CLIPS file.

D. Self-Development of Optimization Rule-Base

This module generates the set of rules that is responsible for providing the optimal settings of input variables that would best satisfy the maximization and/or minimization of the selected output variables. Fig. 6 describes the framework.

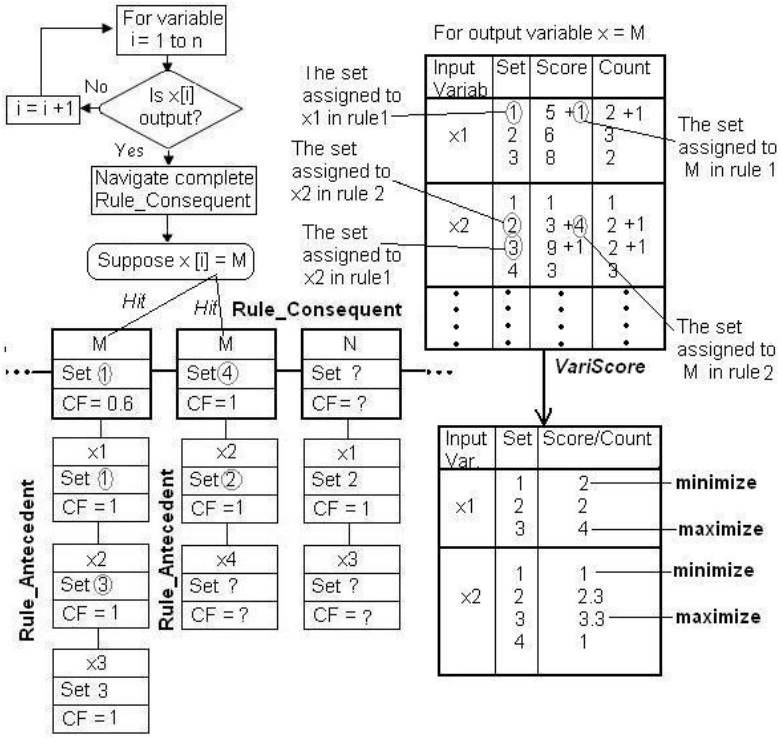

Fig. 6. Framework for self-development of optimization rule-base

Table 1. Limited experimental data for rookie knowledge-base

No.	Predictor Variables			Response Variables	
	Speed (m/min)	Rake (°)	Orientation	Tool life (mm^2)	Cutting force (N)
1	150	–8	Up	3601	910
2	150	–8	Down	5500	643
3	150	5	Up	3583	885
4	150	5	Down	4912	532
5	250	–8	Up	2991	1008
6	250	–8	Down	4004	769
7	250	5	Up	2672	986
8	250	5	Down	3609	704

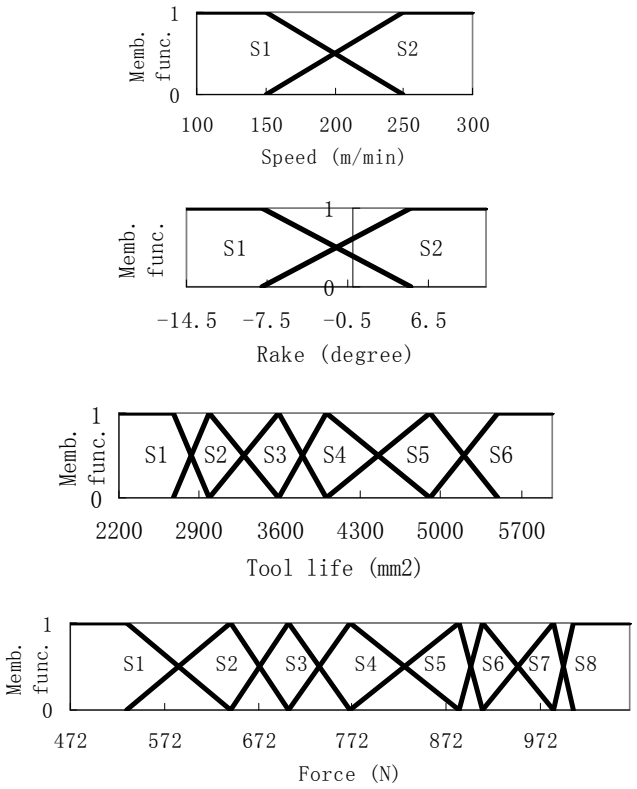

Fig. 7. Self-developed fuzzy sets for numeric variables

The idea exploited in development of this module is that for maximization of any output variable select an ideal fuzzy set for each numeric input variable, which, on average, would generate the maximum value of that output variable. For the minimization purpose, select those fuzzy sets for respective input variables that would result in the least possible value of the output variable, available in the data records.

IV. Application in Optimizing Machining Process

Machining is the most wide-spread of all the manufacturing processes and amount of investment being done in it is an indication of wealth of the nations [12].

For demonstration of applicability of the self-developing expert system in optimization and prediction of the processes included in the manufacturing domain, the process of milling (a machining process in which tool rotates and workpiece remains stationary) has been chosen. Table 1 presents limited data regarding milling process. The first three variables, namely, speed, rake, and orientation (milling-orientation) are predictor (input) ones, while the other two, tool life and cutting force are response (output) variables.

Table 2. Self-developed prediction rule-base

Rule No.	Antecedents			Consequents			
	Speed	Rake	Orientation	Tool life	CF	Force	CF
1	S1	S1	Up	S3	0.88	S6	1
2	S1	S1	Down	S6	1	S2	1
3	S1	S2	Up	S3	1	S5	1
4	S1	S2	Down	S5	1	S1	1
5	S2	S1	Up	S2	1	S8	1
6	S2	S1	Down	S4	1	S4	1
7	S2	S2	Up	S1	1	S7	1
8	S2	S2	Down	S3	0.82	S3	1

If the knowledge-base is developed based entirely upon the data presented in the table, it is very likely that the expert system may provide anomalous results because of the fact that the other influential milling parameters have not been taken care of, and thus the self-developed knowledge-base can be termed as "Rookie Knowledge-Base".

Suppose the expert system is asked to develop its rule-bases and update its interface based upon the data provided and it is also asked to include tool life, but not the cutting force, as output variable for optimization. Fig. 7 shows the detail of triangular fuzzy sets of numeric variables (input + output), developed itself by the expert system, in addition to the sets of the objective, as shown in Fig. 3.

Following is the detail of six rules, self-developed by the self-development mode and to be operated by the optimization module of the expert system:

Rule 1: IF *Objective Tool_life* is High AND *Speed* is not fixed THEN *Speed* is S1.

Rule 2: IF *Objective Tool_life* is High AND *Rake* is not fixed THEN *Rake* is S1.

Rule 3: IF *Objective Tool_life* is High AND *Orientation* is not fixed THEN *Orientation* is Down.

Rule 4: IF *Objective Tool_life* is Low AND *Speed* is not fixed THEN *Speed* is S2.

Rule 5: IF *Objective Tool_life* is Low AND *Rake* is not fixed THEN *Rake* is S2.

Rule 6: IF *Objective Tool_life* is Low AND *Orientation* is not fixed THEN *Orientation* is Up.

Out of these six rules the first three perform the maximization operation, while the others perform minimization. Table 2 presents the detail of eight rules, self-developed by the expert system and to be operated by its prediction module.

Fig. 8 shows the interface of the expert system related to the rookie knowledge-base. The slider bar, shown at middle of the figure, prompts the user whether to maximize or minimize the selected output variable and by how much desirability. Suppose the expert system is provided with following input:

- Objective: maximize tool life with desirability of 98%
- Rake angle of tool prefixed to 0 degree.
- Cutting speed and milling-orientation: *open* for optimization.

Pressing the *Process* button starts the processing and following results are displayed in the information pane:

- The recommended orientation is down-milling cutting speed is 154.2 m/min.
- The predicted tool life is 4526.4 mm^2 and cutting force is 649.68N

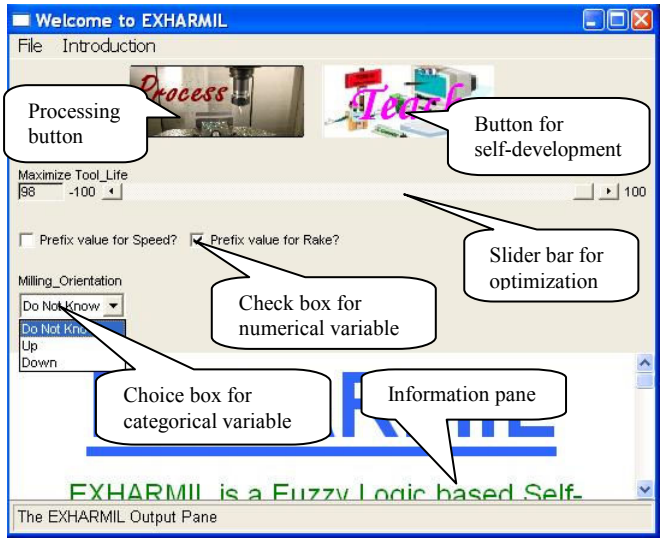

Fig. 8. Self-developed interface of expert system utilizing rookie knowledge-base

Fig. 9. Self-developed interface of expert system utilizing veteran knowledge-base

Suppose the same expert system is provided with more experimental data, covering effects of all the influential parameters of milling process. When the expert system is asked to develop knowledge-base from that set of data, the resulting knowledge-base would be a veteran knowledge-base. As more and more data will be provided to the expert system it will keep improving its accuracy of optimization and prediction processes.

Fig. 9 presents the interface of the expert system from the experimental data provided in papers [13, 14] in addition to that provided in table 1.

V. Conclusion

This paper presents a unique approach for designing mechanism of a novel self-developing expert system. The system possesses abilities to manage new variables, to self-develop fuzzy sets, to self-generate rules for optimization and prediction modules, to resolve the conflict among contradictory rules, and to keep its interface updated. The discussion shows the distinctiveness of the presented expert system along with its high degree of applicability to process of optimizing the machining process in particular and manufacturing in general. Brisk and automatic development of knowledge-base makes it well adaptable to ever-changing industrial environment.

References

[1] J.L. Castro, J.J. Castro-Schez, and J.M. Zurita, "Use of a fuzzy machine learning technique in the knowledge-acquisition process" *Fuzzy Sets & Syst.*, vol.123, 2001, pp.307–320

[2] H. Lounis, "Knowledge-based systems verification: A machine-learning based approach" *Expert Syst. Appl.*, vol. 8(3), 1995, pp.381–389

[3] K.C. Chan, "A comparative study of the MAX and SUM machine-learning algorithms using virtual fuzzy sets" *Eng. Appl. Artif. Intell.*, vol. 9(5), 1996, pp.512–522

[4] G.I. Webb, "Integrating Machine Learning with knowledge-acquisition through direct interaction with domain experts" *Knowl-Based Syst.*, vol. 9, 1996, pp.253–266

[5] A. Lekova and D. Batanov, "Self-testing and self-learning fuzzy expert system for technological process control" *Computers in Industry.*, vol. 37, 1998, pp.135–141

[6] Y. Chen, A. Hui, and R. Du, "A fuzzy expert system for the design of machining operations" *Int. J. Mach. Tools Manuf.*, vol. 35(12), 1995, pp.1605–1621

[7] B. Filipic and M. Junkar, "Using inductive machine learning to support decision making in machining processes" *Computers in Industry.*, vol. 43, 2000, pp.31–41

[8] S. Cho, S. Asfour, A. Onar, N. Kaundinya, "Tool breakage detection using support vector machine learning in a milling process" *Int. J. Mach. Tools Manuf.*, vol. 45, 2005, pp.241–249

[9] P. Priore, D. de la Fuente, J. Puente, and J. Parreno, "A comparison of machine learning algorithms for dynamic scheduling of flexible manufacturing systems" *Eng. Appl. Artif. Intell.*, vol. 19, 2006, pp.247–255

[10] A. Iqbal, N. He, L. Li, and N.U. Dar, "A fuzzy expert system for optimizing parameters and predicting performance measures in hard-milling process. *Expert Syst. Appl.*, vol. 32(4), 2007, pp.1020-1027

[11] R.A.Orchard, *Fuzzy CLIPS, V6.04A; Users' Guide*. NRC, Canada, 1998

[12] T. Childs, K. Maekawa, T. Obikawa, and Y. Yamane, "Metal Machining: Theory and Applications". Ed. London: Arnold Publishers, 2000

[13] A. Iqbal, N. He, L. Li, W.Z. Wen, and Y. Xia, "Influence of tooling parameters in high-speed milling of hardened steels" *Key Eng. Mater.* (*Advances Machining & Manuf. Tech.* 8)., vol. 315-316, 2006, pp.676-680

[14] A. Iqbal, N. He, L. Li, Y. Xia, and Y. Su, "Empirical modeling the effects of cutting parameters in high-speed end milling of hardened AISI D2 under MQL environment". *Proc. 2nd CIRP Conf. High Performance Cutting*, Vancouver Canada, 2006