Expert Systems for Monitoring and Control

Daniel L. Dvorak

Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78705

Abstract

Many large-scale industrial processes and services are centrally monitored and controlled under the supervision of trained operators. Common examples are electrical power plants, chemical refineries, air-traffic control, and telephone networks — all impressively complex systems that are challenging to understand and operate correctly. The task of the operator is one of continuous, real-time monitoring and control, with feedback. The job can be difficult when the physical system is complex (tight coupling and complex interactions). Also, there may be faults not only in the system but also in its sensors and controls. Deciding the correct control action during a crisis can be difficult; a bad decision can be disastrous.

This paper surveys existing work in the field of knowledge-based systems that assist plant/process operators in the task of monitoring and control. The goal here is to better define the information processing problems and identify key requirements for an automated operator assistant. A high-level design of an "expert system for operators" is presented. The design relies on a functional/causal model of the physical system as a source of deep knowledge (in addition to several sources of shallow knowledge). The major processes described in this design are *focusing, model-building, tracking, envisioning* and *advising.*

Keywords: expert system, operations system, real-time monitoring, causal models.

On March 28, 1979 in a power plant in Pennsylvania some water leaked into the instrument air system of the plant. Initially this caused two water pumps to 'trip' (turn off). With two valves for emergency cooling in the wrong position (and the indicator that would have shown the error obscured), a pressure-relief valve opened but then failed to reseat, but its indicator showed that it had closed. All of this happened in thirteen seconds and it set the stage for a drama that unfolded over the next two weeks into the worst nuclear power plant accident in the United States — the Three Mile Island accident.

1 Introduction

"System accidents" happen all too often — we only hear about the ones that are "newsworthy". Other news-making examples in recent history were the 1977 New York City blackout and the near-tragedy of the *Apollo 13* space flight in 1970. Analysis of such accidents often reveal that, in spite of a fault in some mechanism or a mistake by a person, an operator could have brought things under control if he/she had been able to construct a mental model of the system that fit all the facts. This is hard enough to do with a complex system, and almost impossible under the pressure of an emergency situation when there may be multiple faults, numerous alarms, conflicting data and missing or incomplete information.

The purpose of this research is to assist the operators of complex engineered systems by (1) helping them to deduce the state of the system from the (possibly faulty) sensor data, and by (2) providing expert advice on possible actions, even in the face of incomplete knowledge. The focus is on finding ways to assist the on-duty operator; less attention is given to the off-line tasks of administration, scheduled maintenance, and training.

This paper surveys a rather broad range of topics associated with the operations problem; it is certainly not focused on one specific issue. This was deliberate since the purpose of this paper was to discover and understand the *requirements* of the larger problem. This is a necessary step in trying to formulate an architecture for solving this class of problem. Later research will focus on specific issues, but it can then be done with a better understanding of how it contributes to solving the larger problem.

2 Definition of Terms

Some of the terms used throughout this report may be unfamiliar to the reader or may be loosely defined in the literature. In either case, the following definitions are suitable for the

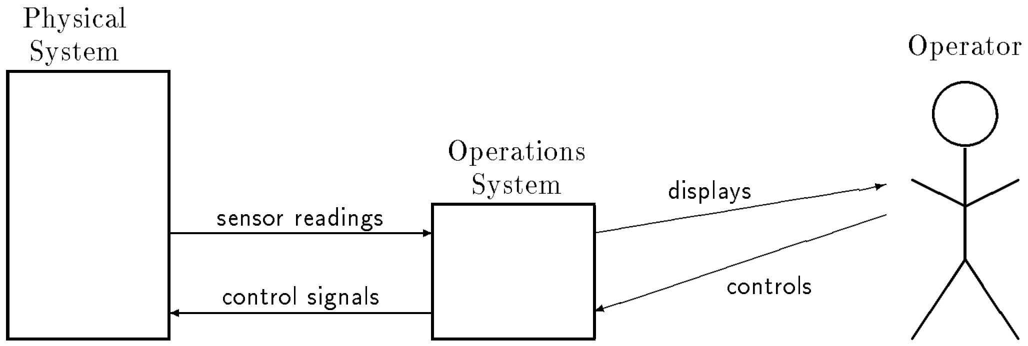

Figure 1: The role of an operations system

purposes of this report.

- Subject system, often abbreviated to just system, is used throughout this report to refer to the physical plant/process/system that is being monitored and controlled by an operator, usually via an operations system.

- Operations systems[1] are computer-based systems to assist a human operator in the job of monitoring and controlling a subject system. Typically, an operations system centralizes into one control center the display of many remotely monitored system parameters and the operation of many remote controls. It is the job of the operator to interpret the state of the subject system from the monitored parameters and take appropriate actions.

- Observables are the set of parameters of the subject system that are monitored and whose readings are available to the operations system. Some parameters may be measured automatically through sensors; others may be measured manually.

- Control systems perform *automatic* control of a physical system according to the value of a "performance index" (also called "objective function"). The performance index (*p.i.*) is often expressed in financial terms as a profit per unit time. The *p.i.* must include a measure of all effects that might result from control action. How comprehensive a control system is depends on the selection of the *p.i.* and the controlled variables [Tay66]. Control systems, then, attempt to achieve optimal operation in an error-free system.

- Model-based reasoning is reasoning about the behavior of a system using a *model* based on the structure and function of the system. By basing the model on system design and physical laws rather than empiric knowledge, the model can predict all possible behaviors[2], even those unforeseen by the designer. The model may use any combina-

[1] The term *operations system* emerged after computer-based systems for operations began to appear in the mid-1960s. As described in *Engineering and Operations in the Bell System* [Rey84] the term covers computerized systems for administration and maintenance as well. The definition used in this report emphasizes only the role in operations.

[2] More precisely, a model can predict all possible behaviors at the level of abstraction that it was built for. A model's predictive power is always limited by the simplifying assumptions that it is based upon.

tion of reasoning schemes (numerical, qualitative, first-order logic, etc.) — the main requirement in this research is that it correctly predict the possible behaviors of the subject system given a starting state. Optionally, models may also aid in providing explanations of the state and behavior of the system.

- An envisionment is a graph that shows all behaviors of a system from some initial state. Each node represents a qualitatively distinct state of the system; each arc represents a valid state transition.

- A fault model is a model in which one or more components, connections or processes of the subject system are treated as inoperable, or in which a new (abnormal) connection or process has been introduced.

- Expert systems are sophisticated computer programs that manipulate knowledge to solve problems efficiently and effectively in a narrow problem area [Wat86, page xvii]. An expert system provides high-level expertise to aid in problem solving. The expertise (knowledge) is explicit and accessible. Two capabilities of expert systems that are particularly important in this work are *predictive modeling* and *explanation*.

3 Operations — the Information Processing Problem

In large plants the duties of *operations*, *administration* and *maintenance* are usually divided among different staffs. Maintenance people are responsible for equipment diagnosis and repair. Administrators perform a variety of system management activities such as system configuration, capacity planning, maintenance scheduling, and production planning. Operators monitor and control the physical system from a control room during production. One source [SPT86] describes well the difficulties faced by operators:

> *"...computer technology allows more complex systems to be controlled but it also enables far more information about the system to be displayed. For example, process control and instrumentation systems such as those found in nuclear power plants may have around 2000 alarms in a control room in addition to the displays of analogue plant data. When plant crises occur, these data can change very rapidly. In one simulated loss-of-coolant accident, 500 lights went on or off within the first minute, and 800 in the second.*
>
> *It is in these kinds of real-time problem-solving situations that many of the limitations of humans are at their most apparent. Their tendency to overlook relevant information, to respond too slowly and to panic when the rate of information flow is too great all contribute to lower than desired levels of performance."*

Briefly, the operator must:

— monitor sensor readings/displays and recognize abnormal conditions;

— combine all information into some notion of the system's state;

— determine what parts of the system are faulty, including sensors and controls;

— understand how the normal system functions and how various faults alter that;

— respond during abnormal conditions to restore the system to a safe, efficient operating condition;

— make decisions in limited time;

— make a "best guess" in situations for which there is insufficient information to make a *certain* decision; and

— maintain plant/process safety policies and follow "recommended operational procedures."

The key cognitive skill is the formation of a mental model of the system that fits the current facts and enables the operator to correctly predict the system's behavior and predict the effect of possible control actions.

The operations job is difficult because the operator must make timely decisions using information that is incomplete and uncertain. Making the information more complete by providing new sensors is not a solution since cognitive overload is already a problem in emergencies involving complex systems. Providing more time to make a decision means slowing down the whole system, thus reducing throughput (but the economic pressures are to *speed up* production). Providing more operators to share the workload, and more expertise to handle unusual situations, can indeed help. It is in this spirit that an expert operator assistant is proposed.

4 Lessons from System Accidents

What improvements should be made to existing operations systems? One set of clues can be found in the records of notable accidents involving complex systems. In his book *Normal Accidents: Living with High-Risk Technologies* [Per84], Charles Perrow has analyzed the records of many accidents involving nuclear power, petrochemical plants, aircraft, marine vessels, space, and weapons. Although Perrow describes the technical details of several accidents, he is more concerned about the nature of the larger system and the human behavior of those operating it. The following observations from Perrow offer some insight into the problems of operating complex systems.

- The systems that have the highest potential for catastrophic failure are those in which interactions are complex and coupling is tight (see Figure 2). Nuclear plants are the most extreme exemplar in this category; some others are aircraft, space missions and DNA research.

- The accidents that have occurred at commercial nuclear power plants are partly a result of having "undisciplined, untrained and, unmotivated people." This is in contrast to the naval reactor program which is "run with an iron fist, . . . , intense training, intense discipline on the operators."

- The computerization of process/plant control has the effect that it "does not encourage broader comprehension of the system — a key requirement for intervening in unexpected

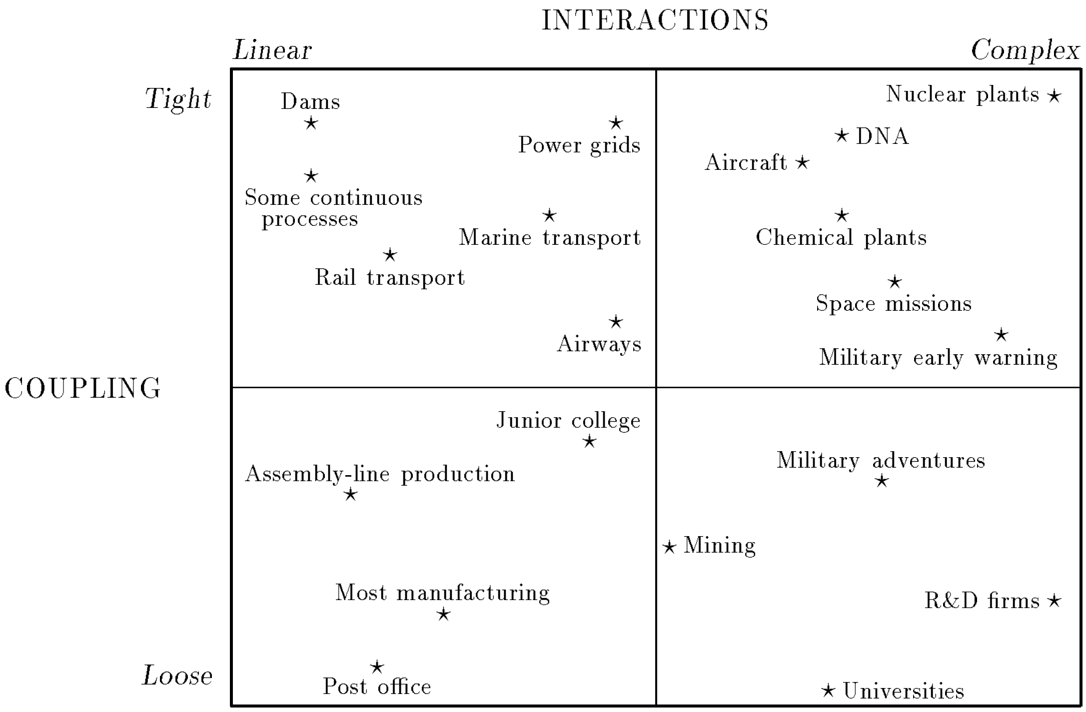

Figure 2: Interactions/Coupling Chart

interactions."

- In many accidents caused by human error, the operator "built perfectly reasonable mental models of the world, which worked almost all the time, but occasionally turn out to be almost an inversion of what really exists."

Perrow ultimately concludes that, for some systems, *"systems are too complex, and our reach has surpassed our ability to grasp."* He argues that some system designs, such as the typical PWR[3] nuclear reactors, are too dangerous and should not be built. He criticizes the "technological fix" which says "Just give the leader more information, more accurately and faster." This just worsens the problem of cognitive overload that often occurs in crises.

A different lesson, one that Perrow (a sociologist) does not discuss much, is that we need better aids for operating complex systems. These aids must help the human operator to understand the system and to form correct mental models when abnormal situations arise. The aids must do this not by offering more raw data but by analyzing and interpreting the available data and by giving expert advice.

[3]PWR = Pressurized Water Reactor (the most common type of nuclear reactor).

5 Related Work

There has been a variety of work in expert systems for real-time monitoring & control but, to my knowledge, it has never been collected and examined as a class. This section attempts to do that. This is not an exhaustive list — the number of projects in this area is continually increasing. Rather, this is an attempt to identify projects that offer some insights into defining and/or solving the problems of knowledge-based monitoring and control. The project descriptions have been grouped according to the area of what I believe to be their principle contribution in this study. Admittedly, this is a somewhat artificial division since most of the projects address a range of issues.

5.1 Problem Definition

The following articles help to define the problems faced in knowledge-based monitoring and control of engineered systems.

5.1.1 Typical Tasks

In a survey study on the application of knowledge-based systems to industrial environments, [ACG+85] lists the following as typical tasks:

- Diagnosis of malfunctions of the components of a system based on the operator's observations plus automated analysis of the signals collected from the system being monitored.
- Assessment of the operating conditions and dynamic trends of a complex process that could cause end-product quality degradation or possible hazards for the installation.
- Online assistance to the human operator of a complex plant in emergency recovery, fault detection and diagnosis, short-term intervention planning, etc.
- Planning the appropriate interventions for plant maintenance and repair, in compliance with given production and quality objectives (an *administrative* task).
- Offline assistance to maintenance technicians (a *maintenance* task).

5.1.2 CAM

CAM [RS81] (CAusal Monitor) is a project whose aim is to develop a framework for the construction of intelligent, causally-based real-time monitors for arbitrary physical sites, such as a nuclear power plant or a NASA mission control room. The authors define an ideal causal monitoring system as one that extends and broadens the operator's ability to collect and synthesize relevant information in time-critical situations. Such a system would:

- continually sense and symbolically characterize all sensors in a way that reflected their relative importance and acceptable ranges in the current operating context;

- continually verify that causally related sensor groups obey symbolic rules expressing the nature of their causal relationship;
- be aware of the human operator's expressed intent (*e.g.*, "running preventive maintenance check 32", "executing morning power-up") and adjust causal relations and expectations and lower-level parametric tolerances accordingly;
- have knowledge of component and sensor failure modes and limit violations, their detectable precursors, their indicators, their probable causes, and their corrective procedures (both in the form of automatic correction algorithms and recommendations to the human);
- have a knowledge of standard "maneuvers", expressed as action sequences with stepwise confirmable consequences (for both automatic execution and as an archival reference for the human);
- continually synthesize all important aspects of the system and from this synthesis, identify which aspects of the system to display to the human controllers (in the absence of specific requests from the humans);
- decide on the most appropriate screen allocation and display technique for each piece of information under the current context.

CAM's uses a frame-like knowledge representation and a high throughput forward/backward reasoning scheme based on a dependency lattice (essentially a compiled production rule system).

5.1.3 VM

VM [FKFO79] (Ventilator Manager) was an early (1978) example of an expert monitoring and control system; its purpose was to provide diagnostic and therapeutic advice in a hospital's Intensive Care Unit. VM's inputs are 30 physiological measurements provided periodically by an automatic monitoring system. VM's functions and outputs are to:

- provide a summary of a patient's physiological status appropriate for the clinician,
- recognize untoward events in the patient/machine system and provides suggestions for corrective action,
- suggest adjustments to ventilator therapy,
- detect possible measurement errors, and
- maintain a set of patient-specific expectations.

VM is a data-driven (forward-chaining) rule-based system that cycles through the rule set each time new data is available. VM reached the stage of a field prototype.

5.2 Expert Operations Systems

The following are real-time knowledge-based operations systems and, as such, provide early examples of the external characteristics of an expert operations system (EOS) and some approaches to the program design.

5.2.1 YES/MVS

YES/MVS [GHK+84] is a continuous, real-time expert system developed at IBM Research for monitoring and controlling an MVS operating system. YES/MVS exerts interactive control over the MVS system as an aid to computer operators. The project was motivated by a chronic shortage of skilled operators and the increasing complexity of the operator's job. Most of the knowledge in YES/MVS is based on "rules of thumb" used by operators and system programmers.

The technical focus is on the challenge of operating an expert system in a dynamic real-time environment and the problems of maintaining a consistent model of the subject MVS system. To handle real-time on-line data, a data-driven inference engine (OPS5) was chosen. In order to meet the requirements for continuous real time operation, the following extensions were made to OPS5:

- An OPS-WAIT function was added to suspend rather than terminate the inference engine when no rule is eligible to fire.
- The right hand side of rules was compiled to improve speed.
- A TIMED-MAKE was added to be able to initiate an action at a given time (a fundamental requirement in real-time control).
- A REMOTE-MAKE was added as a natural communication primitive between separate production systems (YES/MVS uses two).
- Some critical problems such as handling hardware error messages have a real time requirement that necessitates explicit control over the rule firing in the inference engine. To do this, a Priority Mode was added to the conflict resolution strategies. Each rule is tagged with a task-id, and each task-id has an associated priority.

These OPS5 extensions are generally applicable to other real-time interactive problems such as process control.

5.2.2 ESCORT

ESCORT [SPT86] (Expert System for Complex Operations in Real-Time) provides an environment for building real-time expert systems to help relieve 'cognitive overload' of operators overwhelmed by the volume of information presented during an emergency. ESCORT performs four knowledge-based tasks:

— filters the plant data looking for occurrences or events which may indicate plant problems or operator error;

— prioritizes these "primary events" based on the relative importance of different areas of the plant and the alarms within them, and also on whether a problem has been previously inferred for each event and, if so, how long ago;

— analyzes each event to infer the underlying plant or instrument problem or operator error;

— prioritizes the the underlying problems by their relative importance in maintaining plant production and avoiding shut-downs;

To handle large amounts of data and operate in real time, ESCORT uses a knowledge-based scheduler that is aware of the plant state, the diagnoses made and any operator requests. When a primary event occurs, it is investigated and the priority ladder is updated (as described above). When there are no primary events left to analyze, ESCORT continuously re-assesses its earlier diagnoses, checking to determine whether the problem symptoms still persist.

5.2.3 REACTOR

REACTOR [Nel82] is an expert system developed to assist operators in the diagnosis and treatment of nuclear reactor accidents. The purpose of REACTOR is to monitor a nuclear reactor facility, detect deviations from normal operating conditions, determine the significance of the situation, and recommend an appropriate response. The system combines situation knowledge (represented in IF-THEN rules) with functional knowledge (represented in a response tree). The situation knowledge has been accumulated through experience with actual accidents as well experiments in test reactors and simulation results. The functional knowledge is based on the structure of the reactor's safety systems.

Initially, the system uses the situation rules and reasons forward from known facts until a conclusion can be reached. If not enough information is available to reach a conclusion, REACTOR reasons backward to gather additional information. If still no diagnosis can be reached, REACTOR seeks to put the system into a safe state by implementing the safety functions. Here, it switches to the functional knowledge in the response trees. A response tree is a diagram which shows the success paths which can be used to provide a given safety function. Some paths will be unavailable due to equipment faults. REACTOR determines an available path and displays it graphically to the operator on a diagram of the reactor system. REACTOR was developed by EG&G Idaho Inc. and reached the stage of a research prototype.

5.3 Process Control Systems

The reasoning and representation techniques used in knowledge-based process control systems bear close examination since they share many of the problems of operations systems.

5.3.1 PICON

PICON (Process Intelligent CONtrol) [MHKC84], a commercial product of Lisp Machine, Inc., is a real-time expert system shell for process control. To cope with the problem of a major process upset in which hundreds of parameters may change significantly in a few minutes, PICON is designed to perform inference as would a human expert (since exhaustive search is not possible in real time). The key concepts are *"to quickly recognize process conditions which are potentially significant, and to invoke relevant rule-sets and focus on these problem areas for diagnosis and procedural advice."*

Via a dedicated 68010 processor for real-time data access (and certain low-level inference tasks), PICON is interfaced with a conventional distributed control system with access to as many as 20,000 measurement points. The inference engine running in the Lisp Machine requests data from the 68010 as needed for inference. One way that PICON can focus attention is by changing the time period of measurement in the 68010 and by downloading special processing algorithms into the 68010. The other mechanism for focus is through the priority associated with each rule set.

Process dynamic conditions can trigger forward chaining procedure heuristics to focus PICON's attention for diagnosis. Diagnosis is then performed by backward chaining, with explanation based on the inference path.

For typical disturbances, corrective action is based on documented normal operator procedures. For serious disturbances understood by only the most expert operators, heuristic models of process behavior were used since precise models were not available. These heuristics are of the form

(Action; Time Delay; Expected Result)

Further action is initiated by procedure rule depending on the comparison of the expected result with the actual outcome.

5.3.2 LMA

LMA, the Logic Machine Architecture [LS83], is a system developed at Argonne National Laboratory for their Man-Machine Control System (aimed principally at control of nuclear power plants). The project seeks to reallocate some control functions currently performed by human operators to an automated reasoning system. LMA is a reasoning system implemented in about 50,000 lines of Pascal code (for portability).

LMA is an outgrowth of long-running research in automated reasoning. It is an inference system that relies most heavily on *hyperresolution* and is quite similar to PROLOG-style logic programming (albeit, with some important differences). Model structure and rules are both expressed in LMA's clause formulation. The most demanding aspect of the project according to the authors is *"to formulate the clauses which correspond to rules for deducing the state of elements from their sub-elements, for deducing the correct sequence of operations necessary to achieve a desired state, and choosing new goals when original ones cannot be reached."* A main theme in LMA seems to be that *all* knowledge about the plant and its operation is expressed in clause-form, from which the theorem prover makes all inferences. The authors claim that the representation is easy to use.

5.3.3 FALCON

FALCON [CLD84, Wat86] identifies probable causes of disturbances in a chemical process plant by interpreting data consisting of numerical values from gauges and the status of alarms and switches. The system interprets the data by using knowledge of the effects induced by a fault in a given component and how disturbances in the input of a component will lead to disturbances in the output. Knowledge is represented in two forms — as a set of rules controlled by forward chaining and as a causal model in network form. FALCON reached the stage of a demonstration prototype.

5.4 Examples in Representation and Reasoning

The following systems provide some additional examples of representation and reasoning techniques that may be of value in the design of an EOS.

5.4.1 RiTSE

In nuclear reactor operations, a principle cause of unplanned outages and poor emergency response is human error. Operations and maintenance personnel are not trained to analyze the many subtle and complex interactions of the simultaneous activities of maintenance, testing and surveillance at the plant. RiTSE, the Reactor Trip Simulation Environment [Eps86b, Eps86a], is a frame- and rule-based AI application that aids nuclear power plant operators in determining if a proposed change in the state of a component or process will cause safety systems to automatically shut the plant down.

RiTSE provides a generalized framework within which to model nuclear power generating facilities. Plant entities and rules are both represented in frames. Plant entities include mechanical components, electrical components, controllers, processes, sensors and systems. Rule frames contain a slot for the rule plus other slots for the English version of the rule, pointers to referants, names of heuristics, actions to take, values to return, and conflict resolution strategies.

Given a knowledge base of some 4,000 systems/components and 1,200 rules on an IBM/PC-AT, RiTSE can determine in 10–20 seconds if an initiating event will result in a reactor trip. It does this by propagating the event through a model of the plant. RiTSE's speed is due to data-driven inference in which objects have links to the rules that depend on them and rules have links to the objects they affect (this appears to be similar to the dependency lattice of CAM). The Boolean nature of the knowledge base also contributes to the speed of inference (one user referred to RiTSE as a "Boolean spreadsheet"). RiTSE has been used at the Salem Nuclear Plant and widely reported in the nuclear power industry..

5.4.2 CSA and NPPC

CSA [RG76] (Common-Sense Algorithm) provides a method for describing and simulating the cause-effect behavior of mechanisms. A CSA representation is a graph composed of four types of "event" nodes (action, tendency, state, statechange) connected together by seven[4] relationships (links): cause (one-shot, continuous, or repetitive), enablement, statecouple, equivalence, antagonism, threshold, and rate confluence. Given a declarative (CSA graph) representation of a mechanism, it is converted to a procedural representation wherein each event (node) is converted to a "spontaneous computation" (SC) unit that will execute when its preconditions (links) are true. A "trigger tree" is used for efficient and exhaustive retrieval of all SC's which have the potential for reacting to a given stimulus.

CSA was later applied in the Nuclear Power Plant Consultant (NPPC) [Und82, Nel84], an expert system to help operators determine the cause of some abnormal condition such as "high containment temperature". The system uses a 350-event CSA model of the primary coolant system, (including pumps, reactor, steam generator, and emergency core cooling systems), a CSA network simulator, and a rule-based forward-chaining diagnostic program. The expert system was said to most closely resemble CASNET or EXPERT, differing primarily in the use of CSA nets, the integration of diagnostic rules into the CSA nets, and in its simulation capability.

5.4.3 PROP

PROP [GGSS85] is an expert system for on-line monitoring of the cycle water pollution in a thermal power plant. The expert system architecture is novel in its representation that combines both declarative and procedural knowledge, processing it with a unified inference engine. Specifically, procedural knowledge of data interpretation and diagnosis is represented in a collection of event graphs (similar in form to a Petri net). Control knowledge, which is more declarative and fragmentary in nature, is represented in production rules whose conditions and actions refer to markings in the event-graphs. The event-graph approach used here is very similar to the spontaneous computation units described in CSA above. This heterogeneous knowledge representation scheme offers three benefits:

- knowledge acquisition is easier and knowledge organization more natural;
- non-deterministic reasoning (*i.e.*, rule-base reasoning) is limited to those aspects of the problem domain that actually imply non-deterministic operation;
- explanations can more closely follow the path of human reasoning as a consequence of both the naturalness of knowledge representation and the constrained non-determinism of the inference process.

[4] CSA was part of a larger project to study two aspects of human intelligence — problem solving and language comprehension. The full CSA description language contains 16 relationships (links), but some of them deal with human concepts like intentionality. Only 7 of the relationships apply to mechanisms.

PROP has been implemented in Franz Lisp on a SUN-2 workstation at the CISE laboratories (Italy) and tested in a wide variety of operating situations. Further research is in progress to extend the PROP architecture.

5.4.4 REALM

REALM [TC86] is an expert advisory system to assist the operator in determining what level of emergency exists in a nuclear power plant. REALM interprets sensor input in order to identify and classify emergencies. REALM also interacts with the operator and requests that certain information be entered manually. REALM has the ability to "look ahead" in an emergency situation to predict what else might go wrong. Such a vulnerability analysis evaluates the current knowledge in context and reviews all of the rules relevant to to the next higher level of emergency.

The project (which is ongoing) is presented as a straightforward computerization of a manual task using knowledge-based techniques. The operation of REALM involves a dialogue among several "mini-experts": a Radiation Expert, a Critical Safety Functions Expert, a Fission Product Barrier Expert, an Events Expert (which looks for known accident sequences), and an Emergency Action Level Expert (which uses the output of the other experts to classify the emergency level).

REALM is being implemented on a Xerox Lisp machine within KEE using frames, rules and object-oriented programming (plus some Lisp procedures and utilities).

5.4.5 Qualitative Reasoning about Physical Systems

Qualitative modeling of a mechanism based on its structure provides a source of deep causal knowledge that identifies the qualitatively distinct behaviors of a system. This type of reasoning removes the often irrelevant numeric detail, leaving the qualitatively distinct behaviors that are of primary interest to operators of physical systems. There are several approaches to qualitative reasoning; among these are the device-centered approach of deKleer and Brown [dKB85], the process-centered perspective of Forbus [For84], the qualitative mathematics approach of Kuipers [Kui86], the predictive analysis approach of Pan [Pan84], and the structural/functional approach of Davis [Dav85]. All of these researchers are concerned with modeling physical systems, but Pan and Davis in particular focus on the problem of diagnosis, albeit with different approaches. Pan requires a state-transition model that encodes failure modes as states and failure causes as transitions (possibly requiring the failure cause to persist for a given amount of time for the transition to occur). Davis begins with a structural/functional description of the mechanism and then progressively suspends constraints to find a model consistent with the observed values. In both cases the techniques adopted have been strongly influenced by the selected problem domain.

5.5 Handling Sensor Data

Process control systems and operations systems both need to monitor the physical system under their control. This section illustrates three separate problems in handling sensor data — translating numeric readings into qualitative terms, dealing with errors in sensor readings, and updating time-dependent data.

5.5.1 Qualitative Interpretation

Forbus has presented a theory of how to achieve a qualitative understanding of a physical system based on raw measurement data, taken across time [For86]. The theory describes how to translate observed behavior, which may include numerical data, into qualitative terms. The theory is domain independent and makes two assumptions about the nature of the underlying domain model:

1. Given a particular physical situation, a graph of all possible behaviors – an *envisionment* – may be generated.
2. Domain-specific criteria are available for quantizing numerical data into an initial qualitative description.

5.5.2 PDS

PDS [FLK83] is a forward chaining rule-based system for process diagnosis. It is being implemented in environments in which data acquisition is totally automated, thereby limiting the amount of user interaction. In these environments, problems of spurious readings and sensor degradation have to be solved. PDS applies three methods. Through "retrospective analysis" PDS can store and analyze successive readings of a sensor and apply various forms of filtering and smoothing to deal with spurious readings. Through "meta-diagnosis" PDS detects sensor degradation and then adapts the rules to reflect the reduction of importance of a malfunctioning sensor in the diagnostic process. Through a "composite sensor" schema, PDS applies a transform function (such as voting or ignoring the lowest or highest reading) to combine multiple sensors into a single composite sensor.

5.5.3 Instrumentation Diagnosis

Chess et al. [COGB85] describe several techniques for detecting and diagnosing instrumentation (sensors) through artificial intelligence. These include:

- Using predictable relationships between different sensors at the same location. For example, the sensors for specific conductivity and pH are *both* sensitive to the concentration of ammonia.
- Using like sensors at different locations. For example, agreement among chemical sensors can be checked by routing the same stream past each sensor.

- Active sensor test and calibration. Some sensors can be tested and calibrated by introducing a signal of known strength. The sensor response, if it is working, can be used to calibrate the readings.
- Some types of sensors degrade in a characteristic manner. If this pattern is detected, it can be used to adjust or discount the degrading sensor.
- Operator-assisted diagnosis. If a manually-read sensor can be used to check an on-line sensor, the AI system can request the operator to read the manual sensor and enter the results. Naturally, this should only be done when the on-line sensor reading is suspicious and there is no other method to check it.

5.5.4 CURRENT

CURRENT [SBH85] is an inference engine for selective updating of time-dependent data. Developed for military command and control environments, CURRENT reasons about the reliability and timeliness of data of differing update rates, and makes decisions about obtaining more up-to-date information. Automatically updating an object's value when it's current value expires can consume a lot of computing time. Instead, CURRENT updates objects selectively. CURRENT only extends a backward chain if the object being accessed is older than the time specified for its validity. This results in greater efficiency and real-time responsiveness. It also assures the timeliness of the accessed data. If the timeliness of a particular value is crucial to a computation, the validity duration of that object can be set short enough to assure its recency for each access. Using a hybrid of AI and numerical techniques, CURRENT provides a carefully pruned update schema.

5.6 Mental Models and Human Interfaces

5.6.1 STEAMER

STEAMER [HHW84] is a research effort concerned with exploring the use of AI software and hardware technologies in the implementation of intelligent computer-based training systems. The key idea is that of an *interactive inspectable simulation.* For example, Steamer's dynamic graphical simulation of a steam propulsion plant allows a student to see processes and conditions that are not visible in the physical system. This is felt to be an important contribution in the student's formation of a mental model.

Steamer's main contributions are in the areas of training and human interface. Although these are not the focus of the EOS research, Steamer is important in that it highlights the importance of visualizing the system, even the parts that aren't visible. To the extent that an EOS can describe what is happening inside a system, telling more than just the values of the observables, it will help the operator.

6 Requirements Summary

This section distills key items from the previous sections. The goal here is to collect the various requirements of an expert operations system and state them without implying any particular design or implementation strategy. The requirements are in *slant* font followed by comments.

Size *Be able to handle systems containing a few thousand components, connections, and processes.*

This approximate measure of size comes from nuclear power plants and serves as a test to eliminate strategies that do not scale up. This primarily influences the choice of knowledge representation and reasoning schemes. It certainly suggests that strategies involving the storage of all possible combinations of faults are unworkable[5]

Causal Knowledge *Be able to handle unforeseen situations.*

First-generation expert systems were built upon empiric knowledge that associated symptoms with findings. Such systems are inherently limited to handling the situations that the domain expert and knowledge engineer thought about. For systems having complex interactions (and thus difficult-to-predict symptoms), it becomes necessary to rely upon a causal model of the system, *i.e.*,, a model based on the structure and function of the physical system.

Prediction *Given a current belief of the state of the system, be able to (1) predict subsequent behavior of the system and (2) predict the effects of proposed control actions.*

This is a specific example of the kind of deep knowledge needed. It suggests that reasoning be based on a causal model of the physical system.

Multiple Types of Knowledge *An operator's decisions are based on several kinds or knowledge — causal knowledge of the plant or process, likelihood of various types of faults, characteristic sensor degradation, desirable and undesirable operating conditions, plant safety regulations, recommended operating procedures, and current operational goals.*

This requirement makes clear that, whatever design is adopted, it must accommodate different types of knowledge and possibly different but cooperating reasoning mechanisms. Causal knowledge may be represented as a predictive model, safety regulations as rules, desirable and undesirable operating conditions as Boolean expressions and algebraic formulas, and recommended operating procedures as scripts or procedures.

[5]Consider a system of only 1000 components (smaller than a nuclear power plant). It has 2^{1000} possible combinations of faults. This is far beyond the capacity of any available memory technology to store. Even just saving all triple faults would require storing 166 million combinations. However, large systems are often kept in operation with known (non-critical) faults. The operator makes compensating adjustments and defers maintenance until a more convenient time. Thus, multiple faults are part of the "normal situation" that operators (and therefore EOSs) must deal with.

Real-time *In reaction to changes in the observables, the EOS must produce advice in time to achieve the desired operating conditions, or at least prevent a worsening of conditions if the former is not possible.*

Anything more specific about the real-time requirement will depend on the physical system being controlled. In terms of EOS design, some existing techniques include: a knowledge-based priority scheduler (ESCORT), a priority-mode conflict resolution strategy for prioritized rules (YES/MVS), time delays on actions or expectations (YES/MVS and PICON), and dedicated hardware for real-time data access (PICON).

Diagnosis *Be able to detect and diagnose system faults, including those in sensors and controls.*

Fault diagnosis by an operator is different than fault diagnosis by maintenance staff in three ways:

- Diagnosis must be performed on-line, while the system continues to run (unless there is an obvious critical fault for which the system must be shut down). The reason is that the operator's prime directive is to keep the plant running if possible.
- The operator's objective is to determine how a fault may affect the safe or optimal operation of the system (but not necessarily to pinpoint the actual cause of the fault).
- A repairman can usually trust his instruments. The operator's "instruments" are the remote sensors and controls, and they are often just as susceptible to failure as any other component.

Advice *Supply advice to the operator or, alternatively, perform automatic control.*

Advice to the operator is the end-purpose of the EOS. To be successful, the advice must be timely and must be of expert quality. Previous expert operations systems have typically generated advice from situation–action rules.

Human Interface *Show the operator, either continuously or on demand, the EOS's current belief of the physical system's state. Likewise, supply explanations for any recommendations.*

This is an important requirement in the design of a production-quality EOS, but it is not a focus of this research.

7 Program Design

Figure 3 depicts a very high-level design for an EOS. The proposed framework identifies major knowledge sources and reasoning activities working together in performing the EOS task. A central goal of the design is to find the one model (or a very small set of models) that accounts for the observed behaviors of the physical system. The hypothesis here is that in order to give expert advice about the current situation, the EOS must first "embody" that situation in a predictive model. The major components of the design (shown in Figure 3) are described below.

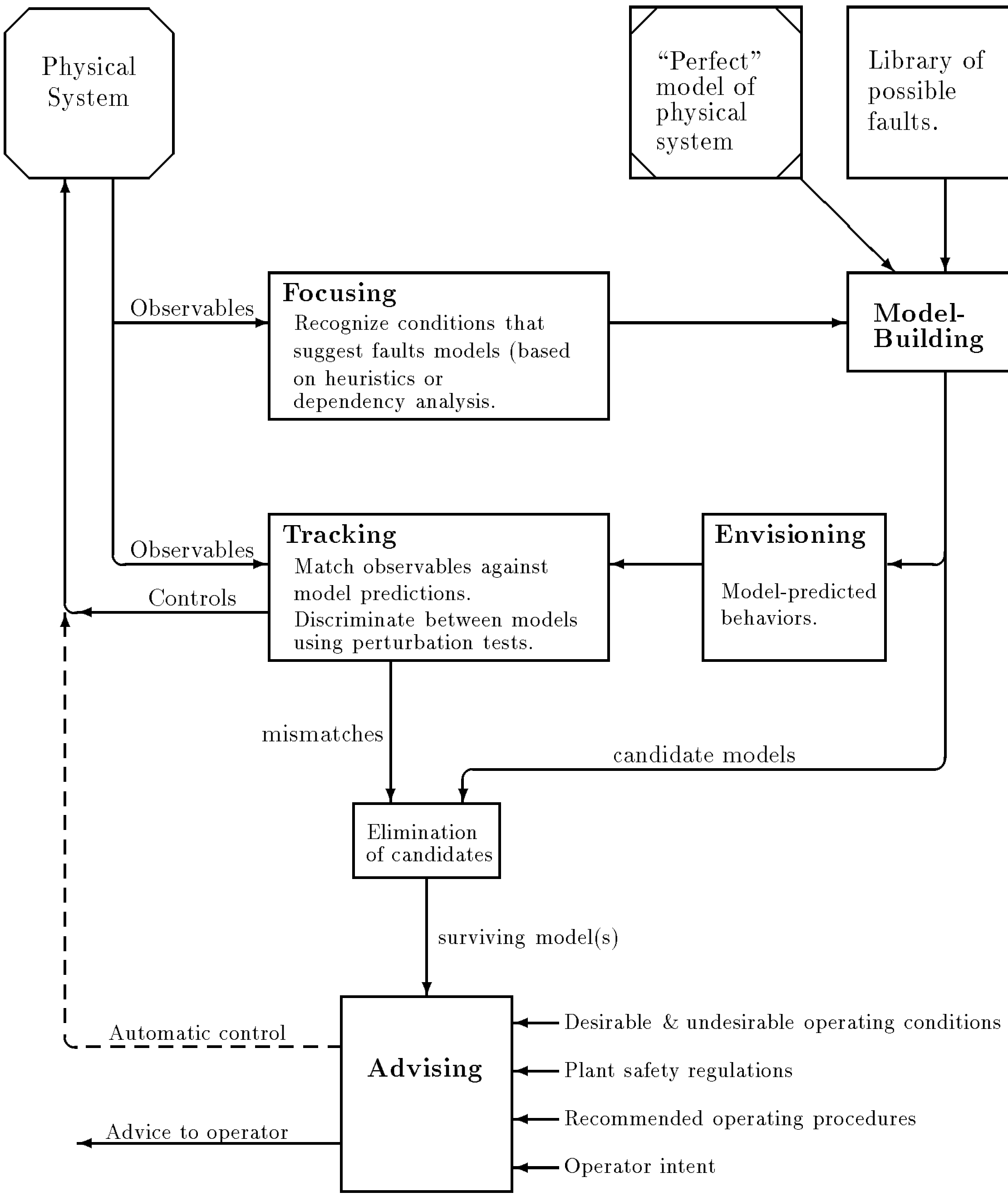

Figure 3: Top-level design of an expert operations system.

Model Library Two elements are needed to construct a model for any situation: (1) the "perfect" model of the physical system describing its structure and function (without faults) and (2) the characteristic faults of the system's components and connections. These can then be combined to model a system with any combination of possible faults.

Focusing An experienced operator recognizes the symptoms of a wide variety of faults. This knowledge may be learned from experience or realized from a cause-and-effect understanding of the system. In either case, it helps the operator focus very quickly on a few possible faults out of a very large number. The *focusing* module does the same thing. Some of its knowledge may be stored as associations between symptoms and possible causes. Other knowledge may be derived from a dependency analysis of a model of the system. The output of "focusing" is a list of possible faults, possibly ordered by probability.

Model-Building Given a perfect[6] model of the system and a set of possible faults, the task here is to construct fault models. This is a straightforward task of injecting the fault into the model.

Envisioning The candidate models are important because of their predictive power. "Envisioning" is the task of running the model to see what behaviors it produces from a given input and state.

Tracking Given a hypothesis about the physical system as represented in a candidate model, the tracking task is to determine if the behaviors predicted by the model match the observables of the physical system. Models that don't match are eliminated. In the case where two or more models continuously match the current sequence of observables, the EOS may discriminate between them by running a perturbation experiment that, regardless of the outcome, must eliminate one of the models.

Advising The end-purpose of an EOS is to advise the operator and, in situations that the operator has authorized, perform automatic control on the physical system. Correct advice often depends on having the correct model (as opposed to the perfect model), especially in an emergency situation. But, just having the correct model doesn't tell what ought to be done. The advice must also be based on goals to achieve, conditions to avoid, and rules about how to do certain things. The "advising" task is a large knowledge-intensive task; the assumption made in this research is that much of this task can be solved with existing expert systems techniques.

All of the reasoning tasks described above are continuous. The "interesting" reasoning occurs when an unpredicted behavior occurs or when a predicted behavior does not occur.

[6] No model is ever "perfect" since it is always an abstraction of reality. In this paper the term "perfect model" means a model of the physical system with no faults (at whatever level of abstraction deemed appropriate for the needs of the operator). It's interesting to note that the perfect model will rarely be one of the current candidate models. The fact is that large physical systems usually contain *at least* one fault; as long as these faults are not critical, the operator will be expected to keep the plant running.

8 Summary

An expert operator maintains a "mental model" of the physical system and bases his/her operational decisions on that model. However, in numerous cases of accidents involving operator error, operators either had failed to consider alternate models that also fit the observables, or had an incomplete understanding of the cause and effect in the complex physical system. This research seeks to overcome both problems.

The approach described here advocates the use of a functional/causal model of the physical system as a source of deep knowledge to determine the true model and state of the physical system at each moment in time. *Focusing* aids in selecting candidate models from an enormous space of possibilities. *Envisioning* uses the predictive power of the model to create expectations which are checked against observables by the *Tracking* mechanism. *Advising* combines the surviving model(s) with a variety of domain-dependent constraints and heuristics to provide advice to the operator about what to do in the current situation.

9 Acknowledgements

The advice and guidance of Prof. Benjamin Kuipers has been very important in shaping this research. The support of AT&T Bell Laboratories is greatly appreciated.

References

[ACG+85] F. Arlabosse, S. Celi, M. Gallanti, G. Guida, E. Horowitz, and A. Stefanini. Literature survey on qualitative modeling of physical systems, temporal reasoning, and distributed kbs architectures. Topical Report SPS/RT/85/001, CISE, P. O. box 12081, 20134 Milano, Italy, June 1985.

[CLD84] D. Chester, D. Lamb, and P. Dhurjati. Rule-based computer alarm analysis in chemical process plants. In *Proceedings of the Seventh Annual Conference on Computer Technology*, pages 22–29. IEEE, March 1984.

[COGB85] J. D. Chess, R. L. Osborne, A. J. Gonzalez, and J. C. Bellows. On-line diagnosis of instrumentation through artificial intelligence. In *Proceedings of the Instrument Society of America Power Symposium*, May 1985. Westinghouse R&D Center.

[Dav85] Randall Davis. Diagnostic reasoning based on structure and behavior. In Daniel G. Bobrow, editor, *Qualitative Reasoning about Physical Systems*, pages 347–410. MIT Press, 1985.

[dKB85] Johan de Kleer and John Seely Brown. A qualitative physics based on confluences. In Daniel G. Bobrow, editor, *Qualitative Reasoning about Physical Systems*, pages 7–84. MIT Press, 1985.

[Eps86a] Steven A. Epstein. Reasoning and representation in ritse. In *Proceedings of ?*, 1986.

[Eps86b] Steven A. Epstein. Ritse: The reactor trip simulation environment. *ACM SIGART Newsletter*, 97:23–24, July 1986.

[FKFO79] Lawrence M. Fagan, John C. Kunz, Edward A. Feigenbaum, and John J. Osborn. Representation of dynamic clinical knowledge: Measurement interpretation in the intensive care unit. In *Proceedings, IJCAI-79*, pages 260–262, Los Altos, August 1979. Morgan Kaufmann Publishers, Inc. [VM].

[FLK83] Mark S. Fox, Simon Lowenfeld, and Pamela Kleinosky. Techniques for sensor-based diagnosis. In *IJCAI-83*, August 1983.

[For84] Kenneth D. Forbus. *Qualitative Process Theory*. PhD thesis, Massachusetts Institute of Technology, 1984.

[For86] Kenneth D. Forbus. Interpreting measurements of physical systems. In *AAAI-86*, pages 113–117, Los Altos, August 1986. American Association for Artificial Intelligence, Morgan Kaufmann Publishers, Inc.

[GGSS85] Massimo Gallanti, Giovanni Guida, Luca Spampinato, and Alberto Stefanini. Representing procedural knowledge in expert systems: An application to process control. In *Proceedings, IJCAI-85*, pages 345–352, Los Altos, August 1985. Morgan Kaufmann Publishers, Inc. [PROP].

[GHK+84] J. H. Griesmer, S. J. Hong, M. Karnaugh, J. K. Kastner, M. I. Schor, R. L. Ennis, D. A. Klein, K. R. Milliken, and H. M. VanWoerkom. Yes/mvs: A continuous real time expert system. In *AAAI-84*, pages 130–136, Los Altos, CA, August 1984. American Association for Artificial Intelligence, Morgan Kaufmann Publishers, Inc.

[HHW84] James D. Hollan, Edwin L. Hutchins, and Louis Weitzman. Steamer: An interactive inspectable simulation-based training system. *AI Magazine*, pages 15–27, Summer 1984.

[Kui86] Benjamin Kuipers. Qualitative simulation. *Artificial Intelligence*, 29(3):289–338, 1986.

[LS83] Ewing L. Lusk and Rex Stratton. Automated reasoning in man-machine control systems. In *Proceedings of the Ninth Annual Advanced Control Conference*, pages 41–47. Technical Publishing Company, September 1983. [LMA].

[MHKC84] Robert L. Moore, Lowell B. Hawkinson, Carl G. Knickerbocker, and Linda M. Churchman. A real-time expert system for process control. In *Proceedings of The First Conference on Artificial Intelligence Applications*, pages 569–576, Washington, D.C., 1984. IEEE Computer Society Press. [PICON].

[Nel82] William R. Nelson. Reactor: An expert system for diagnosis and treatment of nuclear reactor accidents. In *Proceedings AAAI-82*, August 1982.

[Nel84] William R. Nelson. Response trees and expert systems for nuclear reactor operations. Technical Report NUREG/CR-3631, EG&G Idaho, Inc., Idaho Falls, Idaho 83415, February 1984.

[Pan84] Jeff Yung-Choa Pan. Qualitative reasoning with deep-level mechanism models for diagnoses of mechanism failures. In *Proceedings of The First Conference on Artificial Intelligence Applications*, December 1984.

[Per84] Charles Perrow. *Normal Accidents.* Basic Books, Inc., New York, 1984.

[Rey84] R. F. Rey, editor. *Engineering and Operations in the Bell System.* AT&T Bell Laboratories, Murray Hill, NJ, second edition, 1984.

[RG76] Chuck Rieger and Milt Grinberg. The causal representation and simulation of physical mechanisms. Technical Report TR-495, Department of Computer Science, University of Maryland, College Park, Maryland 20742, November 1976.

[RS81] Chuck Rieger and Craig Stanfill. Real-time causal monitors for complex physical sites. In *Proceedings, AAAI-80*, pages 215–217, Los Altos, August 1981. Morgan Kaufmann Publishers, Inc. [CAM].

[SBH85] Helen Stein, Richard Brandau, and Merryll Herman. Selective updating of time-dependent data. In *Proceedings of Artificial Intelligence Applications to the Battlefield*, May 1985. [CURRENT].

[SPT86] Paul A. Sachs, Andy M. Paterson, and Michael H. M. Turner. Escort – an expert system for complex operations in real time. *Expert Systems*, 3(1):22–29, January 1986.

[Tay66] David Taylor. An introduction to integrated process control. In *Integrated Process Control Applications in Industry*, pages 5–13, London, September 1966. IEE Control and Automation Division, The Institution of Electrical Engineers. Engin Lib TS 155 I 577.

[TC86] Robert A. Touchton and Mike Casella. Reactor emergency action level monitor: a real time expert system. In *Instrument Society of America Convention*, October 1986. [REALM].

[Und82] W. E. Underwood. A csa model-based nuclear power plant consultant. In *Proceedings AAAI-82*, pages 302–305. American Association for Artificial Intelligence, August 1982.

[Wat86] Donald A. Waterman. *A Guide to Expert Systems*. Addison-Wesley Publishing Company, 1986.