MICHIGAN STATE UNIVERSITY LIBRARIES
3 1293 00800 6433

MSU
LIBRARIES

RETURNING MATERIALS:
Place in book drop to remove this checkout from your record. FINES will be charged if book is returned after the date stamped below.

115

JUN 13 1994

'ALMOST' REAL-TIME DIAGNOSIS
AND CORRECTION OF MANUFACTURING SCRAP
USING AN EXPERT SYSTEM

By

David Raymond Chesney

A THESIS

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

MASTER OF SCIENCE

Department of Mechanical Engineering

1987

ABSTRACT

'ALMOST' REAL-TIME DIAGNOSIS
AND CORRECTION OF MANUFACTURING SCRAP
USING AN EXPERT SYSTEM

By

David Raymond Chesney

Findings are presented on an expert system that uses both operator and transducer inputs in 'almost' real-time to diagnose scrap type and recommend corrective action to reduce/eliminate further production of this scrap type. During the development of the expert system, equal consideration was given to system logic, implementation in a manufacturing environment, and knowledge acquisition. The system is applied to a specific manufacturing process; however, the ideas are applicable to a wide range of problems in a production environment.

This thesis is dedicated to my grandparents:

John, Sylvia, Alfred, and Ellen;

because they give history.

It is also dedicated to my nephews:

Kyle and Kent;

because they are the future.

ACKNOWLEDGEMENTS

The following people deserve kudos and salutations for their assistance to this thesis:

Mike Rieke, for his continuous interest and curiosity about the system. He easily found both strengths and weaknesses in the expert system. He encouraged me to develop, and therefore better understand, the expert system's strengths. And, he pointed out the weaknesses so that I might find a better way.

Thanks also to Erik Goodman and Carl Page for their guidance and assistance as committee members. Their insights and thoughts continually aided me in the development of this project.

There are three ingredients to an expert system of this type: control experts, domain experts, and me, the knowledge engineer. If any of the triad are missing than the system will not work.

Thanks go to Mark Hunt and John Raymond, control experts. Their constant effort and belief in the expert system made it possible. Also, the experts deserve thanks. Especially, John

Douro, Jimmy Compeau, and Mark Brodfuehrer. Their thorough technical and common sense understanding of the manufacturing process made the knowledge base in the expert system complete and valuable.

Dr.'s Abdul Esfahanian, and Mihran Tuceryan for their direct and indirect ideas which applied to the expert system.

Thanks to the many authors of the many papers I have read, for little clues which added up to the big picture.

My parents Dale and Bev, my brothers and sister-in-laws Dale, Nanette, Doug, and Julie for keeping the faith and keeping me smiling.

Jean, a caring friend. Writing and recieving letters from her provided a necessary link to complete this research. Her sense of humor, intelligence, and creativity kept me going through some particularly difficult times. And, for that I am forever in her debt.

My friends Dave L., Scot, Hushk, Fridge, Jorg, Yogi, Jeff, JO, Jane, Dave M., Ace, DPH, and others I have forgotten to mention for sharing in the struggle of higher education, and therefore, some of the accomplishment and frustration associated with the journey.

And, Penny Pullmain and Marci VanDerwill, for typing and editing the manuscript.

TABLE OF CONTENTS

LIST OF FIGURES

INTRODUCTION

Artificial Intelligence, in simple terms, is the application of computers that in some way simulates human thought or information processing. There are numerous approaches to AI. A philosopher or psychologist might ask "What is thought?", while an engineer might ask "How can some area of intelligence or knowledge be algorithmically simulated?"

Any reasonable solutions to the above questions could, at best, be general enough to apply to a wide range of similar applications, and would necessarily be refined to apply to a more specific domain.

This is a top-down approach from a general to a specific approach. The research for the expert system described in this thesis, however, is just the opposite - a bottom-up approach. The expert system will analyze and recommend corrective action for a specific manufacturing process, but the ideas have wide use in the more general AI arena.

Expert systems are an area of artificial intelligence in which the knowledge of an expert in a specific subject domain is recorded in computer software. The system is used to aid and educate non-experts in arriving at logical conclusions similar to those of an expert.

The specific expert system discussed in this thesis diagnoses and recommends corrective action to reduce and/or eliminate scrap production for the lost foam pattern molding process. The expert system bases its decisions on both quantitative and qualitative inputs and operates in 'almost' real-time. The long term intent is for the expert system to maintain on-line control to eliminate scrap before it is produced.

The lost foam process is a metal casting process in which expanded polystyrene foam patterns (positives) are inserted in a metal box containing dry unbonded sand which is subsequently compacted by vibration. The polystyrene is vaporized when molten metal is introduced. The internal gas pressure from the decomposing foam maintains the shape of the exterior of the pattern until metal solidification is complete; thus the casting is a duplicate of the polystyrene it displaces.

This expert system works on the pattern molding phase of the lost foam casting process. Pattern molding is the production of the polystyrene patterns which will be assembled and eventually vaporized. Pattern molding is a fifteen-step manufacturing process with a plethora of process variables. Briefly stated, pre-expanded polystyrene beads are blown into a preheated die cavity. Application of steam over time allows the beads to fuse together in the shape of the cavity. Next, the foam pattern is cooled by conduction through the water-cooled die. Finally, the pattern is removed from the die after it has cooled enough to stabilize dimensionally. The entire process typically requires approximately sixty seconds.

Pattern molding is an ideal application for an expert system since it requires considerable expertise critical to producing quality castings. The efficiency of the lost foam casting process will be greatly enhanced if pattern scrap can be diagnosed, corrected, and eventually preventd during the foam pattern molding operation.

The thesis is organized as follows:

Chapter 1 is a brief description of current artificial intelligence technology. Also included in this chapter is a summarized history of the most widely accepted area of artificial intelligence -- expert systems.

Chapter 2 is a discussion of the expert system architecture and logic. Also included in this chapter is a glossary of related terms and a discussion of input data types for the expert system.

Chapter 3 is a discussion of the expert system component implementation, installation, and debugging in the manufacturing environment.

Chapter 4 is a discussion of the manpower requirements for the development of the expert system. The triad of expertise and the steps involved in knowledge acquisition are discussed.

Chapter 5 is an evaluation of the expert system using simulated case studies. Four examples, each showing a different strength of the expert system, will demonstrate the utility of the system.

Through research and studies, Artificial Intelligence has presented itself as an expansive, nebulous discipline. Any research in AI furthers and better defines the science in general, as well as improving the quality and quantity of the specific application.

CHAPTER 1 - BACKGROUND OF ARTIFICIAL INTELLIGENCE

Before any discussion of the expert system developed in this thesis, it is prerequisite to understand some major concepts used in artificial intelligence. Also, as an aid in understanding, the history of early significant expert systems will be discussed. When discussing the history of artificial intelligence the names Ballard & Brown, Feigenbaum, Minsky, Newell, Schank, and Winston are often mentioned. Landmark projects, such as DENDRAL, MYCIN, and Waltz's constraint propagation have furthered the art of AI, as well as shown the potential for other areas of research. The expert system discussed in this thesis does not depend too strongly on any individual or theory, but rather fuses applicable ideas from many different authors and ideas.

In "Artificial Intelligence, Second Edition," Winston [1] divides artificial intelligence into the following areas of research: 1. representation schemes; 2. search strategies; 3. constraint propagation; 4. vision; 5. natural language understanding; 6. theorem proving ; and 7. expert systems. Representation, search, and constraint propagation are techniques or tools which are applied to the later mentioned areas. Vision, natural language, theorem proving, and expert systems, although sciences in their own right, use the "tools" for accurate and efficient system design. A simple analogy is a builder who needs a good hammer (representation scheme) to quickly and accurately construct a house (expert system). A good representation scheme is critical for knowledge representation in an expert system, but not vice-versa.

The Tools

Representation schemes are methods by which the knowledge is represented and/or stored. Winston [1] defines representation as "a set of syntactic and semantic conventions that make it possible to describe things." The knowledge is divided into two types: formal and common sense. Formal knowledge is written or recorded knowledge. Common sense knowledge is heuristic, or rule-of-thumb knowledge. The knowledge representation must have certain traits that allow the knowledge to be usable and accessible. Some of the traits are: the scheme must contain a complete set of knowledge in the subject domain; the full set of knowledge must be concisely stored; natural constraints of the system, or of science, must be exposed (example: a steam thermocouple reading below 212 deg F at atmospheric pressure would indicate no steam since the vaporization temperature of water is 212 deg F); explicit rules must be correctly prioritized; implied, rule-of-thumb, and heuristic rules must be made explicit; 'long-shot' possibilities must be deprioritized and yet still be included in the representation; the representation scheme must be transparent to the end user; and it must work well with computers. Some classical examples of representation schemes are semantic nets, frames (Minsky, Schank), and primitives.

Searches are methods of travel between a source and a goal. Searches are sequential in nature and can be classified as either uneducated, educated, or adversarial. Uneducated searches base the search upon an established search algorithm without consideration or knowledge of the domain. Uneducated searches are also described as blind searches. Some examples of uneducated search strategies are depth first, breadth first, and beam search. Educated searches are searches in which the next path followed is determined by some weighting factor based upon the subject domain. In other words, all paths are considered and the path with the greatest possibility of being correct is chosen as the next path to be explored. Examples of educated search strategies are A-star, dynamic search techniques, and branch and bound searches. The last type of search is adversarial. The objective of adversarial search is not necessarily to arrive at the goal quicker; rather, it is to beat an opponent. Some examples of adversarial searches are minimax and alpha-beta pruning.

Constraint propagation is the exploitation of limitations in a domain caused by the domain itself, or by nature. The best way to describe constraint propagation is with examples. An example of nature-based constraint propagation is the thermocouple mentioned above. An example of domain-dependent constraint propagation is Waltz's three-face vertex world. Winston [1] describes the significance of Waltz's idea using figure 1.1 as follows:

> The main problem is to determine which lines are boundary lines that separate objects. We find that boundary, convex, concave, shadow, and crack lines come together at junctions in only a few ways. Then we see that this restriction on junction combinations determines the proper physical interpretation for each line in a drawing. Once correct line interpretations are know, it is easy to use known boundary lines to divide the drawing into objects. Along the way, we will see that some impossible drawings can be detected because there is no way to interpret all the lines consistently.

In other words, we use what is known about possible interfaces between objects in the real world to come up with the correct interpretations of the two-dimensional image. In general, constraint propagation can greatly reduce the amount of search space by limiting the search paths.

The Applications

Vision, or image understanding, involves three areas: seeing, translating, and recognizing or interpreting. Seeing is a hardware problem and can involve cameras and stereoscopy. Next, the image has to be translated into some representation scheme that can be processed by the computer.

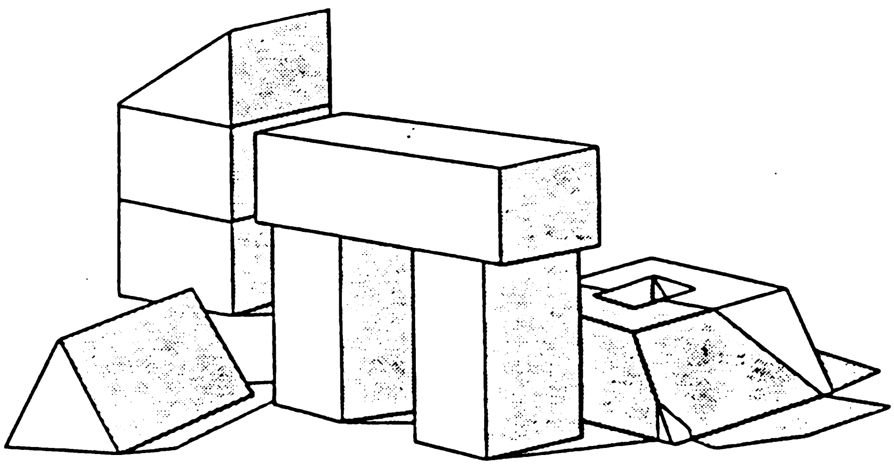


Figure 1.1: Waltz's Blocks

Recognition is the final and most complex step in the vision algorithm. Recognition involves determining what the object is once its shape has been translated.

Natural language understanding is the correct interpretation of the many ambiguities of the written or spoken word. Natural language understanding can be broken into three steps: first, parsing the the sentences into their syntactical breakings (the sentence trees we all did in grade school); second, furthering the understanding by thematic-role-frame or semantic understanding; and last, at the highest level, world model or relational understanding (example: if pi was used in a sentence, the world model would understand that pi equals 3.14).

Theorem proving is the exploitation of classical logic techniques (example: modus ponens) to arrive at a desired proof. In this area especially, good search techniques are necessary because of the potential for combinatorial explosion. A strength of theorem and classical logic techniques is that the theories are concise, have been developed through the ages, and are universally understood. A weakness of using this technique is attempting to 'fit' a problem that could more efficiently be solved using another method.

Expert systems are the most visible area of artificial intelligence. Briefly, expert systems involve programming the knowledge of an expert in a limited subject domain into the computer as an aid to nonusers and nonexperts. The rules are typically in if-then form (example: IF cloudy AND humidity is high AND barometric pressure is falling THEN there is a 85% chance of rain). There are several reasons for the success of expert systems. One reason is that an expert system can be highly profitable to the industry for which it was developed. Successful expert systems have been constructed to find mineral deposits (PROSPECTOR), as well as blood diseases (MYCIN). Another reason for the success of expert systems is that the knowledge of competent and sometimes costly experts can be recorded and saved on a medium available to the masses.

History of Expert Systems

The first expert systems constructed were similar in design, but very different in function. They typically had very limited domain expertise and were not time dependent, however, they did everything from determining the molecular structure of an unknown compound to diagnosing eye diseases. The following is a brief discussion of the history of some of the more significant expert systems compiled from Hayes-Roth, Waterman, and Lenat [3].

DENDRAL is regarded as the first working expert system. It was developed at Stanford by Buchanan, Mitchell, Feigenbaum, Lederberg, and Lindsay around 1964 to do mass spectographic analysis to infer plausible structures for unknown compounds.

Another expert system being developed by Slagle at MIT in 1961 was SAINT. SAINT eventually evolved into MACSYMA with the help of Martin and Fateman in 1971. Its function was to symbolically solve differential and integral calculus problems.

One of the next expert systems developed was MYCIN, which does diagnosis and consultation for infectious blood diseases. This was developed at Stanford in 1972 by Shortcliffe. A domain-independent version was developed in 1979 by vanMelle at Stanford. The domain-independent version, called EMYCIN, was significant because it was the first expert system shell.

Other significant expert systems are EXPERT, which evolved into CASNET for the diagnosis and treatment of glaucoma. It was developed circa 1970 by Weiss, Kulikowski, and Safir. And lastly mentioned will be CADUCEUS from Carnegie-Mellon University which was developed by Pople, Myers, and Miller around 1975. It contains approximately 100,000 associations between diseases and symptoms in internal medicine.

CHAPTER 2 - ARCHITECTURE AND LOGIC

In his paper "Sensor Fusion: The Application of Artificial Intelligence Technology to Process Control," Le Clair [4] defines sensor fusion as:

> the process of aggregating and understanding data from multiple sensors. Its significance and scope are best realized by considering the capability to be emulated -- human sense processing. Human understanding of the environment is accomplished by combining sights, sounds, touch, etc. Evaluation of these combined sense inputs produces a deeper and more reliable perception of the environment than does evaluation of any single sense or separate evaluation of each of them.

Le Clair developed the idea of using multiple outputs from a process to arrive at a correct conclusion. He equated human understanding of the environment using sight, sound, and touch with machine understanding of the environment using sensory inputs such as temperatures, pressures, and rates. By combination of the instrumented sensory inputs a machine can assess the environment much more accurately than with just one sensory type, say temperature. Therefore, the machine can more efficiently and accurately determine the appropriate conclusion or corrective action.

Le Clair's approach is an important step in the use of multiple sensory inputs. It must be noted, however, that the "sensor fusion" discussed in his paper arrives at conclusions based upon only measurable, or quantitative type, inputs. An example of this approach in a manufacturing environment is to instrument a production machine and to base control of the machine on transducer and thermocouple outputs.

While this form of "sensor fusion" might be sufficient for some applications, another valuable source of inputs is being neglected - namely, the machine operator. The machine operator can answer qualitative questions such as: "What does the product look like?"; and "What does the product feel like?" The operator can also answer questions about machine parameters that can't realistically be instrumented. An example is checking the integrity of tooling vents if there is a large number (say 300) of vents in the tooling. Obviously, it would not be realistic to instrument all of the vents.

By using both Le Clair's quantitative "sensor fusion" and the qualitative operator inputs, the most accurate assessment of the environment is possible. The use of all available inputs (quantitative and qualitative) will insure conclusions based upon "sensor - operator fusion."

The expert system discussed in this thesis relies heavily upon "sensor - operator fusion" (SOF). This chapter will build the SOF theory by first presenting a prerequisite glossary. Next, the quantitative and qualitative inputs will be further discussed. Third, the architecture, logic, and triggering for the scrap diagnosis portion of the expert system will be reviewed. And finally, the strategies for the corrective action portion of the expert system will be discussed.

Glossary

Rule-based expert systems are expert systems in which the rules are entered in an established rule format, such as IF-THEN rules. An example of a rule from a rule-based expert system is:

```
IF temperature is greater than 50 deg F
AND humidity is greater than 90%
AND barometric pressure is falling
THEN 75% chance of rain
```

Example-based expert systems are expert systems in which the rules are generated based upon real-world examples. The software in an example-based expert system shell automatically prioritizes and determines the IF-THEN rules. An example of an algorithm that determines rules based upon examples is the ID-3 algorithm which is discussed in Appendix A. Also, an excellent paper on rule generation from data is "Finding Rules In Data" by B. Thompson and W. Thompson [5].

Backward and forward chains: Chaining, in general, is connecting different knowledge bases together. It is the expert system equivalent to structured programming. In the global sense, forward chaining is working from the current state towards the goal state, and backward chaining is working backward from the goal state instead of forward from the initial state.

The global use of forward and backward chaining will be explained using Figure 2.1. In the diagram, the prefix "RF" means raw fact, "DF" means deduced fact, and "DR" means deduced recommendation. An expert system using forward chaining would ask the operator for RF1 and RF2 first. If both were true, then the expert system would deduce DF1. If either RF1 or RF2 were false, then the expert system would ask for the value of RF3. This process would continue until the system could reach either DR1 or DR2, or no further motion through the logic circuit was possible

Using backward chaining, the system would assume DR1 was true. The system would work backwards, seeing that DF1 is deduced from RF1 and RF2 being true. If either one was false, the system would move to DR2 , etc., assuming it true until it was proven true or false. This process would continue until a DR was found true, or until the search through all of the DR's was exhausted.

However, in this application, backward chains are knowledge base modules which are called to answer a specific question and return the answer to the calling knowledge base module. In contrast, forward chains will call another knowledge base when the expert system reaches a result, rather than

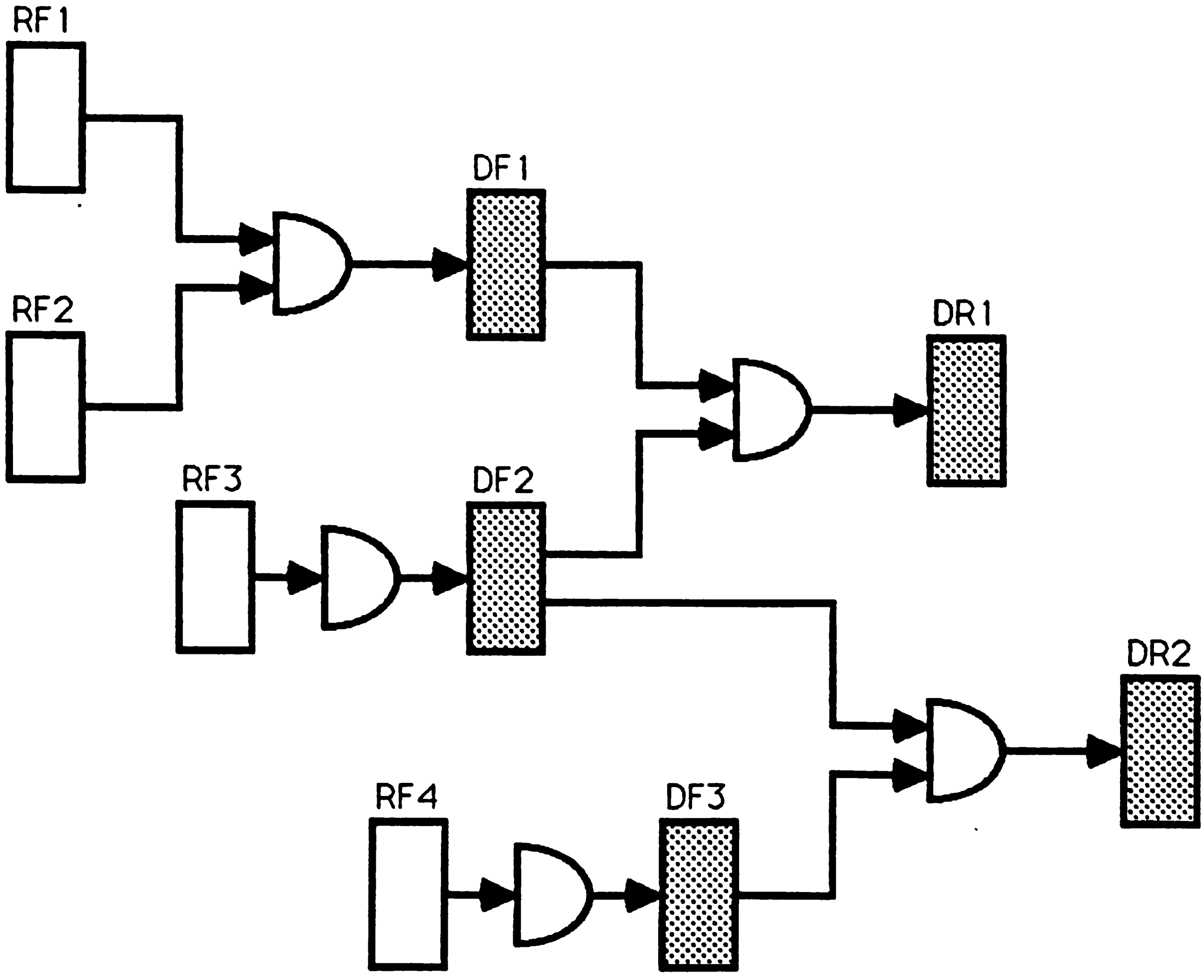


Figure 2.1: Example of Global Definition of
Forward and Backward Chains

reporting the answer to the operator. In graphic form, backward chains will be shown as parallel arrows both exiting and returning to the side of a knowledge base, meaning the objective is to go out, get an answer, and return. Forward chains will be shown as arrows coming out of the end of the knowledge base, meaning that control does not return to the calling knowledge base. See figure 2.2.

As an example suppose the objective of an expert system was to pick the correct wine for dinner. The specific information (result) desired is the vineyard (Bolla, Gallo), type (Rose, Reisling), and year of the wine. A backward chain, called WINECOLOR, could be called to determine the appropriate color (see figure 2.3), based upon entree and sauce. The control of the expert system is momentarily transferred to WINECOLOR before it returns to WINE. After the wine color is determined, more specific questions are asked to determine the specific vineyard (say Bolla) and type (say Rose). The only needed information is the year. The knowledge base WINE can forward chain into the knowledge base BOLLA-ROSE to determine the year. Note that the expert system does not return to WINE after the correct year for the wine is determined.

The expert system shell used for the development of this expert system was 1stClass by Programs in Motion. It is an example-based system shell that works very efficiently with forward and backward chaining.

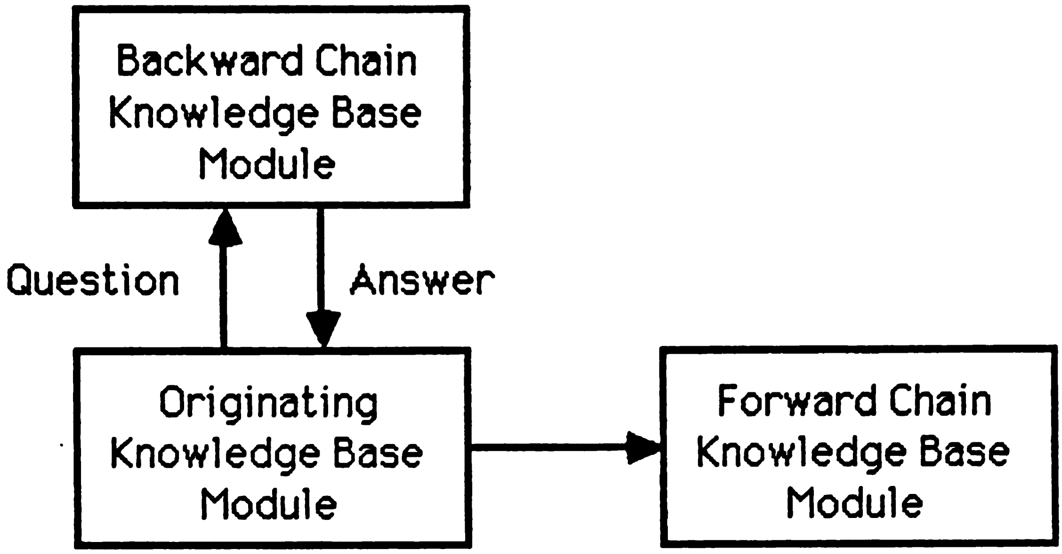


Figure 2.2: Forward and Backward Chains Using 1stClass

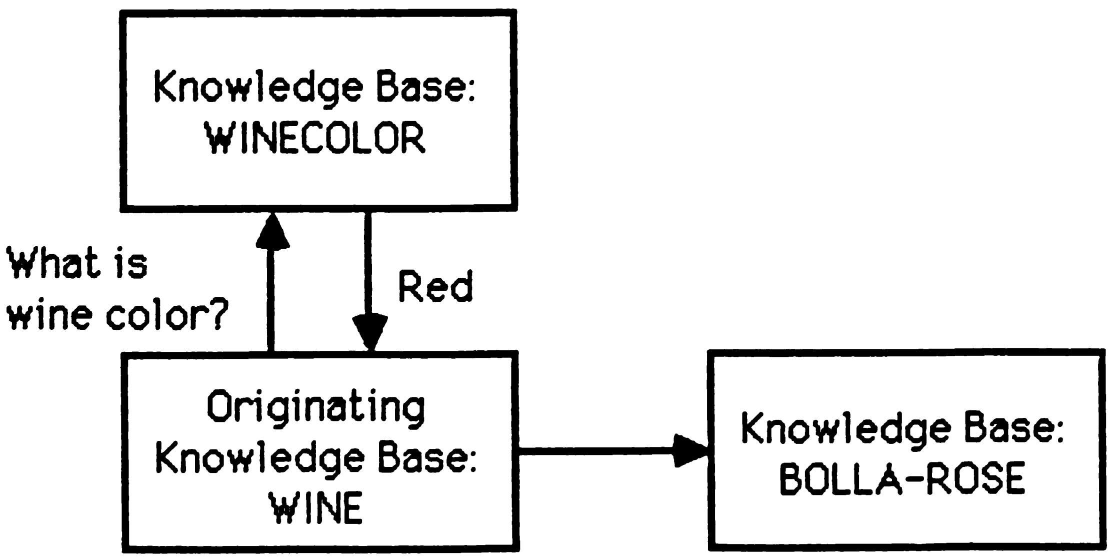


Figure 2.3: Example of Forward and Backward Chains Using 1stClass

Input Types

QUALITATIVE - Qualitative inputs are classified into two types: those that are truly qualitative in nature and those that are quantitative in nature but can't easily be instrumented. An example of the first type of qualitative input is 'What does the part look like or feel like?' An example of a qualitative input that can't easily be instrumented is the integrity of tooling vents if there are (say) greater than 300 vents in the tooling.

QUANTITATIVE - As mentioned earlier, quantitative inputs are operating parameters that are measured from the manufacturing process using transducers. The quantitative inputs for this application are time, temperature, and pressure. It must be noted that all of the transducer inputs can be measured at varying times in the machine cycle. Therefore, it is possible to obtain values for temperature or pressure at the beginning of the cycle step, or at the end. It is also possible to obtain local or global maximums and minimums. Since time can be measured, it is possible to record rates of changes of temperatures and of pressures. In other words, the expert system has an abundance of data on which to base its decisions.

Careful identification of both the critical qualitative and quantitative questions and parameters is necessary: under-identification will leave the expert system lacking enough information to make correct decisions; over-identification of the process may unnecessarily complicate the analysis. Also, another advantage of proper instrumentation of the manufacturing process is the possibility of yielding a process parameter on which machine control should be based.

INPUT DATA INTEGRITY or WHAT'S GRAY, HAS A TRUNK, AND LIVES IN A TREE? - The answer to the riddle used by Brachman [6] is "an elephant, I lied about the tree." The point of the riddle is that a final conclusion will only be as accurate as the facts upon which the conclusion was based. The expert system is basing decisions and branching according to the input data supplied. Suppose the question is asked "Is an object black or white?" and the reply is "black." The expert system depends on the fact that the object is actually black in future decisions. The expert system must be able to depend upon the integrity of the input data.

However, electrical and mechanical measuring systems (transducers) can, and do, fail in harsh environments such as automobile manufacturing. Therefore, some checks should be built into the expert system to insure the integrity of the transducers. The checks can be classified into two types: common sense and error code.

Common sense checks are limits that are based on scientific facts. Examples: A thermocouple reading steam temperature would not be expected to read less than 212 deg F (the boiling point of water at atmospheric pressure equal to 1 atm). If the thermocouple read less than 212 deg F then it could be assumed that the thermocouple is malfunctioning. Other examples are a pressure transducer reading a negative value, indicating the transducer is faulty, or a thermocouple reading negative indicating that the connection is backwards.

The other type of transducer "check" is an error code. Some transducers and thermocouples have intrinsic error codes. Example: A thermocouple that reads 772 when the thermocouple wire is broken. In many cases the error code limits are built into the transducer.

In this particular system, the transducer error code and common sense limits are checked at a higher level in the expert system logic than might seem logical. The reason is because of size limitations of the 1stclass software, rather than lack of sound system logic. The transducer data can be used for process analysis after they have been accepted as not exceeding or matching any common sense limits or error code values. The expert system proceeds from this point assuming that the transducers are giving the actual values.

LOW AND HIGH ORDER ANALYSIS OF QUANTITATIVE DATA - The quantitative data can be used to determine the "health" of the process after validation using error code and common sense limits. There are many different levels and types of analysis using the recorded data. If a machine is properly instrumented, it becomes difficult to sort all the data into a usable format. Rather than not having enough data on which to base decisions, the amount of data can be overwhelming and must be sorted and used appropriately.

The appropriate first level of process analysis for the expert system is using the recorded quantitative data to determine statistical operating parameters. This analysis is the simplest or lowest form and involves determining means and standard deviations of the process. Windows or limits are determined in which the process is said to be healthy. While the manufacturing process continues to operate within the specified windows the assumption is made that the process is OK. If any process limits are exceeded then further evaluation is completed.

However, after the data are available, there are many higher order types of analyses that could (should) be done. Some possible uses of the data are:

Control envelopes (volumes, spaces): As an example, when the steam temperature increases the process need not maintain the same amount of steam pressure. In other words, the values that could be exceeded are represented in a control volume, rather than by a simple control limit.

Seasonal variance: the analysis and data could account for and adjust to seasonal variances in the process. Example: the cycle time for the process might be longer in the summer than in the winter. A possible cause for the increase in cooling cycle time is a higher ambient temperature, and therefore higher cooling water temperature.

Process-related quality rating: Note that in the first-order analysis discussed above, only simple statistical limits were calculated. No comparisons were made against objective quality ratings. In other words, we recorded data and had only a binary value to compare it against (keep or scrap). To take advantage of the strengths of an example-based expert system shell, the data should be ecompared and recorded against objective quality ratings, such as a 1 - 5 scale on surface quality. An example is in figure 2.4. An example-based shell can sort the data and determine the cause of the differentiation in the surface quality based on the operating parameters used to create the specific part.

In-house quality control programs (SPC, Pre-control): In most manufacturing environments a well-established procedure for process control is already established. If a quality control program is established, the math on which it is based can be built into the expert system.

Cycle	Factor 1	Factor 2 •••	Factor M	Surface Quality Rating
1	100	200	0.5	1
2	125	250	0.5	1
3	90	90	1.0	4
4	97	185	0.95	3
•				
•				
•				
N	150	150	0.6	2

Figure 2.4: Example of Quality Rating Table

Architecture

This portion of Chapter 2 is a discussion of the architecture of the expert system. The rationale of why it was constructed will be included with the discussion of how it was constructed.

An attempt was made to simulate the human thought process when designing the overall architecture of the expert system. The overall objective is to decrease the production of scrap in a manufacturing environment. The human algorithm to solve the problem is first, to determine or diagnose the scrap type, and second, to determine the corrective action to reduce and/or eliminate the scrap. The human thought process might also involve diagnosing a process deviation before scrap is produced, but that is beyond the scope of this discussion.

The expert system architecture parallels the human thought algorithm. The overall expert system has two major subdivisions, expert system 1 (ES1) and expert system 2 (ES2). The objective of the first subdivision (ES1) is to interpret the appropriate quantitative and qualitative inputs (sensor - operator fusion) to determine the scrap type. The second subdivision (ES2) searches for the specific cause of the scrap and recommends the appropriate corrective action to the operator. ES1 and ES2 communicate to each other through a "blackboard." ES2 can't be triggered until ES1 writes the scrap type to the blackboard. In other words ES2 begins where ES1 ends. Figure 2.5 is a graphical description of ES1, ES2, and the blackboard.

The following discussions will further explain the logic of both ES1 and ES2.

ES1 Triggering - The expert system (ES1) can be triggered in one of three ways:

1. By a process parameter exceeding some predetermined limit.
2. By an operator who is inexperienced and/or doesn't know the scrap type.
3. By an operator who is experienced and knows the scrap type.

First Method (quantitative) - As earlier discussed, the process limits determined by either the expert system shell software, or higher-order analysis can be used as a quantitative trigger of the expert system. Through some hardware scheme (see Chapter 3) the current-cycle process parameters are recorded and compared against the predetermined limits. If any limits are exceeded then the expert system is 'awakened.' If no limits are exceeded then the process continues.

Second Method - (qualitative inexperienced) - The expert system triggers when the operator realizes a scrap problem

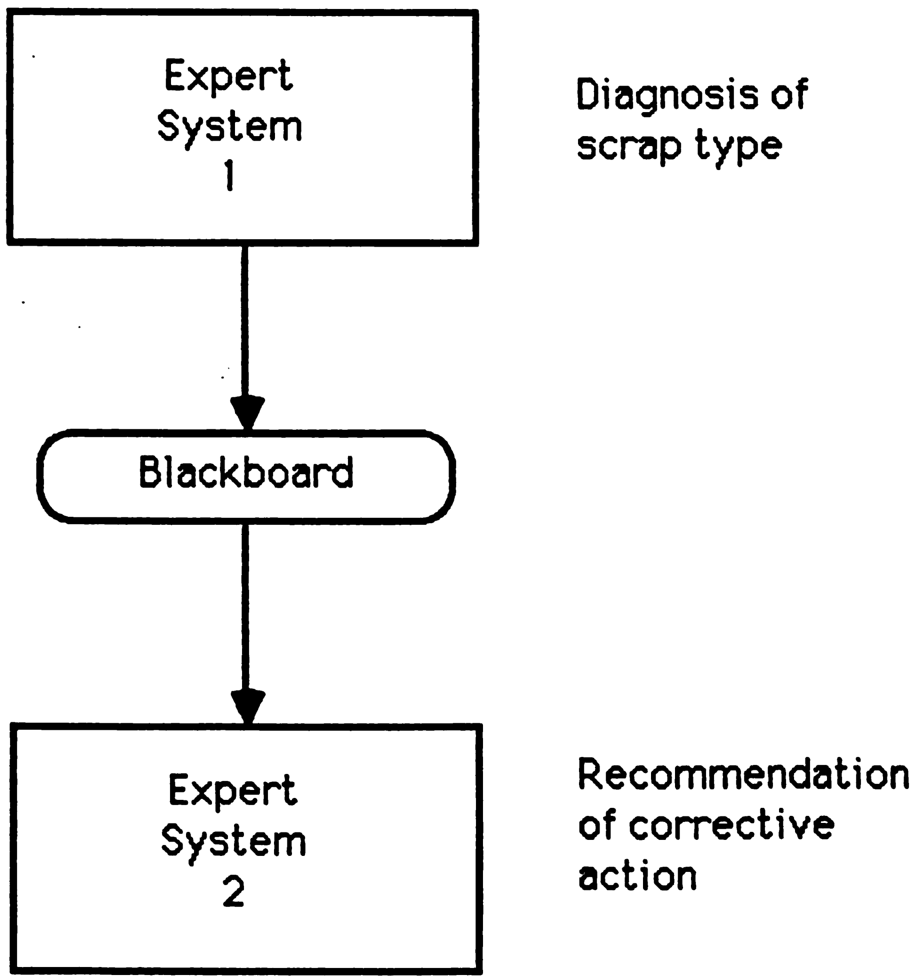


Figure 2.5: Expert System 1, Blackboard, and Expert System 2

exists, but cannot determine the specific type of scrap (i.e. "I know this part in my hand is scrap but I don't know why").

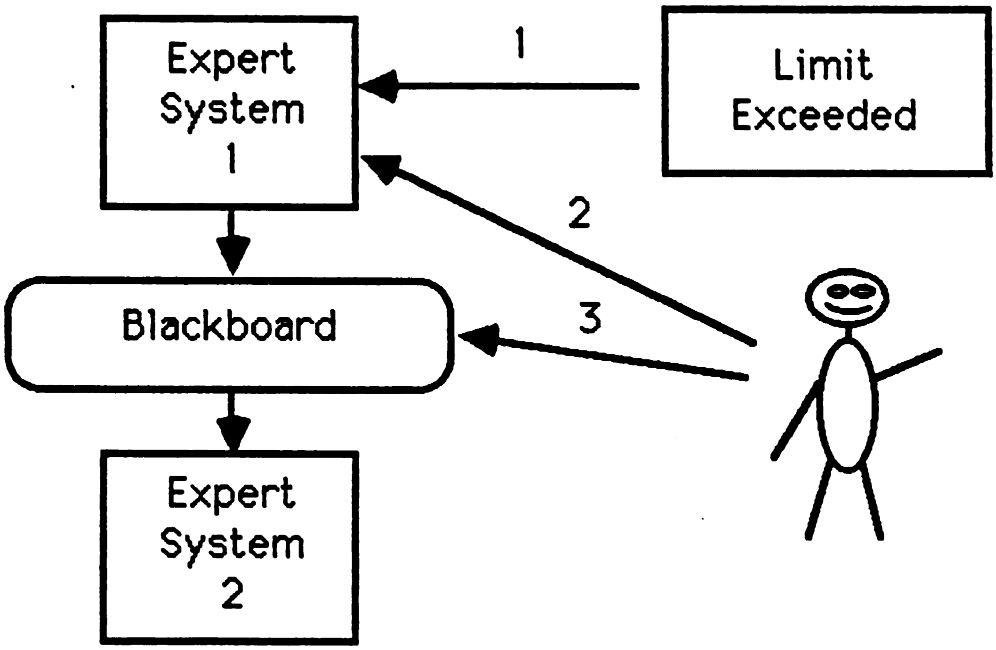
It is critical to ask the minimal number of questions in an expert system to arrive at the correct conclusion, in this case the correct scrap type. Again, too few questions can lead to an incorrect diagnosis of the scrap type. Too many questions can annoy the operator because of the perceived triviality of the answers. Originally, ES1 was developed to ask qualitative questions in the same order as the machine cycle. However, if the scrap type was caused by a later step in the machine cycle, then the operator had to answer unrelated questions in order to finally arrive at the correct scrap cause.

An example will more clearly demonstrate this point. Suppose the manufacturing process is the production of polystyrene (Styrofoam) coffee cups. The machine cycles through the steps: die close, bead inject, steam heat, die cool, die open, and part eject. Thus, if the operator has a scrap coffee cup caused by the part eject step, he/she would have to go through the die close, bead inject,...,die open steps before the actual scrap cause (part eject) would be determined. Although this method has its weaknesses, it is also inherently logical. Another method might be to prioritize the cycle steps in the order of greatest scrap production, rather than sequential. If the bead cool step produced the most scrap then questions to determine if the cup scrap was caused by bead cool would be asked first.

A third procedure, and the method used in this thesis, is to preprocess with large scope questions that can quickly narrow the scrap type choices. A number of questions are asked to get the operator in the correct area before going further to determine the exact scrap type. The sequential method discussed above is used if none of these questions effectively narrows the scrap type choices.

Third Method - (qualitative experienced) - The third triggering method is for the operator to write the scrap type directly to the blackboard. Typically, a highly experienced operator will write directly because he/she already knows the type of scrap. In this case, the operator will immediately be interested in the cause of the scrap and the corrective action. The third method is a bypass of ES1 (the scrap characterization subdivision of the expert system).

The triggering methods are best explained in figure 2.6. A simple example will best show the above triggering types. First, assume the manufacturing process is, again, styrofoam coffee cup production. The process step are die close, bead inject, steam heat, die cool, die open, and part eject. There are only two critical process parameters: die temperature and cooling water temperature.



Method 1: Predetermined Process Limits Exceeded

Method 2: Inexperienced Operator
- knows part is scrap, doesn't know type

Method 3: Experienced Operator
- knows scrap type, bypasses ES1

Figure 2.6: Triggering Methods

The predetermined range for the die temperature is 150 to 175 deg F and the predetermined range for the cooling water temperature is 55 to 85 deg F. The first method would trigger if, say, the measured process parameters for die temperature and cooling water temperature were 160 deg F and 100 deg F, respectively.

Now suppose the operating parameters were within specification but the cup was a visual scrap, and the operator didn't know what the scrap type was. The second triggering method would ask some critical preprocessing questions to attempt to narrow down the possible scrap type, such as "Is the coffee cup shrivelled?" or "Are there indentations in the coffee cup?" ES1 would ask sequential questions to determine the scrap type only if the preprocessing didn't yield any help. In other words, ES1 would ask die close questions, then bead inject questions, then bead heat questions,..., until the scrap type was determined.

An example of the third method is if the operator has a visual scrap and knows exactly what the scrap type is - say, part eject. The operator could then write the scrap type directly to the blackboard, bypassing ES1 completely.

ES1 Logic - The next area of discussion is the logic for ES1. The ES1 logic is the method by which the expert system uses the above-mentioned triggering.

Logic Table:

1. IF (quantitative triggers)
   AND
   (inexperienced qualitative verifies)
   THEN
   (write to blackboard)
   See figure 2.7.

2. IF (quantitative triggers)
   AND
   (inexperienced qualitative does not verify)
   THEN
   (give operator warning)
   See figure 2.7.

3. IF (inexperienced qualitative triggers)
   THEN
   (find specific qualitative type)
   AND
   (obtain process parameters)
   AND
   (write to blackboard)
   See figure 2.8.

4. IF (experienced qualitative triggers)
   THEN
   (obtain process parameters)
   AND
   (write to blackboard)
   See figure 2.9.

The next logical question is "Why the difference in the logic rules?" Rule 1 states that if the quantitative triggers ES1 and the relevant qualitative inputs from ES1 verify the scrap type, then scrap has definitely been produced and the expert system writes to the blackboard. An example is a thermocouple indicating cooling scrap type, and the operator visually verifying cooling scrap. However, if the cell controller triggers but the relevant qualitative inputs do not verify the scrap type (rule 2) then scrap has not actually been produced. An example of this is a thermocouple indicating a cooling scrap for the coffee cup, but the

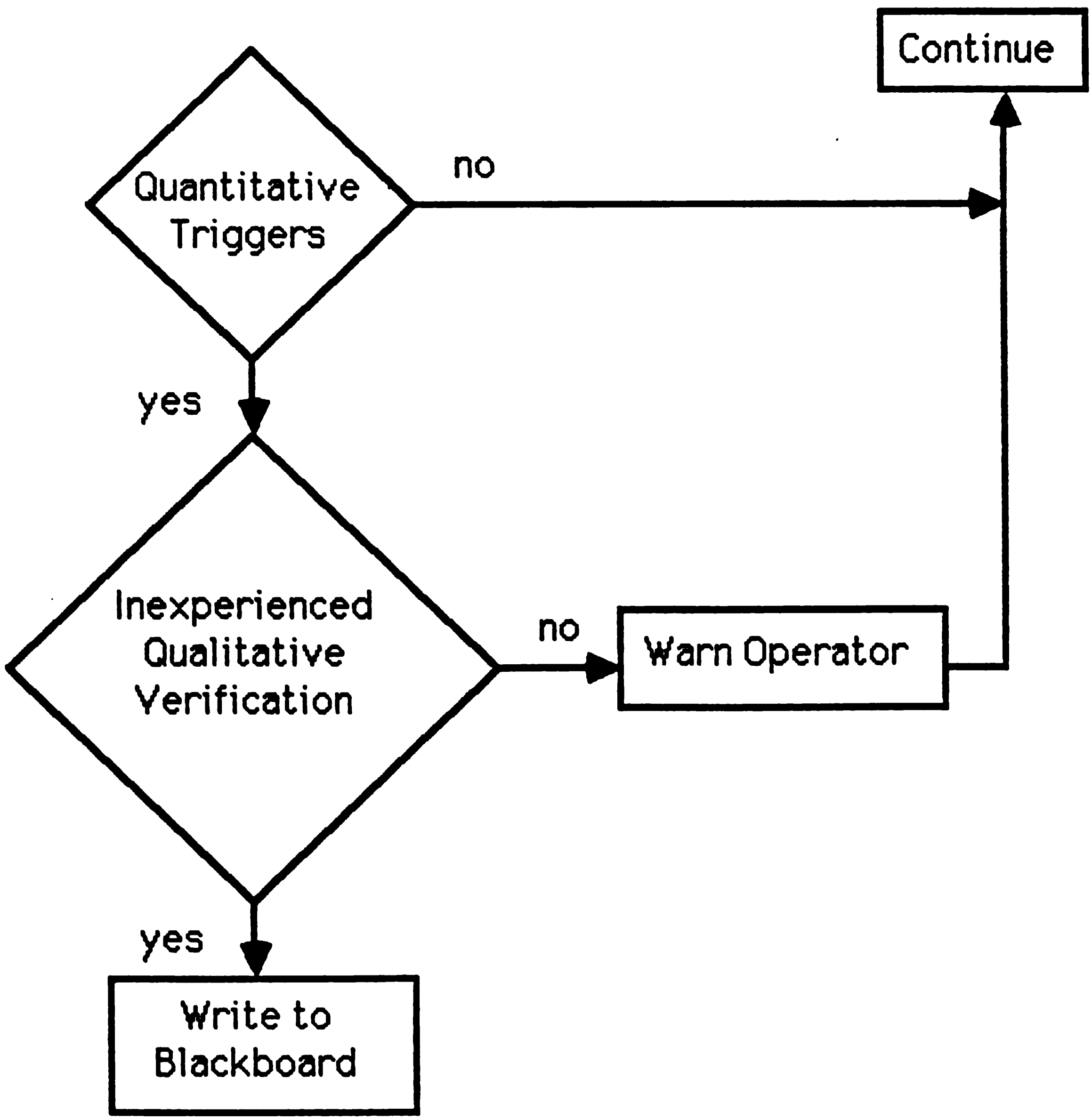


Figure 2.7: Quantitative Trigger

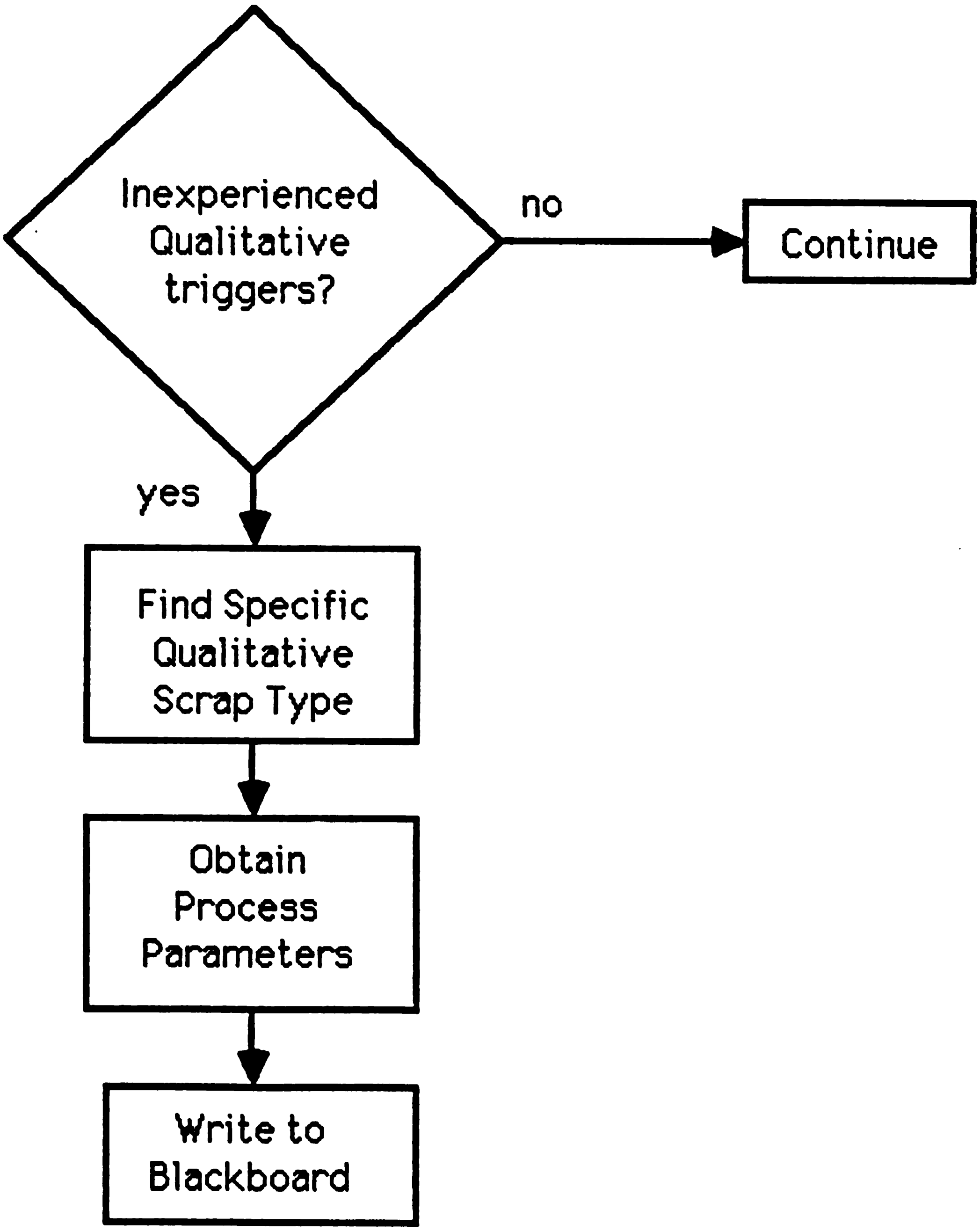


Figure 2.8: Inexperienced Qualitative Trigger

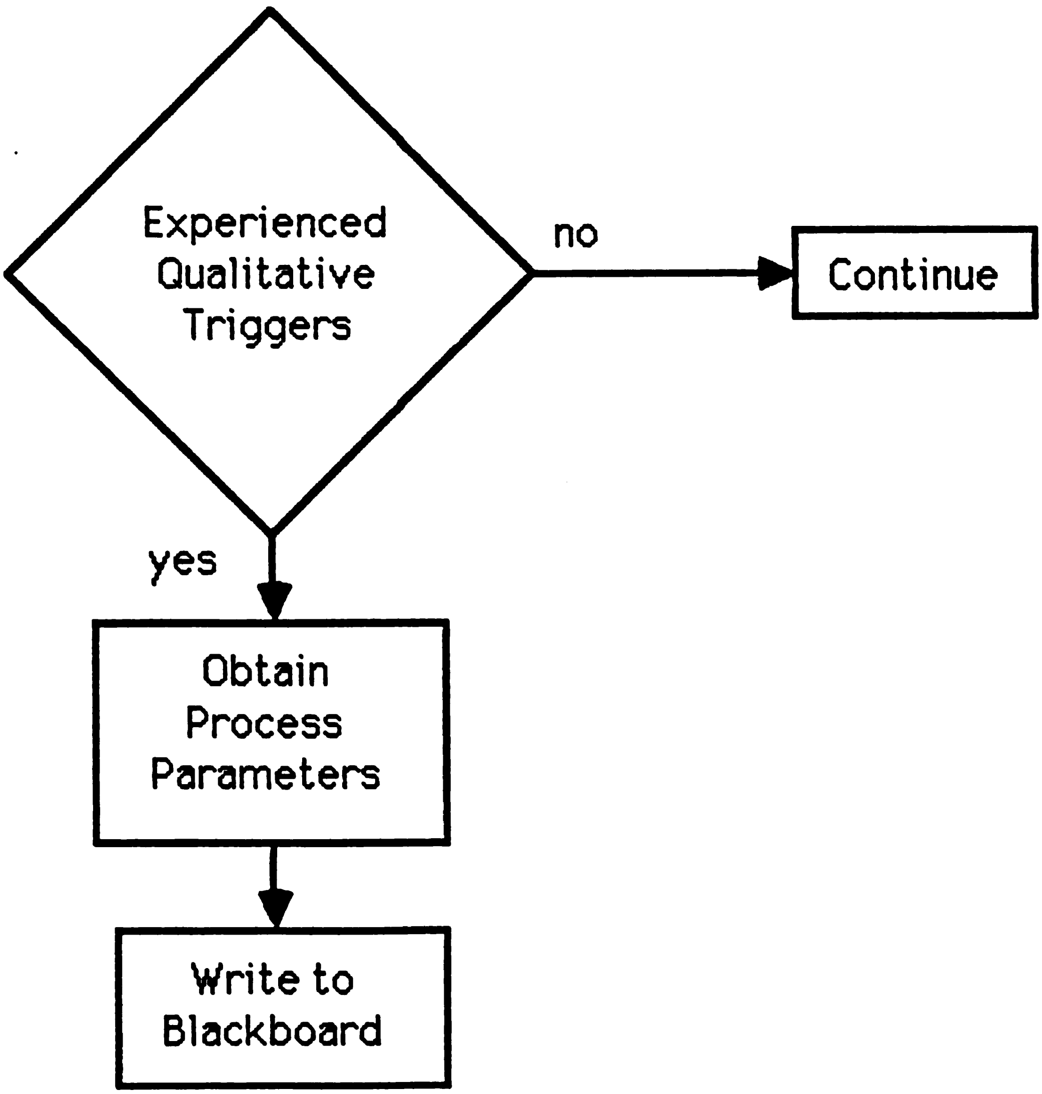


Figure 2.9: Experienced Qualitative Trigger

operator cannot verify the fact that is it is cooling scrap upon visual examination. Therefore, the operator is given a warning and the machine continues to cycle. Rule 3 states that if the ES is triggered by the inexperienced qualitative method then the ES obtains the process parameters from the cycle that produced the scrap coffee cup, and writes to the blackboard. An example of this is an operator seeing a visual defect in the coffee cup, but not knowing the cause of the defect. In other words, the scrap type is determined and written to the blackboard without regard to quantitative inputs. The logic for Rule 4 is obvious. The operator is writing directly to the blackboard and the process data is obtained to be used in ES2. Example: the operator knows that he/she has coffee cup damage caused by part eject and wants to know immediately how to correct the machine so this scrap type does not recur.

ES2 Logic - The logic in ES2 pales in comparison to the logic in ES1. Simply stated, ES2 is a shortest path search from the scrap type to the proper recommendation of corrective action. The search begins (root node) at the blackboard and ends with either a specific recommendation, or information telling the operator what the corrective action should not be.

ES2 relies heavily on forward chains to higher and higher (or narrower and narrower) degrees of expertise. Example: The blackboard result is cooling scrap type. ES2 would first determine if the cause was water harness malfunction or corrosion build up on the tool. If it was determined that the cause was the water harness, then the expert system might forward chain to questions specifically about the harness. Eventually, the expert system would narrow the corrective action to replacing one of the harness guns.

ES2 is easily modifiable because of the compartmentalization of the knowledge base. The trait of being easily modifiable is important for continued system maintenance.

CHAPTER 3 - COMPONENTRY AND INTERFACES

Implementation of a good functional hardware scheme for the expert system is paramount to the overall success of the system. The components must not only work, but also work well together. The description of the hardware used in this project is divided into three areas. First is a discussion of the methods used to gain necessary familiarity with the process, and second is a discussion of the actual implementation of the hardware in the manufacturing environment. Specifically, the manufacturing process, programmable control scheme, and expert system will be discussed. Finally, problems with the hardware used in the expert system will be discussed.

Process Familiarization

The first step in the construction of the expert system is to gain an operational understanding of the subject domain (manufacturing process). During this early stage, the knowledge engineer (KE) spends as much time as possible observing and participating in the manufacturing process. Thorough familiarization with the process is essential to understand the cause-effect relations and other intricacies of the manufacturing process. Familiarization with the process does not make the KE an expert, but it does make it easier for the KE to ask the appropriate questions of the expert. Observation of the machine cycle enables the KE to characterize the types of scrap that could possibly be produced. As described in chapter 2 of this thesis, characterization of the scrap is essential to arriving at the proper corrective action recommendations.

Next, the machine cycle is divided into logical segments. A cycle is the set of steps that the machine goes through to produce one finished product. Typically, a machine cycle is easily and logically divided into steps. For instance, in the coffee cup example used elsewhere in this thesis, the cycle steps were: die close, bead inject, steam heat, die cool, die open, and part eject.

Next, the critical process parameters are determined based upon the above segmentation. The process parameters are chosen using many different criteria. First, expert input is used to list the potential values for instrumentation. The "wish" list is reviewed with the experts until a reasonable list of parameters for measurement is chosen. Next, the list from the experts is reviewed to determine which parameters can be measured at reasonable cost and function. Another criteria for choosing parameters to be measured is the probability that the parameter might later be used as a process control value. Those parameters that could yield the greatest information for triggering to the next step have a higher priority for instrumentation than parameters that have little or no value for process control. The next step in familiarization with the process is having the manufacturing process instrumented. On the surface, instrumentation may seem like a trivial task. However, if the machine is used in production, its instrumentation requires some careful orchestration. The machine is instrumented with transducers and thermocouples that can withstand a harsh manufacturing environment. Also, instrumentation of the machine should take place during scheduled machine downtime, so as not to interfere with production schedules.

Hardware Implementation

In Mark Hunt's thesis [7], he describes:

> Diagnostics, as applied to programmable machine and process control, incorporate additional software to detect faults, hardware modifications to provide 'feedback' of key process variables, and a communications scheme to annunciate any resultant faults. Diagnostics are incorporated at a higher level of program hierarchy, but are integrated in the actual control of the machine or process. ...the diagnostic system will provide the following:
>
> 1. A Data Acquisition System (DAS), which will provide a database of process data for process diagnostics and the on-line Expert System.
>
> 2. Accurate monitoring of key process parameters in the (machines) will be possible. Included in this monitoring will be alarm capabilities when the machine runs outside of a process window.
>
> 3. The ... department will be provided with useful information about the productivity and utilization of the (machines).
>
> 4. Room for expansion, so that more (machines) can be added to the system in the future.

Implementation and debugging the hardware involves making several unlike pieces of equipment not only work together, but work quickly. Specifically, the manufacturing machine, programmable controller, cell controller, cell controller workstation (programming terminal), and the display and touchscreen, or keyboard, at the operator position will be discussed in this section.

Machine - The machine is obviously the manufacturing process which is being analyzed. A thorough understanding of the machine and manufacturing process is a prerequisite for giving the proper advice. In this thesis, the machine initiates its steps based upon instructions from a programmable controller.

Programmable Controller - The programmable controller is a machine control that steps through the machine cycle based upon either events, limits, or preset time. Examples of events are pressure maximums or minimums. Limits are values above, or below, which the control steps. An example of a limit is 100 degrees Fahrenheit. Preset time functions are steps that are triggered once a set time is met, say 3 seconds. The programmable controller (Hunt, 1987):

> has the ability not only to 'solve' logic in control applications but also has the enhanced capabilities of timing events, performing arithmetic computations, memory storage, and making decisions...

An additional function of the programmable controller is for data acquisition from the manufacturing process. The programmable controller will also be the medium through which the machine communicates with the cell controller workstation, and thus the expert system.

Cell Controller - The cell controller is the monitoring device of several programmable controllers. Main functions of the cell controller are monitoring, evaluation, communication, and coordination.

> (Hunt, 1987) The (cell controller) does emphasize these areas by providing the following capabilities:
>
> 1. Communications to other systems on the factory floor. These systems include both machine control devices and host computer systems.....
>
> 2. Machine control device monitoring and control within the cell controller's work 'cell.'
>
> 3. Database capability to store control parameters, status values, and computed values.
>
> 4. Computational and decision making abilities based on given or acquired data.
>
> 5. Direct operation of I/O.
>
> 6. User interfaces to configure the system and graphic output of system status and control variables.

It must be noted that the manufacturing plant was a beta site for the Gould FM1800 cell controller. There are hardware and software problems inherent to any beta site. Many of the bugs were fixed by Gould; however, development of the cell is ongoing.

There are three reasons for using the cell controller as a preprocessor for the expert system. First, the cell controller is designed for numerical evaluation. The expert system would be unnecessarily burdened if the analysis was done in the expert system rather than the cell controller. Second, the cell controller was used for analysis with future expansion to similar manufacturing processes in mind. If the analysis was done at a lower level, say programmable controller level, then the expert system would only work on that individual machine. Third, the numerical analysis needed for the expert system would greatly hinder the function of the programmable controller. The programmable controller might become so involved with numerical analysis that machine control would become secondary.

Workstation - The main function of the workstation is to program the cell controller. The workstation is an IBM AT with a UNIX operating system. The secondary function of the workstation is that it is the 'home' of the expert system. A partition of the workstation memory was formatted in DOS (the operating system for the expert system). Some reasons for locating the expert system in the workstation were: the workstation will only be used for programming the cell controller approximately 5% of the time after it is operational (Note: the workstation cannot be used to program the cell controller and run the expert system simultaneously); the workstation is physically near the machine and the cell controller on the plant floor; there was extra memory space available in the workstation; and it saved the cost and time of purchasing another computer.

Display, Keyboard, and Touchscreen - The display and the keyboard are used as an interface between the operator and the prototype expert system. The display is a standard IBM color display housed in a sheet metal box for protection. A touchscreen is mounted on the front of the color display as the eventual input mechanism. Some criteria for the input mechanism are that it be easy to use and durable enough to withstand the manufacturing environment.

The overall configuration of the hardware is shown in Figure 3.1.

Hardware Problems

The expert system is completely programmed to receive the data from the cell controller. Any bugs that might be in the expert system can only be found once the process parameters are successfully transferred from the cell controller. Problems discussed below are hardware and software bugs in areas other than the expert system.

> (Hunt, 1987) The (cell controller) has the potential to perform as advertised, but it has many software related shortcomings that have so far prevented this. Most notably, the communications, data handling capabilities, and some programming aspects were disappointing.

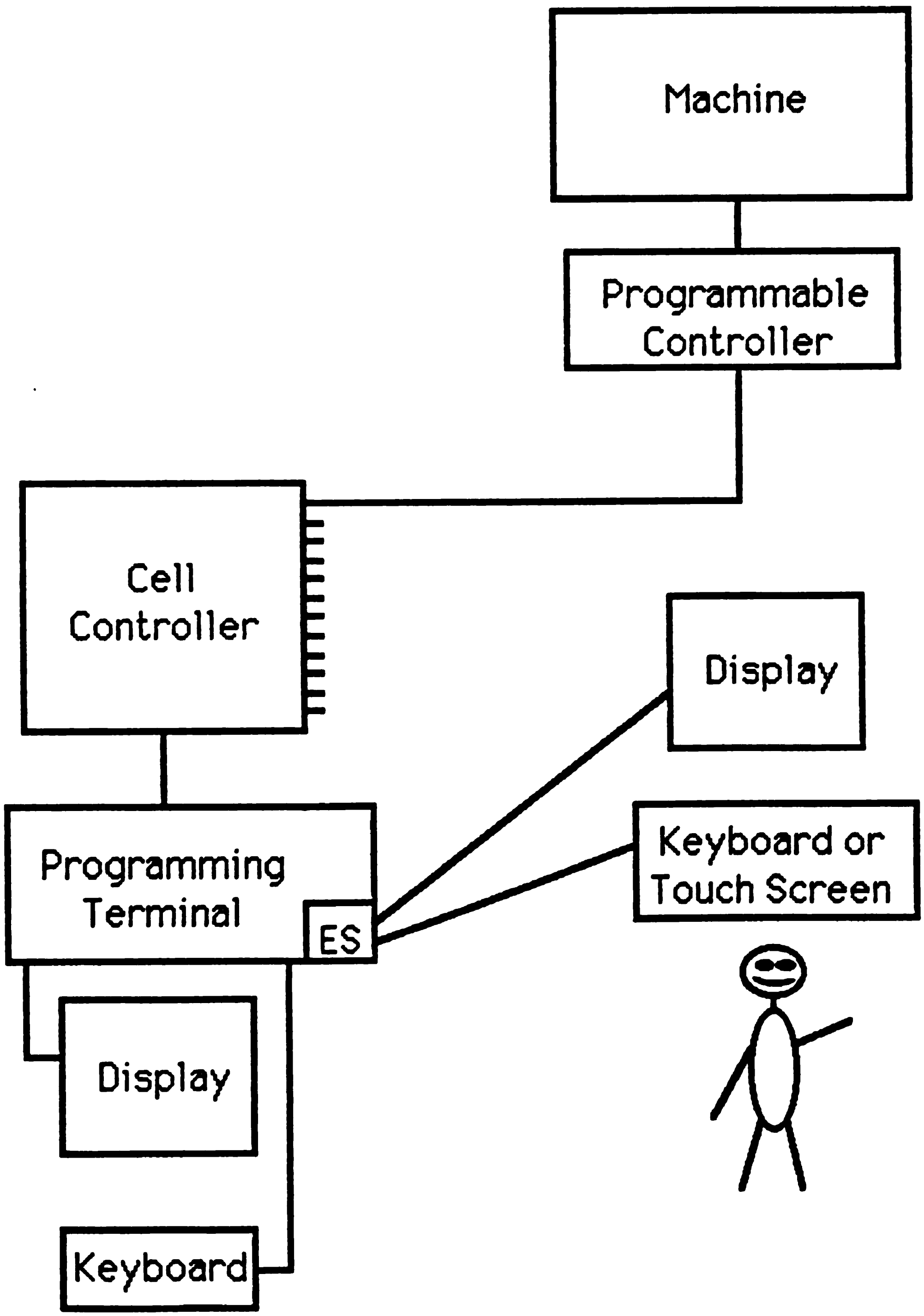


Figure 3.1: Hardware Configuration

As mentioned earlier, the hindrances are currently being worked on between the manufacturing plant and the cell controller manufacturer (Gould, Inc.). As it applies to the expert system, the cell controller and workstation are needed to read data from the programmable controller and bus the process parameters to the expert system if limits are exceeded. Continual problems have prevented the data transfer from occurring consistently.

Some of the problems were: the inability of the cell controller to send an array of the process parameters to the expert system (the cell controller could originally send process data one register at a time); the inability to return control to the cell controller after a consultation with the expert system was completed; and file addressing in the workstation.

CHAPTER 4 - KNOWLEDGE ACQUISITION PROCEDURE

This chapter is a discussion of the manpower requirements for development of the expert system. Although many facets of the manpower requirements will be discussed, the major requirement for a successful expert system can be summed up in one statement - insuring the time commitment of the experts. The importance of the experts to the development of an expert system might seem obvious, but it is possible to overlook prior to initiating a project of this nature.

This chapter will, first, give some necessary background information. Next, the expert system development triad will be discussed. Finally, the procedure for knowledge acquisition will be discussed.

Background

A distinction needs to be made between two very different types of knowledge: public and private. Public knowledge is knowledge that is recorded and accessible to a specific populace. It is typically published, written about, or recorded in some way. Some examples of public knowledge are training or trouble-shooting manuals. Private knowledge is knowledge that is not recorded but is known by the experts.

Private knowledge, which may often be rules of thumb, is knowledge that must be carefully extracted to make an expert system have real value.

Another distinction that will be made is the difference between reasoning strategies (logic representation) and knowledge. Reasoning strategies are methods by which the expert system arrives at a conclusion. It is the If-Then structure along with forward and backward chaining. The efficiency of the expert system is dependent upon the reasoning strategies. However, as Hayes-Roth, Waterman, and Lenat [3] suggest, the strength of an expert system is the knowledge contained in the system, not the methods by which the knowledge is represented.

> ...Elucidating and reproducing such knowledge is the central task in building expert systems.
>
> Researchers in this field suggest several reasons for their emphasis on knowledge itself rather than on formal reasoning methods. First, most of the difficult and interesting problems do not have tractable algorithmic solutions since many important tasks originate in complex social or physical contexts, which generally resist precise description and rigorous analysis...
>
> The second reason for emphasizing knowledge rather than formal reasoning methods is pragmatic: human experts achieve outstanding performance because they are knowledgeable. If computer programs embody and use this knowledge, then, they too [sic] should attain high levels of performance...
>
> The third reason for focusing on knowledge recognizes its intrinsic value. Knowledge is a scarce resource whose refinement and reproduction creates wealth. Traditionally, the transmission of knowledge from human expert to trainee has required education and internship years long.

In other words, without an inclusive knowledge base, the expert system would be highly efficient at doing nothing. An expert system with a vast knowledge base can always be made to work better, but, a highly efficient expert system with a limited knowledge base is of minimal value.

Development Triad

I believe that the best approach for development of this type of expert system is a triad of specialists. The specialists are needed for different quantities of time at different points in the project, but the expert system cannot be constructed and maintained without any of the three. The three specialists are: the domain expert; the control engineer; and the knowledge engineer.

As mentioned earlier in the chapter, the most critical step in the development of the expert system is gaining the time commitment of the experts. A possible reason for the inability of insuring the time commitment of the domain expert is that the domain (process) expert will probably be in great demand on the manufacturing floor. The expert's time might be so precious that it is needed to keep production operating, which would not allow time for expert system development.

In a project of this type it is also critical to gain the time commitment of a control expert. The control expert's responsibilities will be further elucidated later in the discussion. Briefly stated, in the development of this expert system, the control expert develops the communication network for the various pieces of hardware.

Domain Expert - The domain expert is a technical name given to the person who is knowledgeable about the subject matter of the expert system. There were three domain experts used to construct this particular expert system: the machine operator, the operator's supervisor, and a manufacturing process engineer. The machine operator had knowledge regarding keeping the machine operating and quick fixes. The supervisor had a higher order understanding of the process, including the theories behind the process and preventive maintenance. Very high level expertise in specific areas of machine operation could be obtained from the process engineer.

Besides being a source for the knowledge in the expert system, a second important function of the domain expert is iterative refinement of the knowledge base. In other words, the expert reviews the expert system with the knowledge engineer to note the system's strengths, weaknesses, accuracy, and completeness. Through the iterative procedure the expert system takes shape, and the questions and answers better simulate the thought procedure of the expert. It is helpful to use real examples of manufacturing scrap from the plant floor during the system refinement. Common and uncommon scrap was saved by the machine operator and examined using the expert system to find if the system arrived at the proper corrective action.

Another aid to constructing is to have the expert put the knowledge into flow chart form. A flow chart is easily translatable into a rule and effectively displays the algorithm that the expert goes through to solve the scrap type. The questions can be better prioritized, so as to be asked in the proper order. Greatest probability, least effort questions should be asked first. However, given the choice between the two, it is better to choose a least effort question first. Example: A machine operator would probably rather check all the low probability potential problems before checking a higher probability scrap cause that would involve dismantling the tooling.

Controls Engineer - The function of the controls engineer, as applied to this project, can best be summed up in one word - communications. The controls engineer is the vital link in setting up the communication scheme between the manufacturing process, controls of the process, and the expert system.

In this project, the controls expert worked with millwrights, electricians, and pipefitters to get the machine instrumented to measure the desired process parameters. As mentioned elsewhere, instrumentation had to be orchestrated with a production schedule on the machine. A scheme also had to be developed to record the process parameters after they were output from the transducers and thermocouples. Next, software had to be developed in the cell controller to compare the process parameters to the process envelope. Finally, software had to be installed to communicate the process parameters to the expert system if the process was operating outside of the envelope.

Knowledge Engineer - The knowledge engineer's main responsibility was the overall coordination of the expert system development. The knowledge engineer works with both the domain expert and controls expert to insure reasonable progress in the development of the expert system. Coordinating with the domain expert involves extracting domain knowledge, as well as iteratively refining the knowledge. Coordinating with the controls expert involves developing consistent and usable data structures for use throughout the expert system.

Another major responsibility of the knowledge engineer is knowledge acquisition from the experts. The long term usefulness of the expert system is directly dependent upon the amount of knowledge obtained from the experts.

Another responsibility of the knowledge engineer is the development of the knowledge and logic representation schemes. A thorough discussion of the architecture, logic, and triggering for the expert system is contained in Chapter Two of this thesis.

Knowledge Acquisition Procedure

Knowledge acquisition is the method used to gain the knowledge to construct the expert system. It involves, first, gaining familiarity with the process by spending time on the plant floor. The time is spent learning who the domain experts are, observing the process, and gaining a basic understanding of the cause-effect relation of scrap production.

Next, an operational understanding of the manufacturing process is gained through public knowledge. Appropriate process documentation is read to increase the understanding of the process. At this time, organization of the knowledge into acceptable architectures can be attempted. Architecture development is an iterative procedure and thus the first attempts might show the knowledge engineer what architectures won't work, rather than which will.

Then, the knowledge engineer clearly identifies the domain experts that will be used for knowledge acquisition. The primary goal of this step is to increase the knowledge base from public to private knowledge usable in the manufacturing environment. Secondary goals are to alleviate any expert's fear of the computer, and to gain the expert's confidence in the expert system as an aid.

The actual steps used to acquire private knowledge were:

1. Develop a rough architecture (shell).
2. Have the domain expert think logically about the problem-solving algorithm he/she goes through for reducing and/or eliminating a particular scrap type. The problem solving can be general (the ejection step) or specific (the water nozzles used in the cooling cycle).
3. Work the new knowledge into the expert system.
4. Debug and refine the just-coded knowledge with the expert.
5. If expert is satisfied with level of expertise then end, else
6. Go to 2.

These steps worked well for the development of this system because the domain expert could observe the growth of the system in the areas that had recently been discussed.

CHAPTER 5 - EVALUATION (OR, WILL IT WORK?)

This chapter contains four case studies using the expert system. Following the case studies is a discussion of the possible levels of expertise of the expert system. Finally, a brief summary of the status of the expert system is presented.

The examples are meant to be diverse enough to show the scope and capabilities of the expert system. Entry into the expert system using both the quantitative and qualitative entries will be explained. The cases explained in this chapter are simulations, based on potential scrap production situations. Actual use of the quantitative portion of the system is awaiting further development of the cell controller. Currently, the qualitative portion of the expert system is installed and being debugged in production.

The four cases will be presented and discussed as follows: first, the symptoms of the scrap type will be discussed. Next, the dialogue and display between the expert system and the operator will be shown. In the dialogue section, 'Q' means questions asked by the expert system, 'A' means answers given by the operator, and 'R' means the results given by the expert system. Next, the logic, in the form of the expert system shell '*.rpt' file, will be presented and discussed. Finally, the effectiveness of the expert system will be critiqued.

The experts who evaluated the expert system were Greg Sanders and Robert Masters from the General Motors Technical Center. Greg is a Project Engineer in the lost foam area of the Metal Casting group. Robert is the Chief Molding Technician in the same area. Both Greg and Robert were given the symptoms, as described in this chapter, and were allowed to ask questions regarding the machine status (i.e. "Are the vents clogged?").

There are questions asked in these case studies that are quantitative in nature (example: Case 3; Are the moving and stationary steps within specifications?). In most cases, the quantitative questions were built into the expert system as a temporary measure while the communications between the pattern molding machine and the expert system were debugged. The quantitative questions can easily be eliminated once the cell controller is operable. There are some cases, however, where quantitative questions are asked which were not instrumented on the pattern molding machine. These parameters were judged as not needing measurement at the time of machine instrumentation.

CASE 1 - Bad Transducer

Note: This example is an actual case history, but the quantitative portion of the expert system was not functional (due to lack of communication capability with the cell controller).

Symptoms

Grossly underfused beads. Disintegrating pattern as ejected. Pattern pieces and unfused beads floated in transfer water. Intermittent, 3-4 times per day with increasing frequency over a period of 3-4 days. When scrap type occurred, manufacturing process had to be stopped and transfer water cleaned.

Dialogue

Note: There is no dialogue between the expert system and the operator in this case. The transducer analysis is transparent to the machine operator.

R: Stationary side insert thermocouple malfunction.

Logic

A brief explanation of the *.RPT file format will be given here and is typical for all *.RPT files in this chapter. The first column is the name of the knowledge base currently being used. Column two contains one of two types of names:

if there is no "#" prefix (example: 2beintmpmv) then the name is a variable name; if there is a "#" prefix (example: #npreheat) then the name is a backward chain. The final column is the value assigned to either the variable name or the backward chain. As an example: the first row of the *.RPT file below would indicate that the variable "2beintmpmv" was assigned the value of 130.0 in the knowledge base "npreheat." As further explanation, the last row is always the results. The last row, first column is the calling knowledge base, or where the expert system consultation ended. The last row, second column is the title of the result column. The last row, last column is the actual corrective action.

npreheat	2beintmpmv	130.0
npreheat	2beintmpst	411.0
quanchk	#npreheat	high
NES1	#quanchk	nh.prht
ntprht	2beintmpmv	130.0
ntprht	2beintmpst	411.0
NES1	#ntprht	intcmv
NES1	BLACKBOARD	transducer

Although the above discourse between the cell controller and the expert system seems simple, the mechanics of what actually went on are complex. The cell controller was doing compares using process limits determined from production operation. The cell found that a value was out of specifications and wrote all the process parameters to *.ANS files. *.ANS files are answer files which are read directly into the expert system without any operator input. After the files were written, the cell used a BASIC 'SHELL' command to awaken the expert system, specifically the NES1 (quantitative expert system 1) knowledge base.

The first knowledge base in NES1 is npreheat. 'npreheat' looked for an answer file called 'npreheat.ans.' The values in npreheat.ans were read into the expert system and compared with numeric rules. The first value (2beintmpmv) was ok, however, the second value (2beintmpst) was high. It must be noted that npreheat is a backward chain from quanchk, and quanchk is a backward chain from NES1. 'npreheat' sent the value 'high' back to 'quanchk', which in turn sent the answer 'nh.prht' (quantitative high preheat) to NES1. NES1 has logic built in to check the out of spec parameter against error code and common sense limits for the transducer or thermocouple in question. The global value for 2beintmpmv was 411.0, which was higher than the error code limit for that particular thermocouple. The result, which was written directly to the blackboard, was that the thermocouple had malfunctioned. In this case, all of the transactions between the expert system and the machine were transparent to the user.

Critique

There is virtually nothing to critique in this case. A simulation of the transducer outputs, in the form of *.ANS files showed that the logic worked. All experts agreed that checking the integrity of the transducer outputs was a worthwhile idea. The question was asked by one expert, Will the transducer checks work during the same cycle as the scrap was produced?" The answer is no. The expert system cannot currently be used within a cycle as an automated feedback control mechanism, but rather in "almost" real-time as a diagnosis tool.

CASE 2 - Quantitative Trigger, Qualitative Does Not Verify

Symptoms

Three of the four pattern molding machines reach the cooling step at approximately the same time. Therefore, all four machines experience an excessively low cooling water pressure. Transparent to the user, a transducer reading of cooling water pressure indicates an out-of-spec condition. The expert system asks the user if undercool scrap is being produced. In the case of cooling scrap, only one question needs to be asked (typically, five to fifteen questions would need to be asked to eliminate a particular scrap type).

Dialogue

Q: Was the pattern dimensionally stable at the end of the cooling step (no bulges in thick sections)?

yes
no

A: yes

R: BLACKBOARD

WARNING: Transducer inputs indicate undercool 1 scrap,
Operator inputs do not verify,
Therefore continue process

Logic

npreheat	2beintmpmv	133.00
npreheat	2beintmpst	133.00
npreheat	2tim	52.00
npreheat	2rtintmpst	120.00
npreheat	2rtintmpmv	141.00
quanchk	#npreheat	ok
nfill	4beairprs	82.00
nfill	4minairprs	81.00
quanchk	#nfill	ok
nfus1	6beintmpmv	186.00
nfus1	6tim	34.00
nfus1	6rtintmpmv	89.00
nfus1	6endchtmpmv	207.80
nfus1	6endchprsmv	12.00
nfusion	#nfus1	ok
nfus2	7beintmpst	202.00
nfus2	7tim	20.00
nfus2	7rtintmpst	90.0
nfus2	7endchtmpst	201.00
nfus2	7endchprsst	10.00
nfusion	#nfus2	ok
nfusdwl	8beintmpmv	213.00
nfusdwl	8beintmpst	217.00
nfusdwl	8befomprs	104.00
nfusdwl	8tim	76.00
nfusdwl	8maxfomprs	225.00
nfusdwl	8rtfomprs	160.00
nfusdwl	8endchtmpmv	231.50
nfusdwl	8endchprsmv	21.00
nfusdwl	8endchtmpst	225.00
nfusdwl	8endchprsst	23.00
nfusion	#nfusdwl	ok
quanchk	#nfusion	ok
ncool1	10befomprs	112.00
ncool1	10beintmpmv	224.00
ncool1	10beintmpst	210.00
ncool1	10bewtprs	30.00
ncool	#ncool1	under
quanchk	#ncool	udrcool1
NES1	#quanchk	nu.cl1
ntcool	10bewtprs	30.00
ntcool	10bewttmp	85.00
NES1	#ntcool	ok
lcool	no.post	yes
NES1	#lcool	ok
NES1	BLACKBOARD	warning

The preliminary parts of this logic discussion are similar to case 1. The cell controller was doing compares and found that a value was out-of-spec. All the process parameters were written to *.ANS files, and the expert system was awakened.

The expert system checks the parameters in the same order as the machine cycle sequence. The preheat step was checked first and all values were found to be within specifications, therefore, the result of 'ok' was returned to the backward chain 'quanchk.' Next, the expert system checked the fill step and returned the reply 'ok' to the backward chain 'nfill.' The same 'ok' response was returned for the fusion 1, fusion 2, and fusion dwell steps.

The expert system discovers that '10bewtprs' (step 10 beginning water pressure) is out-of-spec during the cooling 1 step. The undercool state was passed to the highest level of the quantitative analysis (NES1). 'NES1' first checks the transducer output against error code and common sense limits. The values are found to be within reason, and therefore, the transducer is assumed to be operating properly.

Next, the expert system asks the appropriate qualitative question(s) to determine if the pattern should actually be scrapped. The operator determines that the pattern is not scrap, and thus, a warning is given to the operator, rather than shutting down the process.

Critique

As with case 1, it is difficult to critique the quantitative portion of the expert system during this case. The quantitative expert system is a matter of pass-fail rather than a subjective evaluation. After some minor debugging of the architecture, the quantitative portion of the expert system did diagnose that the cooling water had low pressure.

The experts agreed that the best way to diagnose an undercool scrap was to observe the pattern for localized swelling (post-expansion). While running the expert system through different, related scenarios, the experts had several suggestions for improvement. They are:

1. Compare fusion moving and fusion stationary times to each other, rather than just the predetermined limits. This would give an indication of a fusion problem that is peculiar to one side of the tooling.

2. Record and compare the last five cycles of data. A trend analysis could show any local deviations from the process window.

CASE 3 - Inexperienced Operator

Symptoms

Inexperienced operator notices general beady appearance of patterns. Knows that pattern is scrap; however, is unsure of scrap type, cause, and corrective action.

Dialogue

Q: Do you know what the scrap type is:

yes
no

A: no

Q: Are there any pieces of pattern stuck on the tooling?

no
yes

A: no

Q: Is the beadsize consistent in the area of the defect (no localized oversize beads)?

yes
no

A: yes

Q: What does the overall surface of the pattern look like?

smooth and shiny
beady
shrivelled

A: beady

Q: Is the defect occurring on:

the moving side
the stationary side
both sides

A: the moving side

Q: Are the moving and stationary steps within specifications:

	Fusion 1 (Moving) Temperature deg F	Fusion 2 (Stationary) Temperature deg F	Fusion 1 (Moving) Steam psi	Fusion 2 (Stationary) Steam psi
A pattern:	195 to 215	195 to 215	10 to 12	11 to 13
B pattern:	210 to 230	210 to 230	10 to 12	10 to 12
C pattern:	205 to 225	205 to 225	11 to 13	11 to 13
D pattern:	205 to 225	205 to 225	11 to 13	11 to 13

yes
no

A: yes

Q: Check the vents in the area of the defect. Are all of the vents:
1. in place
2. open
3. undamaged
4. have the proper clearance between vent slots?

yes
no

A: yes

Q: Call up on the keypad step 34 (if moving side) or step 35 (if stationary side).
Run 2 or 3 machine cycles.
Is the thermocouple responding properly?
i.e. no values above 400 deg F.
no elongated step times.

yes
no

A: yes

Q: Cycle the machine.
Is the moving side drain valve light on (valve closed) during the step in question?

yes
no

A: yes

Q: Stop the machine cycle.
Manually energize the drain valve (to close) on the moving side.
Manually energize the water Magnatrol valve (to open) on the moving side.
Is water coming through the vents on the insert?

yes
no

A: yes

Q: Manually energize the water Magnatrol valve (to open) on the moving side without manually energizing the drain valve.
Is water coming through the vents on the insert?

no
yes

A: no

Q: Cycle the machine.
Is the stationary side drain valve light on (valve closed) during the step in question?

yes
no

A: yes

Q: Stop the machine cycle.
Manually energize the drain valve (to close) on the stationary side.
Manually energize the water Magnatrol valve (to open) on the stationary side.
Is water coming through the vents on the insert?

yes
no

A: yes

Q: Manually energize the water Magnatrol valve (to open) on the stationary side without manually energizing the drain valve.
Is water coming through the vents on the insert?

no
yes

A: no

Q: Cycle the machine.
Is the moving side exhaust valve light off (valve closed) during the step in question?

yes
no

A: yes

Q: Does the moving side exhaust valve light on the control panel go on (valve open) at the same time as the moving side exhaust valve is heard opening during step 6 (fusion moving)?

yes
no

A: no

Q: The moving side exhaust valve seems to be defective. Contact a pipefitter for further analysis.

Is the moving side exhaust valve getting pilot air to open?

yes
no

A: yes

Q: Is the diaphram on the moving side exhaust valve intact and untorn?

yes
no

A: yes

Q: Is the moving side exhaust valve plunger intact and functioning properly (not mechanically sticking or broken)?

yes
no

A: no

R: The recommended corrective action is:

Repair or replace the moving side exhaust valve plunger.

Logic

ENTRY1	entry.type	oper.uned
ENTRY1	stuck	no
ENTRY1	beadsize	yes
ENTRY1	surface	beady
ENTRY1	mov.stat	moving
DFS12	process	yes
DFS12	vents	yes
DFS12	t/c.ins	yes
cdvlvm	drn.vlv.me	yes
cdvlvm	drn.vlv.mm1	yes
cdvlvm	drn.vlv.mm2	no
DFS12	#cdvlvm	ok
cdvlvs	drn.vlv.se	yes
cdvlvs	drn.vlv.sm1	yes
cdvlvs	drn.vlv.sm2	no
DFS12	#cdvlvs	ok
cevlvm	exh.vlv.me	yes
cevlvm	exh.vlv.mm	no
DFS12	#cevlvm	not.ok
DEXHVLVM	pilot.air	yes
DEXHVLVM	diaphram	yes
DEXHVLVM	plunger	no
DEXHVLVM	RESULT	fix.plgr

The operator knew he/she had scrap, but didn't know the type. Therefore, he/she entered the expert system as an inexperienced operator. The expert system asked some general questions to determine the scrap type quickly, rather than going through the machine cycle sequence. Thus, the questions about stuck, beadsize, and surface. Once the system determined that the surface was beady, and therefore underfused, it queried the operator to determine whether just the moving, just the stationary, or both sides were involved.

After it was determined that only the moving side was involved, the expert system forward chained to the diagnostic analysis portion of fusion 1 and 2 (DFS12). DFS12 first asked question about the process, vents, and thermocouples. Since all of those were ok, DFS12 backward chained to check the drain valves on both the moving and stationary side. Both drain valves checked out ok, and returned an ok status to DFS12. Next, DFS12 checked the exhaust valve on the moving side and found that there was a mechanical problem. The knowledge base that checked the exhaust valve on the moving side (cevlvm) returned a not.ok status to DFS12, which forward chained to a more specific diagnostic knowledge base for the moving side exhaust valve. Three questions were asked to arrive at the conclusion that the plunger in the exhaust valve on the moving side needed to be repaired or replaced.

Critique

Both experts doing the evaluation agreed that the recommended corrective action of the expert system was a possible correction to this scrap type. In using the expert system the evaluators tried to play the role of an inexperienced operator. Obviously, simulating the role of an inexperienced operator was difficult for someone with a high degree of expertise. The evaluators also agreed with the general order in which the questions were asked, and felt that the questions were pertinent.

Both experts followed different paths through the expert system prior to arriving at the desired corrective action. This simulated real-life, because no attempt was made to make the simulations be the highest probability scrap cause. In other words, an inexperienced operator would probably reach some other corrective action first, try it, and then iterate through the system until the appropriate corrective action was recommended.

Also, during the evaluation, some questions were asked that could be confusing to an inexperienced operator. Wording corrections will be built into an updated version of the expert system. An interesting note is that one expert system path led to a boiler check when a steam pressure problem was recurring. The expert experienced that exact scrap type while pattern molding, and commended the expert system for its inclusion.

CASE 4 - Experienced Operator

Symptoms

An experienced operator enters the expert system knowing he/she has an underfill scrap type, but unsure of the exact cause. The scrap is consistent and localized in a fill gun area.

Dialogue

Q: Do you know the scrap type:

yes
no

A: yes

Q: What is the type of scrap:

preheat
underfill
underfusion moving
overfusion moving
underfusion stationary
overfusion stationary
underfusion dwell
overfusion dwell
post-expansion (undercool)
low layover (pattern didn't layover onto the moving side)
high layover (pattern was collapsed onto the moving side)
low ejection (pattern did not eject from the tooling)
high ejection (pattern ejected too harshly from the tooling)
pattern damage
pattern contamination

A: underfill

Q: Check the beads in the hopper. Are there sufficient beads and are the beads free from flakes?

yes
no

A: yes

Q: Manually blow the beads back into the hopper. Manually open the fill gun tips and blow air through the fill guns. Recycle the machine.

Has the scrap problem been eliminated?

yes
no

A: no

Q: Is the fill time sufficient?

A pattern: 4.0 s
B pattern: 3.5 s
C pattern: 3.5 s
D pattern: 3.0 s

yes
no

A: yes

Q: Is any portion of the pattern stuck on the tooling?

no
yes

A: no

Q: Is the defect occurring consistently (at least once every 10th cycle) or intermittently?

consistently
intermittently

A: consistently

Q: Is the defect local (less than 20% of the pattern) or global (30 to 100% of the pattern)?

local
global

A: local

Q: Since the defect is localized, is it directly on top of:

vents
ejector pin bushings
fill gun
none of the above

A: fill gun

Q: Is the fill gun in the area of the defect operating properly?

yes
no

A: no

Q: Is the fill gun tip near the defect opening and closing?

yes
no

A: yes

Q: Is the fill gun tip responding fast enough (within 1 second)?

yes
no

A: yes

Q: Are the flow controls open?

yes
no

A: yes

Q: Is the hopper shutoff opening during the fill cycle and responding fast enough?

yes
no

A: yes

Q: Check the vacuum on the fill gun. Is the vacuum on the fill gun at least 8" Hg at 60 psi?

yes
no

A: yes

Q: Check the fill air on the fill gun. Is it sufficient?

A pattern: 75 to 85 psi
B pattern: 85 to 95 psi
C pattern: 55 to 65 psi
D pattern: 70 to 80 psi

A: yes

Q: Is the fill tube attached and free from obstructions?

yes
no

A: yes

Q: Is there air leaking back into the hopper at any time other than step 5 (blowback)?

no
yes

A: yes

Q: Is the hopper shutoff open or partially open?

yes
no

A: no

R: The recommended corrective action is:

Check the hopper shutoff Mac valve.
-- or --
Replace the seal in the hopper shutoff.

Logic

ENTRY1	ex.inex	ex
ENTRY1	qual.ex	eu.fil
DFIL	chk.beads	yes
DFIL	man.blow	no
DFIL	fill.time	yes
DFIL	complete	no
DFIL	con.int	consis
DFIL	loc.glob	local
DFIL	loc.spec	fillgun
DFIL	fill.gun	no
DFILLGUN	tip.aut	yes
DFILLGUN	tip.quik	yes
DFILLGUN	flow.cont	yes
DFILLGUN	hpr.quik	yes
DFILLGUN	gun.vac	yes
DFILLGUN	fill.air	yes
DFILLGUN	fill.hose	yes
DFILLGUN	air.hppr	yes
DFILLGUN	hpr.shtoff	no
DFILLGUN	RESULT	mac.vlv.hpr
DFILLGUN	RESULT	rep.hpr.sl

The operator entered the expert system as an experienced operator. The system asked the operator about the scrap type, and the operator answered 'eu.fil' (experienced underfill). The expert system asked more and more specific questions in an attempt to refine the possible scrap cause.

Specifically, the expert system identified the scrap cause as a fill gun. The system further narrowed the search until it was determined that the cause was related to the bead hopper seal. In this case, the expert system gave the operator multiple corrective actions.

Critique

The experts were generally satisfied with the depth and breadth of the expert system in determining a fill scrap caused by a fill gun malfunction. The machine operator expert stated that "I would have played around more, but basically would have followed the same path." His indication was that he would have tried different higher probability scrap causes before arriving at the same conclusion as the expert system. While iterating through the expert system the expert noted another strength of the expert system - that it will point to other possible scrap types if the current type is not the problem. An example is: the expert system indicates an underfill scrap. The expert system points to overfusion if all questions indicating underfill scrap are answered negatively.

One of the experts had difficulty with the question "Is the fill gun operating properly?" A question like this should not be asked unless there is a list of more specific items to check, such as:

Check

1. fill tubes
2. fill tips

Is the fill gun operating properly?

Some other possible corrections to the expert system were: to screen the beads after they were blown back into the hopper (a fused bead plug may have been blown back into the hopper); and to check the vents around the fill gun if the scrap is localized to a fill gun but the gun is operating properly.

LEVELS OF EXPERTISE

There are different levels to which the expert system can be developed. The extent to which the system is developed is directly dependent upon the amount of time which is put into maintaining and improving the system. There are realistically six sequential steps in the development of the expert system.

1. New user trainer and reasonable accuracy as process correction. The system is used to train new operators in the manufacturing process. It works as an aid to help the operator successfully diagnose the scrap type and corrective action, as well as to think logically about the problem solving technique. At this time in the systems's development, it has a reasonable degree

of accuracy for process correction. However, the system could be updated, as needed, to refine the knowledge.

2. Remember low probability corrective actions. The system will remember low probability corrective actions that the operator or supervisor might not remember, or might not have seen before. An example is the contamination of the beads used to make the coffee cups caused by the infestation of cicadas every seventeen years.

3. Keep all operators at the same level as the best operator or supervisor. Assuming there is one expert, the knowledge base will be as good as the information supplied by this expert. Therefore, the expert system will be approximately equivalent to the best "expert."

4. An expert system that is better than any one operator. Eventually, all the operators will be brought to the level of the best operator, since that knowledge is easily accessible in the expert system. As the operators further develop in competence, they will gain knowledge in areas in which other operators are less competent, such as individual machine-dependent scrap causes. As these "pockets" of information are gained by the operators, they can be coded into the expert system, making the system "smarter" than any one operator.

5. Anticipate process corrections. In the long term, the expert system will anticipate scrap problems before they occur. In other words, it will be a preventive, rather than a corrective system.

6. Closed loop system. In the very long term, it may be possible for the system to be completely closed loop. The system would correct itself once it had determined it had, or was about to, produce scrap.

SUMMARY

The final summary of the thesis will briefly describe the overall reaction of the evaluators to the expert system, and the status of the expert system when the prototype was installed in the manufacturing plant.

The evaluators were satisfied, in general, with the overall breadth and depth of the system, especially when compared with earlier versions. Typically, the expert system followed a sequence of questions similar to the expert's train of thought, and asked the expert to evaluate appropriate machine conditions. As with any new technology, the experts offered numerous suggestions for improvement. Many of the suggestions are mentioned in the above discussion and are currently being implemented.

A prototype version of both the qualitative and quantitative portions of the expert system is installed on the plant floor and is currently being debugged. A thorough evaluation of the system was originally planned for late-July to early-August. The evaluation will determine the amount of resources that the plant will assign to maintaining and improving the expert system.

The evaluation of the system will be delayed until: the communication problems with the cell controller are eliminated; and lost foam production stabilizes (the production area which is aided by the expert system was shut down for 6 to 8 weeks beginning July 6, 1987).

APPENDIX

APPENDIX A - ID-3 ALGORITHM

The ID3 algorithm is an algorithm commonly used to efficiently prioritize attributes based upon examples in an example-based expert system. It was historically used for chess end games and is based on a statistical property known as entropy. The work presented is this appendix is based upon papers by Thompson and Thompson [5], and Michelski, Carbonell, and Mitchell [12].

Entropy is described as a measure of the uncertianty of the classification of an object that can be classified into several different groups. In other words, the attribute with the lowest entropy, or greatest certianty, should be chosen as the attribute for splitting in a decision tree.

Say, we have N classes: $c_1, c_2, \ldots, c_N$; entropy is defined as:

$$H(C) = - \sum_{i=1}^{N} p(c_i) * \log_2 p(c_i)$$

For this discussion, let's say we have 10 pieces of data with the folowing attributes and results:

	Humidity	Temperature
1.	up	hot
2.	up	hot
3.	down	ok
4.	up	cold
5.	up	hot
6.	down	cold
7.	up	ok
8.	down	cold
9.	down	ok
10.	up	hot

So the entropy is:

$$H(C) = H(\text{hot, cold, ok}) = - \sum_{i=1}^{N} p(c_i) * \log_2 p(c_i)$$

$$= - p(\text{hot}) * \log_2 p(\text{hot}) - p(\text{cold}) * \log_2 p(\text{cold})$$

$$- p(\text{ok}) * \log_2 p(\text{ok})$$

$$= - 0.4 * \log_2 (0.4) - 0.3 * \log_2 (0.3) -0.3 * \log_2 (0.3)$$

$$= 1.57$$

This number represents the uncertianty about the temperature being hot, cold or ok. It doesn't yield any information about the attributes used to arrive at these conclusions.

Therefore, the entropy if classification, given an attribute is defined as:

$$H(C|a_j) = - \sum_{i=1}^{N} p(c_i| a_j) * \log_2 p(c_i| a_j)$$

where:

$p(c_i|a_j)$ is the probability that the class is c_i, given the attribute is a_j.

So entropy of classification is:

$$\begin{aligned}
&H(C|\text{humidity} = \text{up}) \\
&= - p(\text{hot}|\text{humidity} = \text{up}) * \log_2 p(\text{hot}|\text{humidity} = \text{up}) \\
&\quad - p(\text{cold}|\text{humidity} = \text{up}) * \log_2 p(\text{cold}|\text{humidity} = \text{up}) \\
&\quad - p(\text{ok}|\text{humidity} = \text{up}) * \log_2 p(\text{ok}|\text{humidity} = \text{up}) \\
&= - 0.67 * \log_2 (0.67) - 0.17 * \log_2 (0.17) \\
&\quad - 0.17 * \log_2 (0.17) \\
&= 1.26
\end{aligned}$$

And,

$$\begin{aligned}
&H(C|\text{humidity} = \text{down}) \\
&= - p(\text{hot}|\text{humidity} = \text{down}) * \log_2 p(\text{hot}|\text{humidity} = \text{down}) \\
&\quad - p(\text{cold}|\text{humidity} = \text{down}) * \log_2 p(\text{cold}|\text{humidity}=\text{down}) \\
&\quad - p(\text{ok}|\text{humidity} = \text{down}) * \log_2 p(\text{ok}|\text{humidity} = \text{down}) \\
&= - 0.0 * \log_2 (0.0) - 0.50 * \log_2 (0.50) \\
&\quad - 0.50 * \log_2 (0.50) \\
&= 1.00
\end{aligned}$$

So, in order to find the entropy of all of the examples after the split using humidity, or:

$$H(C \mid \text{humidity})$$

we take the sum of the entropy of each of the values of the attribute multiplied by the probability that the value will appear in the examples.

$$H(C \mid A) = H(C \mid \text{humidity}) = \sum_{j=1}^{M} p(a_j) * H(C \mid a_j)$$

$$= p(\text{humidity = up}) * H(C \mid \text{humidity = up})$$

$$+ p(\text{humidity = down}) * H(C \mid \text{humidity = down})$$

$$= 0.60 * 1.26 + 0.40 * 1.00$$

$$= 1.16$$

Now, if we perform the same calculations for different attributes in the examples, say wind speed, and find:

$$H(C \mid \text{wind speed}) = 0.73$$

Since the entropy of wind speed is 0.73, and the entropy of humidity is 1.16, we would use wind speed as the initial split in the decision tree because it has a lower entropy, and therefore would decrease the uncertianty of the final classification.

LIST OF REFERENCES

LIST OF REFERENCES

(1) P. H. Winston. Artificial Intelligence, Second Edition. Reading, Massachusetts: Addison-Wesley, 1984.

(2) R. C. Schank. The Cognitive Computer. Reading, Massachusetts: Addison-Wesley, 1985.

(3) F. Hayes-Roth, D. A. Waterman, and D. B. Lenat. Building Expert Systems. Reading, Masachusetts: Addison-Wesley, 1983.

(4) S. R. LeClair, "Sensor Fusion: The Application of Artificial Intelligence Technology to Process Control", in 1986 Rochester FORTH Conference, Rochester, NY, June 11-14, 1986.

(5) B. Thompson, W. Thompson, "Finding Rules in Data", BYTE, November, 1986, pp 149-354.

(6) R. Brachman, "I Lied About the Trees", AI Magazine, Fall, 1985, pp 80-93.

(7) J. M. Hunt. Machine Diagnostics for the Inline Pattern Assembly Machine and Pattern Molding Machines. Thesis for GMI Engineering & Management Institute, submitted: June, 1987.

(8) W. F. Kaemmerer, P. D. Christopherson, "Using Process Models with Expert Systems to Aid Process Control Operators", in Proc. of the American Control Conference, June 19-21, 1986, pp 892-897.

(9) B. Thompson, W. Thompson, "Creating Expert Systems from Examples", AI Expert, January, 1987, pp 21-26.

(10) C. W. Lee, "Composite Cure Process Control by Expert Systems", in 1st Conference on Composite Materials, American Society for Composites, Dayton, OH, October 7-9, 1986, pp 187-196.

(11) A. V. Aho, J. E. Hopcroft, and J. D. Ullman. Data Structures and Algorithms. Reading, Massachusetts: Addison-Wesley, 1982.

(12) R. Michalski, J. Carbonell, and T. Mitchell. Machine Learning: An Artificial Intelligence Approach. Los Altos, California: Kaufmann Publishers.

MICHIGAN STATE UNIV. LIBRARIES
31293008006433