Artificial Intelligence in Power System Operations

BRUCE F. WOLLENBERG, SENIOR MEMBER, IEEE, AND TOSHIAKI SAKAGUCHI, MEMBER, IEEE

Invited Paper

Power system operators often reach a cognitive barrier when information arrives too fast during a power system emergency. At such times it becomes difficult to reach a correct diagnosis of the problem or to formulate the correct decision when actions must be taken. Artificial Intelligence gives designers of Energy Management Systems a way to solve many of the diagnosis and decision problems so as to make the EMS more useful. This paper explores reasons why AI techniques, such as knowledge-based expert systems, are being used in EMS designs and the differences between knowledge-based expert systems and traditional numeric algorithm development. The differences between expert systems and the numeric approach extend to the basic conception and design of the applications. This is illustrated using a relay fault diagnosis system, showing both the traditional and rapid prototyping approaches to its development. Finally, issues concerned with the implementation of AI in EMS computers are explored along with the authors' predictions of possible AI applications to power system operations.

I. Introduction

Modern power systems are operated by highly skilled operators through computerized control systems. The energy management system (EMS) is the center of a control system organized in a hierarchical structure utilizing remote terminal units, communication links, and various levels of computer processing systems. The function of the EMS is to ensure the secure and economic operation of the power system as well as to facilitate the minute-by-minute tasks carried out by the operations personnel. The EMS is mainly designed to be used in the "normal" state where such functions as state estimation, security analysis, and optimal power flow are used to ensure secure operation while functions such as automatic generation control, economic dispatch, unit commitment, and load forecasting are used to ensure that the most economic operation is obtained. Much of what happens in normal operation is now computerized and human operators only intervene to carry out the few manual tasks required.

Manuscript received November 17, 1986; revised May 10, 1987.
B. F. Wollenberg is with Control Data Corporation, Minneapolis, MN 55441, USA.
T. Sakaguchi is with the Central Research Laboratory, Mitsubishi Electric Corporation, 1-1, Tsukaguchi Honmachi 8 Chome, Amagasaki, Hyoga 661, Japan.
IEEE Log Number 8717877.

The picture is quite different, however, when we look at the use of an EMS during an unforeseen event or a failure of major components on the power system. In such instances, the EMS serves mainly as an information gathering and reporting system and the sophisticated application software that functions in normal operation may be of little use. For example, in a regulatory shutdown of all nuclear units, human operators will take over the economic dispatch and unit commitment to reschedule economic operation. Similarly, when a sudden loss of transmission equipment occurs it is human operators who must understand what has happened and decide on what actions to take. It is especially during such emergencies that conventional software is less effective. The requirement for smarter software thus becomes more important in such instances.

Coping with emergency events is referred to as a diagnosis and decision process. Such processes are illstructured and their solution rests heavily on the experience and skill of the human operators to react correctly. The key to human ability in such situations lies in the experience with similar events and the use of heuristics to map existing complex situations onto learned past events to solve problems. Since Artificial Intelligence allows the realization of heuristic techniques in a computer, the way is now open for many new applications of computers in power system operations.

There is also a need to incorporate heuristics in many of the functions now carried out by large application programs. For example, security analysis programs may fail to converge or economic analysis programs may fail to meet certain operating constraints because the constraints are difficult to express mathematically. In such cases, human intervention must be relied on to solve or circumvent the problem. Research using artificial intelligence is also beginning to yield methods of embedding complex heuristics into conventional application software for use in normal power system operations.

In this paper we are specifically addressing the use of expert or knowledge-based systems which is one of the principal branches of artificial intelligence. Several experimental and near-practical-use expert systems have been developed for use in power system operations. Sakaguchi

0018-9219/87/1200-1678$01.00 © 1987 IEEE

and Matsumoto [1] discuss the use of an expert system to capture heuristic knowledge from operators and operations manuals that direct the steps taken in restoring power after a large system failure. Wollenberg [2] and Larson *et al.* [3] describe an experimental expert system to intercept alarm messages and present the operator with a summary of the most pertinent information. The authors in [4]–[8] discuss the use of expert systems in application programs.

Power system operations is not the only real-time control environment where expert systems are being applied. References [9]–[11] describe the use of expert systems in process control applications and [12] describes an experimental expert system to aid nuclear plant operators in diagnosing reactor shutdown problems. Several common features can be found in these systems. First, they normally require human operators 24 h a day and this gives a great advantage to an expert system that can provide the right expertise at any hour and any day of the week. Secondly, the expert system can potentially provide a rapid reaction to emergency events by summarizing information quickly and checking many more applicable rules than a human operator could in the same period of time.

This paper begins by analyzing the diagnosis and decision process in power system operations and discusses how this process might benefit from the use of expert systems. After describing typical examples, efforts are made to define what the AI techniques are and how they differ from numeric techniques such as linear and nonlinear programming. The strength of AI over conventional techniques and how knowledge-intensive problems are solved is also discussed. Early experiences so far have indicated that implementation of AI in EMS computers poses a difficult problem. Several alternate implementation schemes are proposed and explored. Finally, some predictions about future uses of AI in EMS are made and some needed technological innovations are listed that are required before AI can deeply penetrate power system operations.

II. Diagnosis and Decision Processes in Power System Operations

Is there a need for knowledge-based software in power systems operations? We believe there is. A fundamental motivation for such software is the need to overcome the human cognitive barrier which conventional EMS installations encounter during emergency operation or when application programs are used beyond their design limitations.

The cognitive barrier is felt as the complexity of power system operations increases without sufficient efforts to cope with it. This is true of today's EMS installations where the quantity of data gathered and the rate at which they are gathered can overwhelm a human operator. It must be noted that the driving force in EMS complexity is the desire to operate the power system closer to its limits so as to make better use of generation and transmission facilities. This, in turn, has made a qualitative change in system operations requiring quicker diagnosis and decision making by operators. Fig. 1 illustrates this situation. While system complexity increases steadily, the operator's ability to cope with it decreases. Since the complexity of power system operations is very likely to continue to increase in the future,

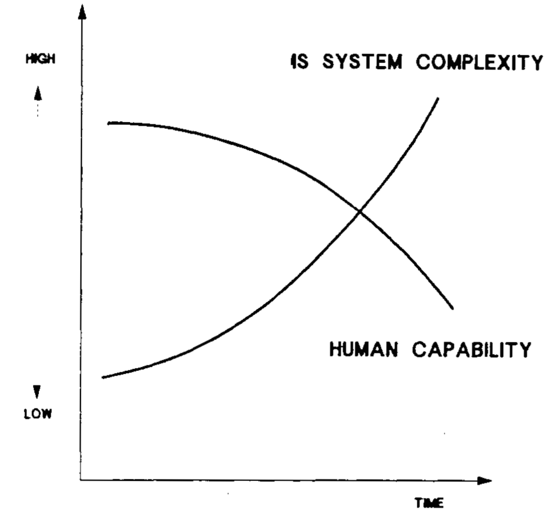

Fig. 1. The risk of enlarging the human cognitive barrier.

there is a risk of human operators being unable to manage certain functions unless their capability is enhanced.

As indicated earlier, the cognitive barrier is quickly realized in power system operations when sudden and unforeseen events occur. When human operators meet such an event, they have to understand the situation (diagnosis) and determine actions (decision) to return the system to normal. As all the tasks have to be done in real time, the operators are exposed to heavy mental stress and this makes the cognitive processes distinctively different from that experienced by others. For example, experienced designers of large-scale integrated circuits often face similar cognitive barriers but solutions are not required in real time.

There are several ways to help operators overcome this cognitive barrier. First, operators need to understand what is happening on the power system and the AI software can give guidance by showing various scenarios that explain the situation consistently. Operators can then check for the most plausible scenarios, some of which may have been overlooked. Similar guidance can be expected in using large application programs where the AI system can guide the operator in its use. Further, since power system operations are filled with many fragmented tasks that are done almost routinely, smart software to do such routine tasks could relieve operators and allow them to devote their time to more important tasks.

III. Knowledge Intensive versus Numeric Problems

An expert system is a software paradigm where knowledge concerning a complex problem is encoded into a computer program. The framework of expert systems is designed to enable easy encoding of knowledge and easy checkout of the expert system's performance. A general architecture for expert systems is shown in Fig. 2. Four major software elements comprise an expert system: the knowledge base, an inference engine, building and checkout utilities, and the user interface.

In order to use a knowledge base on a computer some facilities are needed to read a module from the knowledge base, decide whether it is to be executed, and to carry out the execution. The inference engine is responsible for this task and functions much like an interpreter for conven-

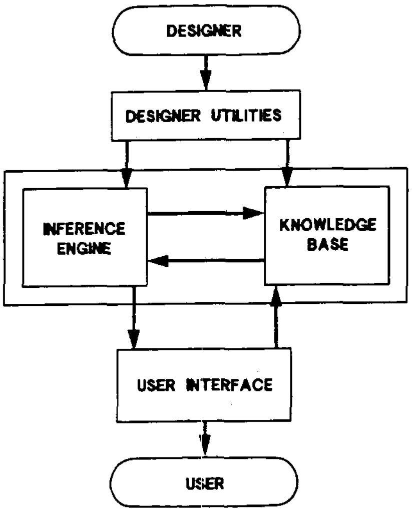

Fig. 2. Architecture of an expert system.

tional programs. The actual execution or computation in the inference engine, however, is quite different from an interpreter. The other software elements are for the convenience of the developer and user of the software. Typical expert system programs are very large and the complexity of the knowledge base necessitates advanced editors and browsing tools to debug them. Expert systems also provide the ability to explain the reasoning used (e.g., to trace the rules used in a rule-based system) which is important in checking it out.

Depending on the representation scheme, an AI program becomes either rule-based, frame-based, or logic-based as shown below.

A. Rule-Based System

The popular OPS5 expert system, [13], will be used as an example. A piece of knowledge is represented in a form called a production rule, which has a premise-part (IF-clause) and an action-part (THEN-clause) as shown below:

(p rule_name
  ⟨premise-part⟩ → ⟨action-part⟩)

The rule-based system has two kinds of memory: short-term (or working memory) and long-term. The short-term memory (STM) contains factual knowledge, to be modified as the computation proceeds. The long-term memory (LTM) contains the production rules themselves. The inference engine of the rule-based system tests the premise-part by matching it against the factual knowledge in the STM (matching cycle). If it succeeds, the action-part of the rule is executed resulting in some changes to the STM (firing cycle). The engine then goes back to the matching cycle. There may be more than one rule which succeeds in matching and the inference engine then invokes a conflict resolution mechanism to decide which rule shall be used.

B. Frame-Based System

In the rule-based system, factual knowledge is stored in the STM without regard to relationships between different objects. There are, however, relations between the objects of many problems and a frame-based knowledge representation allows the user to set up and make use of these relationships.

For example, consider the objects of a substation such as breakers, switches, buses, transformers, and transmission lines. Several objects comprise a substation, and a set of substations becomes an area. Depending on the status of individual breakers and switches, buses may be split or de-energized. Transformers and lines may be connected, open-ended, or de-energized depending on the status of the terminating bus sections, etc. This is illustrated by the following set of frames:

```
SUBSTATION FRAME:
  Name:           character string
  Breakers:       (pointer to breaker frame)
  Bus sections:   (pointer to bus section frame)
  Lines:          (pointer to line frame)
  Transformers:   (pointer to line frame)

BREAKER FRAME:
  Name:                     character string
  Duty rating:              real constant
  Terminal bus sections:    (pointers to bus section objects)
  Status:                   open/closed/unknown
                            determination: access-breaker-data
LINE FRAME:
  Name:                     character string
  Rating:                   real constant
  Terminal bus sections:    (pointers to bus section objects)
  Status:                   connected/open-ended/de-energized
                            determination: line-status-algorithm

BUS SECTION OBJECTS:
  Name:                     character string
  Voltage class:            real constant
  Status:                   energized/de-energized/unknown
                            determination: bus-section-algorithm
```

This type of structure allows an expert system that knows something (i.e., some object has been set to a value in the STM) about some part of the substation (say, the status of a breaker) to use the frame relationships to directly infer conditions of other objects in the substation (say, the status of a line). The Automatic Reasoning Tool (ART) expert system language [14] is an example of this type of structure.

C. Logic-Based System

The frameworks we have dealt with so far are appropriate to represent procedural knowledge such as: what to do when certain conditions are met. A different way to represent knowledge requires one to specify "what" instead of "how." A logic-based system provides us with such means. Prolog is a programming language to represent a "what"-type knowledge based on predicate calculus [15].

For example, the following Prolog statement might be used in an expert system to guide operators in system restoration:

```
restoration_required( X, yes ) ←
  component( X ),
  charge_state( X, Y ), =/=( Y, is_charged),
  fault_state( X, Z ),    =/=(Z, is_faulted).
```

This example states what must be true for object X in order for it to have a restoration requirement of "yes." The statement says that X must be a component and that X is not charged and is not faulted.

Logic-based systems have an advantage when specifying system requirements, but they have a disadvantage in specifying procedure-oriented knowledge.

D. How Expert Systems Differ from Numeric Methods

A good illustration of the difference between expert systems and numerical methods occurs when one tries to apply both techniques to the power system restoration problem. In the case of expert systems, a set of rules which govern the procedures (or specify conditions to be met in the case of logic-based systems) is written. The inference engine then uses the rules to find a sequence of switching actions to restore operation to the system. The same problem can be posed as an integer programming problem and solved using a general-purpose integer programming routine. Two important points must then be considered, the generality and the framework of the solution.

Generality has advantages and disadvantages. The fact that a technique such as integer programming can be applied to this problem may not be of much benefit since the feasible solution space is very large and the restoration problem does not meet the convexity assumptions needed to guide the selection of alternative solutions. Further, it is very difficult to define a criterion for optimality and, therefore, the framework within which a numeric method can be applied is limited.

The inadequacy of numerical techniques is true for many other functions needed in power system operations. Human operators, when presented with problems, recognize the situation and decide on a course of action in an all-encompassing way, implicitly taking account of many factors. Thus we believe that the best path to solve ill-structured power system operating problems is to construct a knowledge-based program that emulates human operators.

The strength of AI techniques over conventional programming can be summarized as follows:

1) *Flexibility:* Expert systems are suited to solving ill-structured problems. Furthermore, the environment used to construct expert systems allows them to be prototyped rapidly and incrementally so that many alternatives and frequent updatings can be tried.

2) *High Performance:* Expert systems try to implement the level of performance exhibited by a person with recognized expertise in the problem domain.

3) *Understanding:* Expert systems can explain the line of reasoning used as well as the contents of the knowledge base. This is a key element when the designer debugs the system and increases the confidence of the user.

IV. Development of an AI Application

The development of AI programs usually involves different procedures than used in the development of numeric programs. To illustrate this difference we review an application that deals with the diagnosis of power system faults and is constructed using both numeric and AI programs.

The problem is to identify a faulty element in a power system by observing the relay and circuit breaker tripping signals. Fig. 3 illustrates a rough outline for a conventional

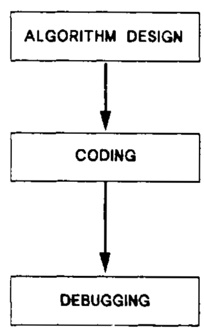

Fig. 3. The "pipeline model" of software development.

program development sequence and is referred to as the "pipeline model." This term was chosen because conventional program development usually proceeds in a fairly rigid sequence of steps. As with the restoration problem discussed earlier, no algorithm exists for this problem. Initial efforts are made to design an algorithm for diagnosis. A document, written in unexecutable natural language and flow charts, is completed at the end of the first phase. The following serves as such a sample document for system fault diagnosis:

Fault diagnosis is made based on the relay and circuit breaker tripping signals. First, de-energized circuit elements are grouped into several areas that are topologically disconnected. Then, the following diagnostic algorithm is applied to each area.

Determine protected elements of all relays which operated in one area and calculate their conjunction. Three cases are possible:

1) If a single element comprises the conjunction, then assume the faulty element to be in that area.

2) If two or more elements are included in the conjunction assume that some relays have not operated and designate one element within the area as the probable faulty element. Check the consistency among

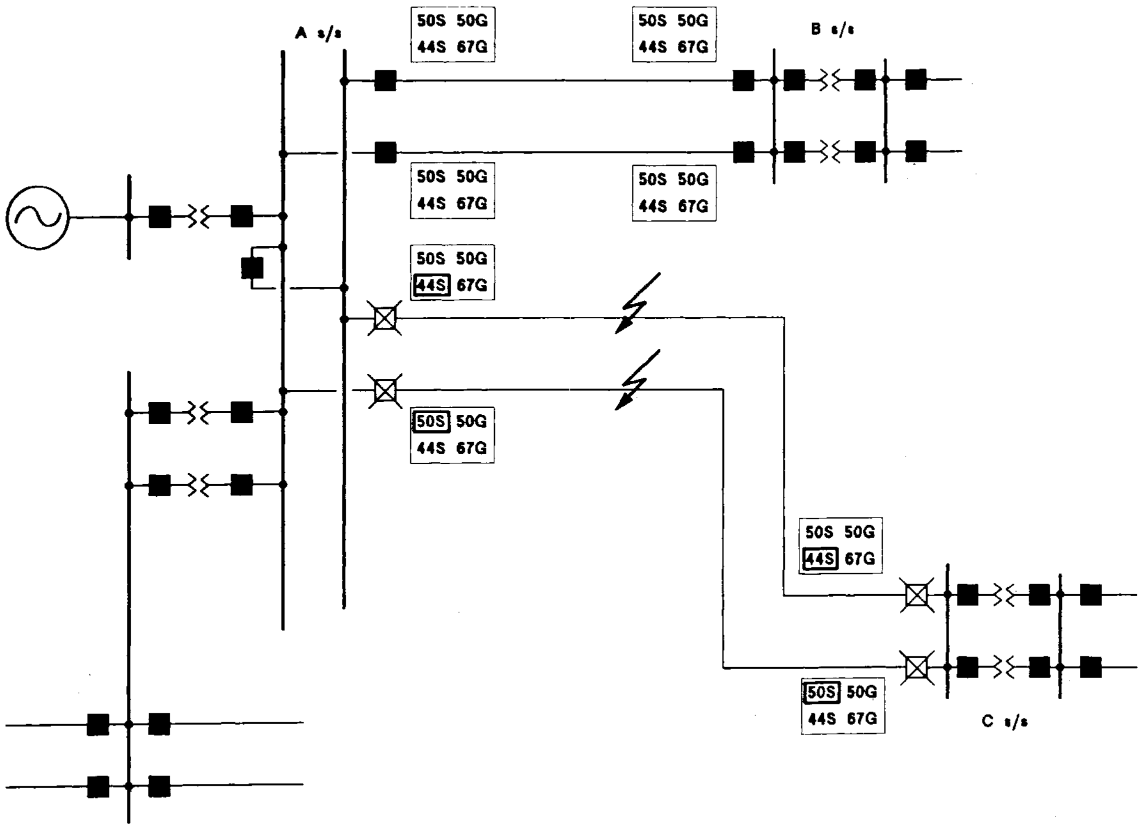

Fig. 4. Power system fault problem.

the assumptions, possible unoperated relays, and measurements until the designated element is accepted or eliminated. If eliminated try another element.

3) The third case involves no elements in the conjunction and results from more than double faults within that area or some relay misoperations. Find a set of active relays so that the set of protected elements has a nonempty conjunction. If successful, apply the diagnosis algorithm to each set, otherwise assume relay misoperation.

Now coding the algorithm is begun. The algorithm might be written in a system description language and its source code compiled into Fortran source code. This is followed by the usual compile, load, link, and run sequence which requires repeated changes to the program source code and may even require changes to the system description itself.

The human resources to accomplish this task are estimated to be eight man-months for the first phase, four man-months for the second, and six man-months for the last. The total amounts to one and a half man-years. The final Fortran source code is estimated to be about 15 thousand lines and will run in about 10 s on a 32-bit process computer to diagnose a fault as shown in Fig. 4.

The major human resource is needed during the design and debugging of the algorithm. This is easily understood since the diagnosis algorithm cannot be described in a compact way. As it is an ill-structured problem, one could not finish a design document which specified the algorithm completely. This results in missing specifications, incorrect code, and frequent modifications as the design becomes clearer. This problem is made worse by the fact that numeric programming languages are less readable and less understandable when applied to knowledge-intensive problems.

The experience in applying an expert system to the same problem is quite different. The software development paradigm is illustrated in Fig. 5 and is called the "rugby model."

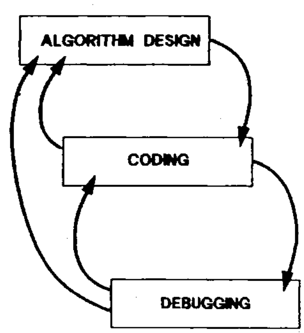

Fig. 5. The "rugby model" of AI software development.

As in the pipeline model shown in Fig. 3, there are design, coding, and debug phases. However, they are strongly interrelated in the rugby model and rigid sequencing of phases is not intended. The output of the design phase is not a document but a knowledge base that can be executed "as is" on the computer. In other words, the goal of the first phase is to prototype the algorithm as quickly as possible (rapid prototyping). As it is executable on a computer, both user and developer can observe how it works at an early stage of the project. If the prototype meets requirements, the knowledge base is transported to the target machine without changes. If it does not, some changes will be made during its transport to the target machine.

The actual human resource needed to apply expert system techniques to the fault identification problem was three quarters of a man-year. Most of this time was spent in the first phase. A sample of the knowledge base written in the OPS5 rule base language is shown in Fig. 6 along with a natural language description of the rules.

```
(p rule_for_local_backup_protection
        o
        o
        o

   (protection  ^relay_type      dz
                ^pro_type        local
                ^id              <x1>
                ^location_at     <y1>
                ^object          <z>
                o
                o                       )
   (protection  ^relay-type      dz
                ^pro_type        local
                ^id              <x2>
                ^location_at     { <y2> <> <y1> }
                ^object          <z>
                o
                o                       )
   (make        fault_information
                ^main_relay      nil
                ^backup_relay    dz
                ^spec            local_backup_protection
                ^fault_at        <z>
                o
                o                       )
   )
which means:
   "If the back-up relay <x1> of type dz at the location <y1>
       that protects the element <z> is operated and
       the back-up relay <x2> of type dz at the location <y2>
       that protects the element <z> is operated and
       <y2> is not equal to <y1>,
    then the fault is concluded at the element <z> and
       it is cleared by local back-up protection."

Note:  The following notation is assumed.
       (1)  All syntax follows the OPS5 language [13].
       (2)  An attribute is preceded by the symbol " ^ ".
       (3)  A variable is quoted by the symbols < and >.
```

Fig. 6. Rule-based representation of diagnosis knowledge and its meaning in natural English.

Although it is still early in the development, experience in applying AI technology to the diagnosis process has shown benefits to both utility companies and manufacturers. To utilities, the new software promises advanced automation of power system operations never possible before. To the manufacturer AI has great potential in helping to control the costs of an ever-increasing demand for more complex software.

V. Implementing AI in Energy Management Systems

Implementing AI in an EMS is more difficult than adding a new application program written in an engineering language such as Fortran. AI programs are generally written in special languages because of the needs for symbolic processing and in some cases these languages require special hardware.

Most AI development work is currently done in Lisp or Prolog. However, it is unclear at this time whether these languages are the best to use in implementation of AI into EMS systems. As discussed in [16], there are basically two approaches that can be taken:

1) Implement AI programs in languages such as Pascal or C that contain many of the necessary language features needed in AI programming. This is illustrated in Fig. 7(a). The alternative illustrated in Fig. 7(b) would implement the AI programs directly in Lisp on conventional EMS hardware. This approach suffers, however, in that such hardware is often not efficient at running programs written in Lisp.

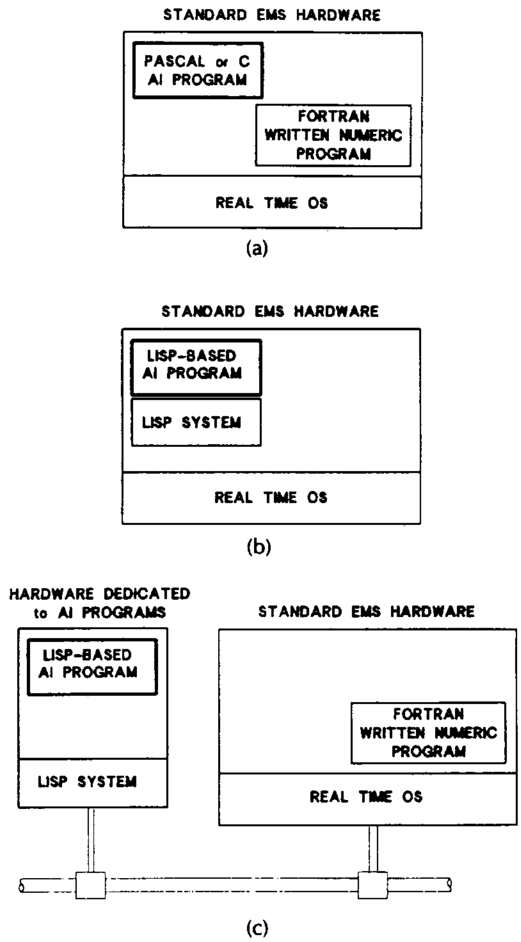

Fig. 7. Implementing AI in an EMS system.

2) Attach special hardware to the EMS that runs Lisp or Prolog efficiently as illustrated in Fig. 7(c).

The decision is further made difficult by the fact that AI development thrives best in a "prototyping" environment which may not necessarily be the same environment as the "delivery" environment.

Going with the first approach allows the AI programs to interface directly with existing database, display, application, and communication software but requires rewriting (or translating) the programs if the development language is different from the delivery language. The second approach eliminates the rewriting or translating but adds the problems of communications between two different processors.

VI. Possible Applications of AI to Power System Operations

References [16] and [17] list various power system operations problems as potential areas for the application of AI in energy management systems. Talukdar and Cardozo [17] make a subjective classification of the various problem areas according to one of four metrics: operating cost savings, capital cost savings, improved quality of service, and generalizability. In this paper we make our own predictions as to the problems most needing AI attention and these are gathered into three groups labeled real-time control, operations planning, and operator training.

A. Real-Time Control Problems

1) Alarm Processing: The alarm processing problem is really an extension of the diagnosis problem. When a serious disruption occurs on the power system, operators can be overloaded with alarm messages. Because many of the alarm messages are redundant or present information related to the same event the operators may have difficulty in understanding precisely what has happened. The use of AI to intercept alarm messages and present a concise diagnosis is now under active development in several organizations, see [2], [3], [18]–[20].

2) Switching Operations: Statistics show that about 40 percent of the tasks at a power system control center are related to operations on circuit breakers and line switches. Therefore, the automation of these tasks should benefit system operators. One potential application is the automatic generation of switching sequences. Some work has been done on verification of the switching sequences, [21]. Another application is the identification and isolation of faulted line sections as shown in [22] and [23].

3) Voltage Control: Incorporation of static optimization techniques such as an Optimal Power Flow (OPF) is common for new control centers which desire to control the system voltage profile. However, the control actions recommended by the OPF do not take account of the future load prediction or past history of control actions and may prove very difficult to implement since many of the controls require manual entry by the operator. Liu and Tomsovic [5] address this problem.

4) Restoration Control: A large-scale blackout may happen on a power system, although quite infrequently. The fact that blackouts happen infrequently makes the operator's job that much harder because of the limited exposure to solving the problem of restoring the system. As a result, most control centers have restoration plans and attempt to train operators in restoration using training simulators. However, the number of possible ways to restore a power system is very large and can change depending on the state of critical components at the time the blackout occurs. To this end, a system which supports operators by giving them timely guidance and provides them with a tool for short-term operations planning is quite desirable. As shown in [1], AI software is essential in constructing such a system.

B. Operations Planning

5) Load Flow Planning: Load flows are run by system operators to determine effects of planned changes to the system and to help the operator study appropriate alternatives should equipment loading fall outside appropriate operating limits. An intelligent and friendly interface to the load flow program will help the operator in setting up cases to be run, interpreting the results of solved cases, and especially in how to interpret results if the load flow fails to converge. Fujiwara *et al.* [6] describe an early effort which proved effective in developing an intelligent load flow interface.

6) Unit Commitment: One of the problems encountered in the use of unit commitment programs is the difficulty of expressing all the constraints that operators must meet in scheduling units. Present practice in many control centers with unit commitment programs is to run the program and then alter the resulting schedule to meet constraints not included in the program. Mokhtari *et al.* [24] describe an expert system that was developed to aid operators in adjusting the input data to the unit commitment program so that the resulting schedule meets all scheduling constraints in an optimal way.

C. Operator Training

7) Personal Tutoring: Large-scale training simulators are installed in power system control centers to enhance operators' skills, [25]–[28]. One point of view states that the operator acquires these skills through the efforts of classroom instruction and over-the-shoulder advice of a training instructor while solving difficult operating problems on the simulator. Another point of view adds the capability of having the training simulator provide custom-tailored advice for a specific operator and a specific training situation. The authors of [29] and [30] describe such training facilities using AI techniques.

8) Scenario Building: Another aspect of operator training is the need to provide adequately difficult training scenario cases for the training sessions. These scenarios need to be made difficult enough and specific enough so that targeted levels of skill can be reached in each aspect of power system operation. Building such scenarios can be quite difficult for training instructors and the research reported in [31] describes an expert system to allow the instructor to build a scenario given a specific level of difficulty for the training exercise and the type of problem that is to be presented to the operator.

VII. Conclusions

The need for the application of AI technology to power system operations has been analyzed. Some initial work done in this area is reviewed to show how it differs from and what its strengths are over conventional numeric programs. We believe that this growing technology will have a significant impact on future EMS design and will allow a level of system automation unattainable with present techniques. However, several technological barriers have to be surmounted before this takes place.

The inference mechanisms that we foresee operating in an EMS must perform at very high speed to be useful in a real-time environment. In the short run, this high performance will be accomplished by improvements in the design of software. In the long run, new computer architectures will be developed that more closely match the needs for AI programs. In addition, we recognize that the usefulness of AI programs depends strongly on the quality of the knowledge base built into the system. Thus we must be able to acquire knowledge efficiently to be able to take the best advantage of AI. The automatic acquisition of such a knowledge base is discussed in [32] but its state is still primitive and knowledge engineers skilled in this area are absolutely necessary.

The knowledge of how to run a power system is resident in the engineers and managers of the electric utility companies while the computer technology to build an AI system belongs to suppliers. Therefore, close cooperation and joint development projects are absolutely necessary to reach successful implementations. Finally, people having the multidisciplinary skills of power system engineering, computer science, and cognitive science must be trained for such tasks.

References

[1] T. Sakaguchi and K. Matsumoto, "Development of a knowledge-based system for power system restoration," *IEEE Trans. Power App. Syst.*, vol. PAS-102, no. 2, pp. 320–329, Feb. 1983.

[2] B. F. Wollenberg, "Feasibility study for an energy management system intelligent alarm processor," in *Proc. 1985 Power Industry Computer Applications Conf.*, pp. 249–254, May 1985.

[3] L. R. Larson *et al.*, "The NSP intelligent alarm processor project," in *Proc. 1985 Minnesota Power Systems Conf.* (Univ. Minnesota, Minneapolis, Oct. 8–10, 1985).

[4] A. Doi and K. Uemura, "Knowledge-based operation guidance method for voltage and reactive power systems," presented at the IFAC Symp. on Planning and Operation of Electric Energy Systems, July 1985.

[5] C. C. Liu and K. Tomsovic, "An expert system assisting decision-making of reactive power/voltage control," in *Proc. 1985 Power Industry Computer Applications Conf.*, pp. 242–248, May 1985.

[6] R. Fujiwara *et al.*, "An intelligent load flow engine for power system planning," in *Proc. 1985 Power Industry Computer Applications Conf.*, pp. 236–241, May 1985.

[7] T. Sakaguchi *et al.*, "Design and development of human interface in engineering workstation for power system planning," presented at the IFAC/IFIP/IFORS/IEA 2nd Int. Conf. on Analysis, Design and Evaluation of Man–Machine Systems, Sept. 1985.

[8] Y. Kohno *et al.*, "An intelligent support system for power system planning," in *Proc. 1985 Int. Symp. on Circuits and Systems*. Also in *IEEE Trans. Power Syst.*, vol. PWRS-1, no. 2, pp. 67–75, May 1986.

[9] R. L. Moore *et al.* "A real time expert system for process control," in *Proc. 1st IEEE Conf. on Artificial Intelligence Applications*, IEEE publ. CH 2107-1/84, pp. 519–576.

[10] T. E. Murphy, "Developing expert system applications for process control," in *Proc. 1985 ISA Conf.* (Philadelphia, PA).

[11] P. A. Sachs *et al.*, "Escort—An expert system for complex operations in real time," *Expert Syst.*, vol. 3, no. 1, Jan. 1986.

[12] D. G. Cain, "BWR shutdown analyzer using artificial intelligence techniques," Rep. NR-4139-SR, Electric Power Research Inst., July 1985.

[13] C. L. Forgy, "OPS5 user's manual," Tech. Rep. CMU-CS-81-135, Dep. Comput. Sci., Carnegie-Mellon Univ., Pittsburgh, PA, July 1981.

[14] B. D. Clayton, "ART programming tutorial," Inference Corp., 1984.

[15] R. A. Kowalski, *Logic for Problem Solving.* Amsterdam, The Netherlands: North-Holland/Elsevier, 1979.

[16] R. P. Shulte *et al.*, "Artificial intelligence solutions to power system operating problems," paper 86 SM 335-4, presented at the 1986 IEEE Power Engineering Society Summer Meet., Mexico City, Mexico.

[17] S. N. Talukdar and E. Cardozo, "Artificial intelligence techniques for power system operations," Rep. EL-4323, Electric Power Research Inst., Jan. 1986.

[18] H. Kaninsono, "Alarm handling by control and load dispatching centers in Japan," CIGRE Study Committee Rep. 39, Sept. 1986.

[19] G. DeMontravel, "A real time expert system for alarm handling in the future EDF regional control centers," CIGRE Study Committee Rep. 39, Sept. 1986.

[20] F. Hein, "Expert system using pattern recognition by real time signals," CIGRE Study Committee Rep. 39, Sept. 1986.

[21] K. Matsumoto, T. Sakaguchi, and K. Uemura, "Verification of switching operations with temporal logic," in *Proc. 8th Power System Computation Conf.*, pp. 1096–1102, Aug. 1984.

[22] C. Fuki and J. Kawakami, "An expert system for fault section estimation using information from protective relays and circuit breakers," *IEEE Trans. Power Delivery*, vol. PWRD-1, no. 4, pp. 83–90, Oct. 1986.

[23] K. L. Tomsovic *et al.*, "An expert system as a dispatchers' aid for the isolation of line section faults," in *Proc. 1986 PES Transmission and Distribution Conf.*, Sept. 1986.

[24] S. Mokhtari, J. Singh, and B. F. Wollenberg, "A unit commitment expert system," in *Proc. 1987 Power Industry Computer Applications Conf.* (Montreal, Que., Canada, May 1987).

[25] J. R. Latimer and R. D. Masiello, "Design of a dispatcher training simulator system," in *Proc. 1977 IEEE Power Industry Computer Applications Conf.*, pp. 87–92.

[26] I. Susumago *et al.*, "Development of a large-scale dispatcher training simulator," *IEEE Trans. Power App. Syst.*, vol. PWRS-1, No. 2, pp. 67–75, May 1986.

[27] D. Magee *et al.*, "A large two computer dispatcher training simulator," *IEEE Trans. Power App. Syst.*, vol. PAS-104, no. 6, pp. 1433–1438, June 1985.

[28] T. E. Dyliacco *et al.*, "Considerations in developing and utilizing operator training simulators," *IEEE Trans. Power App. Syst.*, vol. PAS-102, no. 11, pp. 3672–3679, Nov. 1983.

[29] J. D. Hollan *et al.*, "STEAMER: An interactive inspectable simulation-based training system," *AI Mag.*, vol. 5, no. 2, pp. 15–27, Summer 1984.

[30] B. Woolf *et al.*, "Teaching a complex industrial process," in *Proc. AAAI Conf.* (Philadelphia, PA, Aug. 1986).

[31] R. D. Masiello *et al.*, "Review of the EPRI project on operator training simulator implementation," presented at the 1984 EEI Engineering Computer Forum, Atlanta, GA.

[32] R. S. Michalski, J. G. Carbonell, and T. M. Mitchell, Ed. *Machine Learning.* Palo Alto, CA: Tioga Publ. Co., 1983.

Bruce F. Wollenberg (Senior Member, IEEE) was born in Buffalo, NY. He received the B.S. and M.Eng. degrees from Rensselaer Polytechnic Institute, Troy, NY, and the Ph.D. degree from The University of Pennsylvania, Philadelphia.

He has held positions at Leeds and Northrop Co., North Wales, PA, and Power Technologies Inc., Schenectady, NY. He is presently with the Energy Management Systems Division of Control Data Corporation, Plymoth, MN. He has served as an adjunct faculty member in the Electric Power Engineering Department at Rensselaer and in the Electrical Engineering Department at the University of Minnesota. His research interests include power system operations, large-scale optimization of power system problems, and application of artificial intelligence techniques to energy management systems.

Toshiaki Sakaguchi (Member, IEEE) received the B.S. and M.S. degrees in 1969 and 1971, respectively, and the Ph.D. degree in electrical engineering in 1981, all from Kyoto University, Kyoto, Japan.

He joined Mitsubishi Electric Corporation in 1971, and has been working in research and development on power systems technology at the Central Research Laboratory. He is currently the Research Manager there. His current interest is in the analysis of human heuristics and their use for management and control of large-scale systems.

Dr. Sakaguchi is a member of the IEE of Japan.