00 52 12 04 11 07

Closing the Process Control Expert System Gap

by
Donald K Frerichs
Chief Application Engineer
Bailey Controls Company
Wickliffe, Ohio U S A

Presented at
IBRA
Liege, Belgium
November 23, 1989

Technical Paper

TP89-19

Bailey

CLOSING THE PROCESS CONTROL EXPERT SYSTEM GAP

By:
Donald K. Frerichs
Chief Application Engineer
Bailey Controls Company
Wickliffe, Ohio, U.S.A.

ABSTRACT

Expert systems have long caught the imagination of process control engineers. Yet today, there are very few successful real-time process control expert systems. This paper will review the reasons for past failures and offer a solution for implementing successful process control expert systems that exploits the power of a modern distributed control system.

INTRODUCTION

Over the past 5 years, we have witnessed the first expert systems evolve. The science of expert systems has moved from the academic world into the industrial world. This movement has been fostered by a few very successful applications, as well as the development of powerful desktop computers. Yet, the examples of successful real time expert systems remains as only a handful [1] [2] [3].

This "process control expert system gap' has not occurred due to lack of interest. Process control engineers, by their very training, seek ways to explore new technology. They have envisioned numerous applications where an expert system could be used to capture the heuristics of their particular process. They have recognized that separating the knowledge base from the program flow will yield time savings in creating their application and monumental savings in maintaining the application. Finally, they have recognized that an expert system approach can solve large problems that would test the ingenuity and patience of even the best procedural language programmer.

Certainly, process control expert systems have several unique requirements above and beyond the requirements for stand-alone expert systems [4]. Most important of these is the ability of the expert system to execute in real time for the given process. Usually, this means execution times under 0.5 seconds for any process condition, including major process upsets. Included in the execution time is the time to access plant parameters and to send data to the process base regulatory and sequential controls.

ROADBLOCKS TO SUCCESSFUL IMPLEMENTATION

As mentioned, it is not lack of motivation that has caused the "gap". I suggest that it has been caused by a small set of roadblocks, all of which can (and must) be overcome to achieve a successful real time process control expert system application.

The chance of a given expert system application succeeding can be determined from the following formula.

$$COS = f_a * (1-f_r) * f_{ic} * f_{dc} * f_{kc}^2$$

COS is the chance of success, and the remaining terms are probability factors ranging between 0 and 1. The individual factors will be explained in the following sections.

The first roadblock is the choice of a good application. Modern Distributed Control Systems (DCS's) have evolved to a high degree of power and sophistication. Engineering Workstation (EWS) tools exist to facilitate the configuration of DCS modules in a variety of languages (eg: function blocks, ladder logic, specialized batch control languages, BASIC, C language, etc.). With this wealth of tools, just about any control problem can be solved. The correct application for a process control expert system must be one that will be difficult or inconvenient to solve with the existing tools, yet an expert does exist with knowledge that can be brought to bear on the problem. f_a is the probability that a good application has been chosen. Unless sufficient thought is given to this area, f_a could easily be less than 0.80.

The next roadblock to overcome is end user reluctance. Several expert systems have been both technical successes and practical failures simply because the intended users didn't use them. That reluctance might be based on fear or a lack of discipline, but the fact remains that the end user must accept the system. In the process control world, the end user is typically the process operator. Imposing an 'expert" system that uses 'artificial intelligence may be a large leap of faith for many process operators. A successful process control expert system must be unimposing to the primary end user. f_r then is the probability that there will be end user reluctance to use the new application. Depending on the application and method of implementation, f_r could easily exceed 0.25.

The next roadblock is the initial cost to a corporation to begin to use expert system technology. Usually, expensive software development tools, which may require the purchase of an expensive minicomputer and associated specialized training courses [5]. These development items can cost as much as $100,000, which can kill a potential application before it even begins. f_{ic} then is the probability that the initial cost of expert system development tools will be accepted by your corporation. The cost of most process control expert system development tools can cause f_{ic} to approximate 0.75.

Yet another roadblock is the cost to deploy the developed expert system. This can include another minicomputer for the runtime version of the application, interface hardware and software to pass real time data to and from the expert system, and potentially a royalty fee to the vendor of the development software. Deployment costs are often underestimated during the planning portion of a process control expert system project. Thus, f_{dc} is the probability that the complete deployment costs will be accepted by your corporation. Assuming that your company has accepted the initial cost of development tools, f_{dc} usually approximates 0.85.

THE FINAL ROADBLOCK

The final roadblock, and usually the most difficult to overcome, is that of knowledge engineering. All too often, the individual with the process knowledge (domain expert) is not the same person who will commit the knowledge into the knowledge base of the process control expert system. When true, the complexity of translating the knowledge from one person to another, before it is encoded, is added to the knowledge engineering task. Depending upon the choice of development software, the resulting knowledge base may not be directly readable by the domain expert, thereby requiring another translation. Such a single or double translation environment in the critical area of knowledge engineering is not conducive to quick success in building the knowledge base.

A second aspect of the knowledge engineering roadblock is the capabilities of the development/deployment software package. If it does not contain the features required for process control (eg: speed of execution, ease of real time data transfer, concepts of time, debugging and monitoring aids, etc.), the knowledge engineer may have to resort to unusual methods to bypass the limitations of the package. On the other hand, the features of the development/deployment package may be too robust. This can lead to long training times for the knowledge engineer, and corresponding long initial development times.

f_{ke}, the knowledge engineering factor, is the probability that the knowledge engineering task can be accomplished quickly and efficiently. Due to the critical importance of the knowledge engineering portion of implementing a process control expert system, it is weighted twice as much as the other factors in the COS equation. Experience shows that f_{ke} usually runs about 0.80.

OVERCOMING THE ROADBLOCKS

Using the formula and typical values for the individual factors presented above, the chance of success averages about 25% for most process control expert system projects. The reader may well wonder if it is ever possible to overcome all of the roadblocks. As mentioned earlier, there are a few successful process control expert systems in existence. All of them benefited from a patient paternalistic management structure. Patient from the standpoint that development and deployment time was not critical, paternalistic from the standpoint that total investment cost (development plus deployment costs) was not a major consideration. Unfortunately, most of us work in very competitive industries that do not have this luxury. Thus, we must either overcome the obstacles, forego development, or worst of all fail. The balance of this paper will offer suggestions on how to succeed where others have failed.

CHOOSE THE RIGHT APPLICATION

This is usually the easiest roadblock to overcome. A good rule of thumb to follow is to choose an application problem where the solution set is either imprecise, not common knowledge, the application is one of interpretation, or the tools available to solve the problem are considered inadequate. This means that applications like motor control, automatic sequencing, and conventional regulatory controls usually are not good candidates due to the preciseness of the solution and the existence of good tools to implement the solution. However, applications like alarm interpretation, sequential or regulatory control involving multiple possible conclusions can be good applications. Another class of good applications is control of a process where it is clear that there are good and not so good operators (ie: control is an art, not a science).

A second aspect of choosing the right application is determining if there is a recognized expert who can be made available to the project. This may be the plant engineer, your best operator, etc. Alternatively, the domain expertise can come from literature such as regulations or process hazard studies. In either case, the knowledge engineer must have convenient access to the domain expertise.

The best of all worlds is where the domain expert is the knowledge engineer. This is possible if the development package is designed for direct use by people like a plant engineer.

By choosing a good application, $\mathbf{f_a}$ can be greater than 0.95.

END USER RELUCTANCE

Many process operators have an innate fear or distrust of new technology. Installing an additional computer with it's CRT screen and keyboard for a process control expert system can easily increase the reluctance factor $\mathbf{f_r}$. Usually, the best implementation for a process control expert system is one that is as invisible as the traditional regulatory process controls. Such an implementation is practical using an embedded architecture. That is to say, an architecture where the expert system and traditional controls reside in standard DCS modules (see Figure 1). Thus, the operator interface will use the same techniques and devices. EXPERT-90 is a product which allows the creation of embedded process control expert systems. Using an embedded architecture will reduce $\mathbf{f_r}$ below 0.10.

INITIAL AND DEPLOYMENT COSTS

As stated earlier, some process control expert system development packages can cost as much as $100,000 when all of the initial costs are considered. By contrast, the embedded architecture approach allows the knowledge engineer to use the Engineering Workstation hardware that is used to configure and tune the DCS controls for development of his knowledge base. The incremental cost of software and training for EXPERT-90 is minimal. These features raise $\mathbf{f_{ic}}$ above 0.95.

Because the embedded architecture uses standard low-cost DCS modules, deployment costs are minimal. Indeed, it may be found that the process control expert system can be implemented in spare capacity of existing DCS modules. Because the resulting process control expert system resides in the DCS module, the embedded approach obviates the need for special data interfacing hardware and software. In the case of EXPERT-90, no royalty fees are required for resulting applications. These features raise $\mathbf{f_{dc}}$ above 0.95.

KNOWLEDGE ENGINEERING

From the discussion of the knowledge engineering roadblock, it is clear that the choice of the development/deployment package has a large influence on $\mathbf{f_{ke}}$. EXPERT-90 is a development package specifically designed for process control applications, and for direct use by plant engineering personnel.

The English like rule structure of EXPERT-90 coupled with it's intelligent context sensitive menu driven editor practically eliminates the knowledge translation problem. Domain experts can read the resulting knowledge base directly, without the need for specialized training. The language of EXPERT-90 allows for rules that deal with time (ie: FOR n seconds, AFTER n seconds, etc.) as well as handling the functions normally associated with process control. Finally, EXPERT 90 facilitates the use of uncertainty in rules by incorporating fuzzy logic concepts. Figure 2 contains an extract from a sample EXPERT-90 knowledge base to illustrate the clarity of the rules. Figures 3 and 4 illustrate the ease with which rules can be created with the EXPERT-90 context sensitive editor.

EXPERT 90 contains features to allow execution of the knowledge base in real time (usually 0.25 seconds), even during major process upsets. Because the resulting expert system will execute in standard DCS modules, no unusual methods are required to achieve real time data transfer or monitoring of the developed system.

Together, the features of EXPERT 90 significantly raise the knowledge engineering factor $\mathbf{f_{ke}}$ above that of other process control expert system development packages. Experience has shown that it produces an $\mathbf{f_{ke}}$ of approximately 0.95.

CONCLUSION

While there are very few existing successful process control expert systems, the elements required to achieve success can be identified and analyzed. Tools and techniques now exist to maximize the chance of success in creating new process control expert systems. Using the suggestions presented in this paper will raise the overall chance of success (**COS**) from 25% to greater than 70%.

REFERENCES

1. Grainger, B.E. and Tomasello, M.L., "Expert system interface to a real time data monitoring and control system," Proceedings of the ISA/88 International Conference, 1988.

2. Martin, R.P. and Nassersharif, B., "CATALisp - an expert system for real time diagnosis of multiple-failure transients in nuclear power plants," Proceedings of the 1988 National Heat Transfer Conference, 1988.

3. Jakob, F. and Vernet, D., "Application of expert systems to process control," Journal of the 1986 International Conference on Intelligent Manufacturing Systems, 1987.

4. Chantler, M.J., 'Real time aspects of expert systems in process control," Digest of the IEE Colloquium on Expert Systems in Process Control, 1988.

5. Koch, P., "Some problems of building process control expert systems," Proceedings of the International Conference on Intelligent Autonomous Systems, 1987.

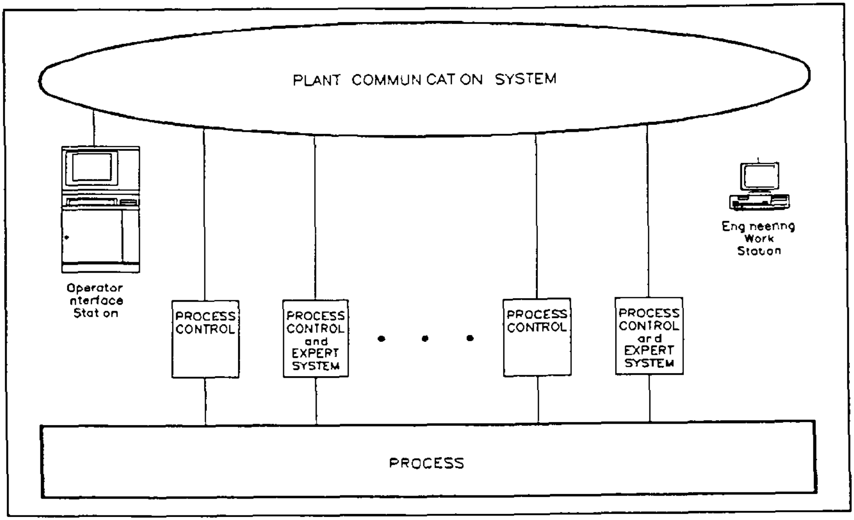

Figure 1 - Embedded Expert System Architecture

Figure 2   Typical EXPERT-90 Knowledge Base

(1) IF HEATER TEMP_RISE INADEQUATE AND MOISTURE_CONTROL_INADEQUATE THEN REQUEST LOAD LIMIT REDUCTION AND SEND MESSAGE_4 {reduce dryer load limit} END

(2) IF SPC_XBAR PATTERN ALARM_1A OR SPC XBAR PATTERN ALARM 4A OR SPC_XBAR_PATTERN ALARM_5A THEN MOISTURE_CONTROL_INADEQUATE END

(3) IF JUST NOW REQUEST_LOAD LIMIT REDUCTION THEN NEXT REQUEST COUNTER - REQUEST COUNTER + 1 AND OLD LOAD LIMIT   LOAD_LIMIT AND SET CHECK_LOAD LIMIT AND (AFTER 180 SECONDS (FOR 3 SECONDS END LOAD LIMIT_CHECK)) END

(4) IF JUST NOW END LOAD LIMIT_CHECK THEN CLEAR CHECK_LOAD_LIMIT END

(5) IF CHECK_LOAD_LIMIT AND (JUST NOW (OLD LOAD LIMIT   LOAD_LIMIT) > .02 * OLD LOAD LIMIT) THEN NEXT ACTION_COUNTER - ACTION_COUNTER + 1 AND CLEAR NEXT CHECK_LOAD_LIMIT END

(6) COMPLY_RATIO   (100 * ACTION COUNTER) / REQUEST COUNTER END

(7) GET LOAD_LIMIT BLOCK 500

(8) GET SPC_XBAR_PATTERN_ALARM_1A BLOCK 505

(9) GET SPC_XBAR_PATTERN_ALARM_4A BLOCK 510

(10) GET SPC_XBAR_PATTERN_ALARM 5A BLOCK 515

(11) GET HEATER_TEMP_RISE INADEQUATE BLOCK 520

(12) PUT COMPLY_RATIO BLOCK 800

(13) PUT SEND_MESSAGE 4 BLOCK 805

```
                              EXPERT 90                    file  newdryer
Token Menu
 CLEAR                                     SET
 (                                         NEXT
 <New Name>                                ACTION_COUNTER
 COMPLY RATIO                              END_LOAD LIMIT CHECK
 MOISTURE_CONTROL INADEQUATE               OLD LOAD LIMIT
 REQUEST COUNTER                           REQUEST LOAD_LIMIT REDUCTION
 SEND MESSAGE 4                            {<Comment>}
  Delete Statement                          Leave Statement Unchanged

Editing
 IF JUST NOW REQUEST_LOAD LIMIT REDUCTION THEN NEXT REQUEST COUNTER -
REQUEST COUNTER + 1 AND OLD_LOAD LIMIT = LOAD LIMIT AND SET CHECK LOAD _IMIT
 AND ( AFTER 180 SECONDS

Enter

      2 type                 5 left
```

Figure 3   Context Sensitive Menu for Creating New Rules

```
                              EXPERT 90                    file  newdryer
Token Menu
 GET                                       PUT
 NEXT                                      <New Name>
 IF                                        ACTION COUNTER
 COMPLY RATIO                              OLD LOAD_LIMIT
 REQUEST COUNTER                           {<Comment>}
  Delete Statement                          Leave Statement Unchanged

Editing
 GET SPC_XBAR PATTERN_ALARM_1A BLOCK 505

Enter

      2 type                        6 right          8 del R
```

Figure 4 - Ease of Obtaining Real Time Process Data

00 52 21 04/11/07

00 52 25 04/11/07

00 52 25 04 11 07

For additional copies contact

Bailey Controls Company

29801 Euclid Avenue Wickliffe Ohio 44092 • (216) 585 8500
Telex 980621 • Telefax (216) 585 8756 or (216) 943 4609

Litho in U.S.A. 1189