A Real-Time Alarm Analysis Advisor

Steven Silverman, James Dixon, Tim Fink, Paul Kotas, Alvin Shoop, Bhashyam Ramesh, Philip Klahr, and Antoine Abche

The system operation computer control system (SOCCS) monitors and assists in the control of electric power transmission and distribution in the New York City area. SOCCS gathers data from monitored points, compares them against expected normal values, and generates alarms when values are abnormal. The SOCCS alarm advisor (SAA) is an expert system that assists operators by identifying and suppressing repeating or toggling alarms, analyzing the electric network's status, and recommending appropriate restoration actions. SAA is a real-time expert system processing a maximum of 200 alarms every two seconds. In the analysis, SAA uses physical, functional, and temporal models to locate problem areas and propagate their effects through the causal network. SAA is operational and provides operators with timely reports on system conditions and operations.

Background

The Consolidated Edison Company of New York, Inc. (Con Edison) operates an electric utility system serving a 593-square-mile area of New York City and Westchester County with a peak hourly load in excess of

10,000 megawatts. The Con Edison bulk power system consists of feeders ranging from 69 kilovolts to as much as 500 kilovolts and generating units. The transmission system consists of underground cables and overhead lines. The installed generating capacity consists of oil- and gas-fired steam units, aircraft and industrial gas turbines, and one nuclear unit.

SOCCS supports Con Edison's operators in the overall control and operation of electric generation and transmission. The SOCCS software monitors remote and local telemetry; maintains a database of current state; generates system alarms; and provides automatic generation control, economic dispatch functions (decide the best source of power), security monitoring, load management, logging, and many other functions. Prior to discussing the expert system that analyzes the alarm conditions produced by SOCCS, we briefly overview the SOCCS system.

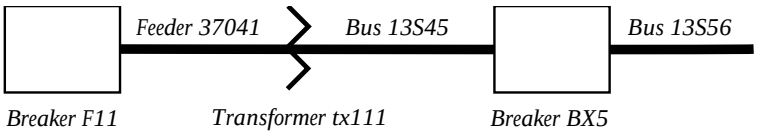
The SOCCS Host

SOCCS hardware consists of redundant supervisory control and data-acquisition (SCADA) and security assessment (SAC) Gould/SEL 3287 computers. This four-computer configuration, as well as redundant disk drives, tape drives, and so on, allows for continued SOCCS operation in the event of any central processing unit failure. Redundant configuration manager microprocessors and automatic *failover* (a turning over of computer operation to the backup system if the primary system fails) software ensure that a minimal amount of time is taken to reconfigure the hardware after a system failure.

The SCADA system scans sensor data at 69 remote terminal units (RTUs) every two seconds, updates 28 color cathode ray tubes (CRTs) on a two-second or demand basis, runs application programs, and processes and logs all alarms. The SAC system executes security assessment (analyzing contingencies), utilizing state estimation to ensure the accuracy of the data being used.

The SOCCS Database

The SOCCS database consists of approximately 100 different *point types*, representing some 12,000 analogs (watts, vars, amps, and volts); 10,000 discretes (breakers, disconnects, links, and circuit switches); 1,200 feeders; 4,200 topology points (representing the electric system configuration, for example, as shown in figure 1); and definitions of generator, load, transformer, and various other types of data points. *Analogs* are floating-point values, and discretes typically have alarm (usually a trip open condition) or normal (breaker closed condition) states. Some discretes also have an indeterminate state (when the true



Internal Number	Point	Entity 1	Type	Terminal	Entity 2	Type	Terminal	Breaker
2434	TPL344	37041	F	1	tx111	T	1	
2435	TPL345	tx111	T	2	13S45	B	1	
243	TPL10	13S45	B	2	13S56	B	1	BX5

Figure 1. SOCCS Database Relations.

discrete state is unknown). Discretes can be telemetered (through on-line sensors) or manual (set by operators). Manual data are not always reliable because new values might not have been updated in a timely manner.

Most points in the database are capable of being alarmed when an abnormal state of a point occurs. The logging subsystem records all alarms as well as events, such as operator-initiated actions, data entry, limit changes, alarm clearings, and software messages.

SOCCS Alarm Processing

Alarms on SOCCS are placed into one of 35 categories depending on the alarm type and the intended user. The alarming subsystem is intended to directly bring the appropriate alarm to a specific operator for immediate action. Alarmed categories are displayed on the bottom two lines of each appropriate color CRT, allowing only specific categories to be displayed at specific CRTs. This approach ensures that if all 35 categories are active at the same time, all categories would be displayed on at least one control room CRT. The alarming software also prioritizes the categories at each CRT so that the first category shown is the most important alarm needing attention at this time.

During normal operating periods, that is, no unusual occurrences or disturbances on the bulk power system, some 1200 logged entries can be seen during a given hour. During times of system disturbances, when the operators need the information the most, the operators could see as many as 200 logged entries during each two-second scan because of the increased number of alarms and events on the system. This volume of alarms is not only cumbersome to the operators because the specific data needed are embedded in the large number of

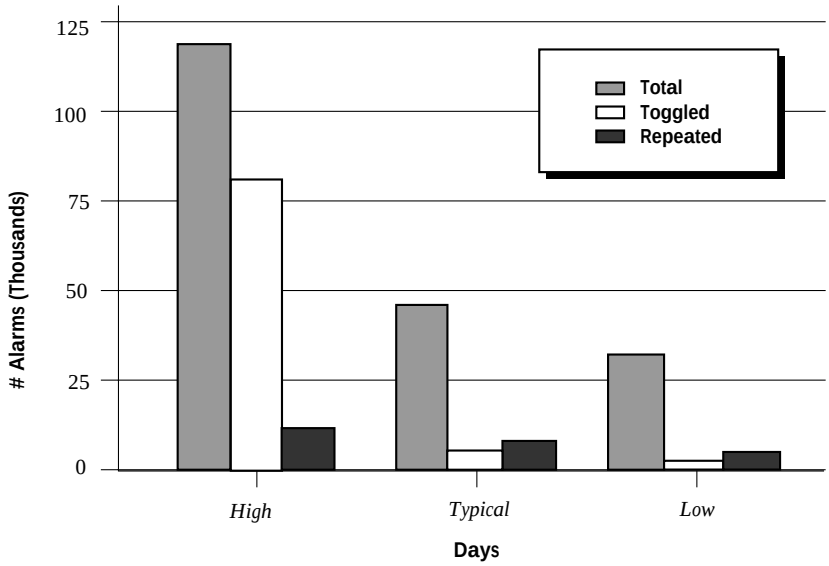


Figure 2. Nuisance (Toggled and Repeated) Alarm Statistics.

```
 seen: 2398 repeated: 1860  toggled: 511
1/16/89 1511:28 LMS  AINSLE ST      4 KV BREAKERS    NORMAL

 seen: 1330 repeated: 1324  toggled: 0
1/16/89 1511:32 LMS  AINSLE ST      4 KV TONE MON    FAILED

 seen: 1155 repeated: 882  toggled: 0
1/17/89 1229:35 PURS M51 MCLEAN AV S CCP R3 SCAN    ALARM UP

 seen: 1104 repeated: 825  toggled: 0
1/17/89 1229:12 PURS M51 W49 STREET  CCP W2 SCAN    CMD TF

 seen: 867 repeated: 725  toggled: 0
1/16/89 2156:57 PVAL FDR F36    96QLV3910    MW TF FAILURE

 seen: 1000 repeated: 462  toggled: 0
1/17/89 1229:35 PURS M51    PURS  SCAN       ALARM UP

 seen: 369 repeated: 353  toggled: 0
1/17/89 1230:01 SC      XFMR 38     VOLTS                    12.0 KV     12.     15.   0.

 seen: 321 repeated: 266  toggled: 0
1/17/89 0003:28 FSKL45  345 KV FSKL45 96QN1397     KV TF FAILURE

 seen: 352 repeated: 183  toggled: 0
1/17/89 0926:43 AST2 MW INPUT FAILURE

 seen: 314 repeated: 173  toggled: 0
1/17/89 0928:41 RAV  GENERATOR 1  STATUS     ONLINE

                    From 11/16/1989 09:17:00
                      To  11/17/1989 12:30:48                    CURRENT CPU: SCADA A
                   CRT                  PC LOGGER
Received:        32016                  65947
    repeated:      10378                   4819
    toggling         514                   6742
Suppressed:        10892                  11561
       %             34%                    17%
```

Figure 3. Filtering Nuisance Alarms.

entries, but it can also slow the logging process and, in severe instances, can affect the entire SOCCS response. This situation is, of course, undesirable and was the justification for undertaking the expert system project.

Overview of the SOCCS Alarm Advisor

The SAA expert system monitors alarms and events observed by SOCCS. Alarms are generated by changes in the power-distribution network topology as a result of discrete operations, measurements of electric values that fall outside preset boundaries, oscillograph operations, telemetry failures, and other conditions. Events reflect the return-to-normal status of the alarms as well as operator actions such as data entry, limit modifications, messages from the New York Power Pool, and other software-generated messages. Entities in the system such as generators, transformers, phase angle regulators, loads, shunts, buses, and feeders are monitored variously and redundantly for watts, volts, vars, and amps.

The first task of SAA is to filter the input stream to remove so-called *nuisance alarms and events*. Nuisance alarms can be the result of an apparent toggling of a discrete (for example, because of bad telephone transmission lines) and can occur thousands of times over a few-minute period. SAA filters out toggling alarms. Similarly, alarms and events that might repeat within a defined time frame are also suppressed. Because these nuisance alarms and events can significantly clutter the input stream, making it difficult for operators to locate other relevant alarms, the use of SAA in this single task can provide substantial benefit. Figure 2 shows that the percentage of nuisance alarms within an alarm stream varies between 13 percent (low day) and 80 percent (high day), with typical days averaging 30 percent. Figure 3 shows that over a 25-hour period, SAA recognized and filtered out 34 percent of the alarms (10,892 out of 32,016) from the CRT display (which only displays alarms).

The second SAA task involves analyzing the system state to determine components out of service, *alive on backfeed* conditions (for example, a feeder is open on one end and closed on another; that is, the feeder is energized but not carrying the load), and other conditions requiring corrective measures. The deduced status of components is compared with available analog data to verify the analysis or find discrepancies (which warrants further SAA alarm analysis through the propagation of alarms through the causal network).

Finally, SAA provides recommended operator actions based on Con

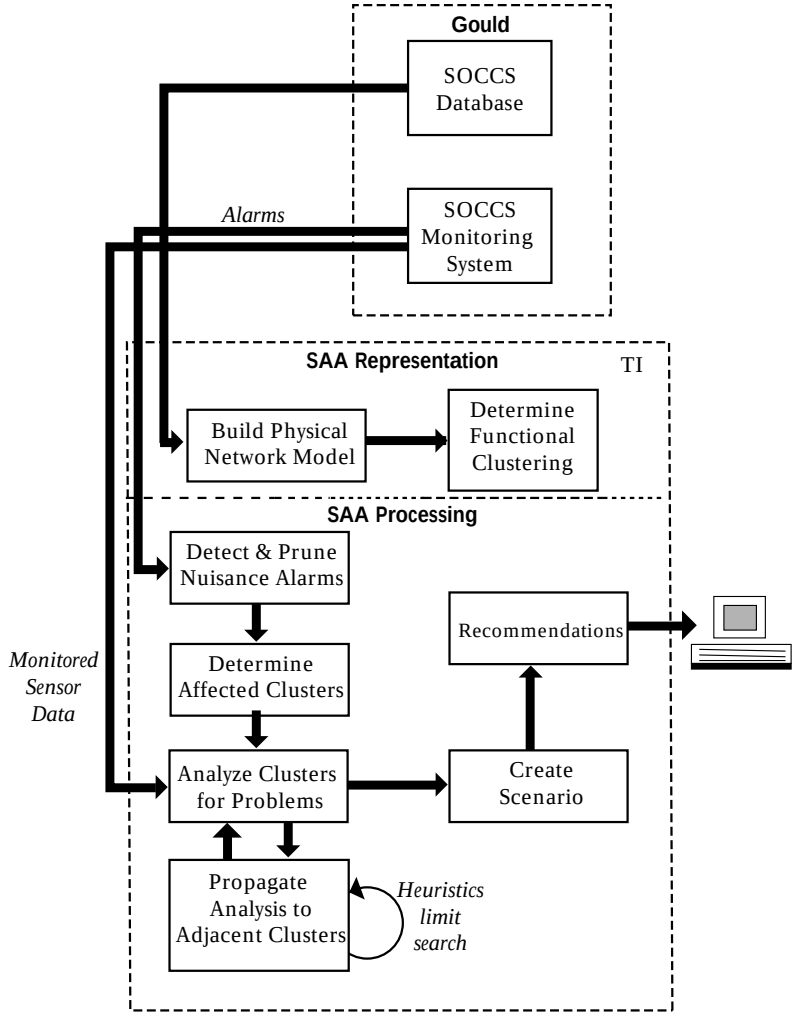


Figure 4. Alarm Adviser Architecture.

Edison's internal procedures. These recommendations are encoded in rules and take into account bus and feeder voltage classes, related alarms, failed operator reclosure attempts, and so on.

Alarm Advisor Requirements and Constraints

SAA's primary goals are to suppress nuisance alarms and provide recommendations for operator actions based on an analysis of alarms, sys-

tem status, and operating procedures. A basic constraint was that the expert system impose no measurable impact on SOCCS processing. No SOCCS software could be modified, all data and alarms from SOCCS had to be automatically transferred to the expert system, and all output from the expert system had to be on separate workstations and not on SOCCS screens.

An additional requirement on SAA was that it process all information in real time. In general, SAA must keep up with the maximum expected data rate. The 69 RTUs are scanned once every 2 seconds by SOCCS, producing a maximum of 200 alarms in each 2-second scan. (Online testing and history analysis confirmed this alarm velocity parameter.) In addition, the operator expects an analysis and recommendation within 20 seconds after the initial alarm trigger. SAA continues to monitor the system, relative to this alarm and the inferred conclusions, for a total of 5 minutes, during which any new data could result in different conclusions and recommendations.

Architecture of the Alarm Advisor

A high-level view of the SAA architecture is found in figure 4.

SAA was developed and is deployed on a Texas Instrument (TI) Explorer II workstation. It is written in the automated reasoning tool (ART) and Common Lisp. The link to the SOCCS host is through Buslink (from Flavors Technology Inc.), which is a direct memory addressing device that establishes physical memory links for reading and writing. Two bus-link connections are used (for the redundant SOCCS systems), and the appropriate failover and real-time machine-recognition routines were written. Fortran and other communications software were provided by TECOSE Inc.

Although bi-directional capability was demonstrated, the initial SAA deployment is a one-way link from the Gould to the TI. Once SAA has been operational for an extended period, Con Edison has the ultimate goal of feeding SAA results back to the SOCCS system. The next phase of SAA will include functions to eliminate nuisance alarms from SOCCS processing. In addition, SAA will execute logging tasks, including those currently done by SOCCS, thus reducing SOCCS system use and effectively extending the life of the SOCCS system.

Alarm Advisor Internal Models

SAA incorporates three internal models: (1) physical, (2) functional, and (3) temporal.

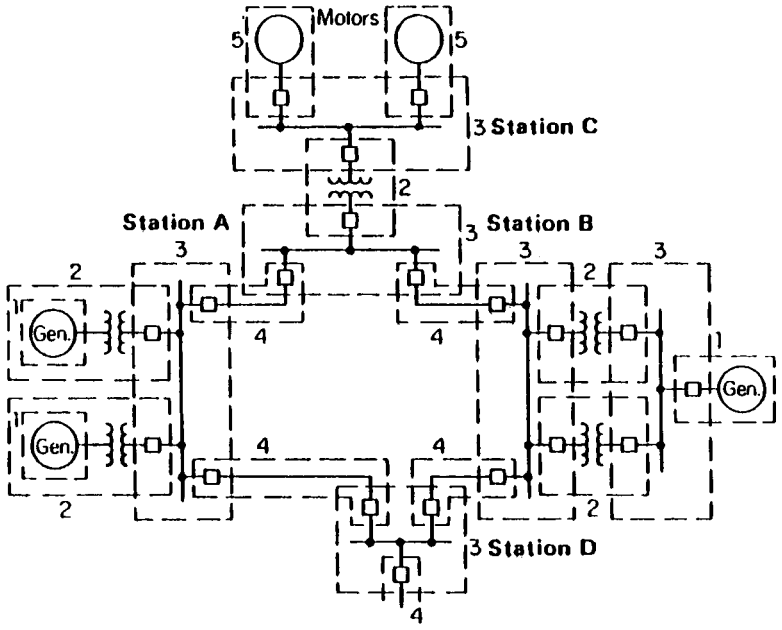


Figure 5. Physical Network with Relay Protection Zones.

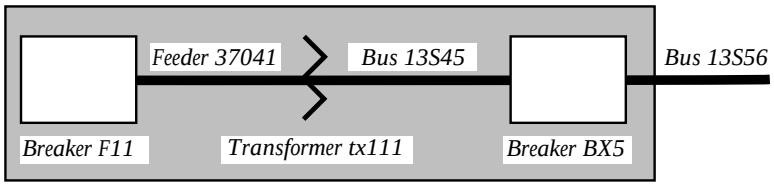
Physical Model

The *physical model* encodes the topological description of the electric power network. It is built from the data points in the SOCCS database. SAA constructs unique objects (also called frames or schemata) based on the concept of electric relay protection zones (figure 5). These objects, called *clusters,* are defined as one or more entities bounded by discrete switching devices. (Figure 6 shows a cluster for the SOCCS data in figure 1.) A cluster is also naturally bounded by generators, loads, or shunts, which are located at the edges of the topology. Any switching device or group of switching devices having a direct physical connection is considered a *switch group).* A switch group's status is determined open if any component device is open; otherwise, it is closed. A switch group connects exactly two clusters. Navigation and propagation from one cluster to another is along the shared switch group.

Functional Model

The *condition,* or state, of a cluster is the functional combination of the state of switch groups at the boundaries of the cluster and is determined by a set of cluster analysis rules. In addition, the *functional model* is used to propagate the effects of the cluster status to neighboring

Name	Type	Entities	Feeders	Buses	Breakers
Cluster 110	*Transformer*	*(tx111)*	*(37041)*	*(13S45)*	*(BX5 F11)*



Internal Number	Point	Entity 1	Type	Terminal	Entity 2	Type	Terminal	Breaker
2434	TPL344	37041	F	1	tx111	T	1	
2435	TPL345	tx111	T	2	13S45	B	1	
243	TPL10	13S45	B	2	13S56	B	1	BX5

Figure 6. Physical Cluster.

clusters. To implement the functional layer, two additional relations are used. A switch-group is *isolated* if it cannot provide power to its neighbor cluster. Isolation results when all other switch groups in the cluster are either open or isolated. Loads and shunts are, by definition, isolated, but they can still serve as sinks (consumers of power). A switch group is a *sink* if it connects to a load or shunt. Sinks propagate recursively, as shown in the following rule:

If a closed switch group is in a cluster containing another switch group that is a sink,
then this closed switch group is also a sink.

Temporal Model

SAA must respond to changing conditions. Conclusions drawn can radically be altered with changes to original parameters. SAA employs a *truth maintenance system* to record logical dependencies between conditions and conclusions. If conclusions are based on conditions that are later retracted, the conclusions are automatically retracted. Thus, SAA maintains a consistent model of the network's state at every moment in time. The *temporal model* also reasons about trends and previous history. The following SAA rule provides an example:

If a Phase Angle Regulator (PAR) hang-up alarm is followed by
an overload on the associated PAR feeder,
then recommend moving the PAR tap or cutting out the appropriate breaker.

Processing Phases of the Alarm Advisor

SAA processing phases consist of (1) identifying nuisance alarms, (2) determining affected clusters, (3) performing cluster analysis, (4) implementing causal propagation, and (5) recommending actions.

Identifying Nuisance Alarms

A set of rules identifies if alarms or events are toggling or repeating; for example:

> If the rate of change of a discrete status exceeds four in six seconds,
> then the discrete status is considered to be toggling,
> and suppress discrete's status changes until a clear period of sixty seconds.

Similarly, alarms and events that repeat within the same or subsequent scans are also suppressed.

Determining Affected Clusters

Alarms pertain to elements located within the physical topology and model. SAA uses hash tables to determine those clusters containing elements involving alarms. These clusters are the initial set analyzed. Hash tables are also used to find the switch groups located within a cluster and, for each switch group, those switching devices contained within it.

Performing Cluster Analysis

Functional model rules determine the state of the entities and breakers within a cluster. Some examples of SAA analysis rules are as follows:

> If all switch groups in a cluster are either open or isolated by adjacent clusters,
> then all entities in the cluster are out of service.
>
> If all switch groups in a cluster are closed and not isolated,
> then all entities in the cluster are in service.

Within certain clusters, it is necessary to analyze the direction of current flow. Transformers, in particular, have primary and secondary sides. A breaker *tripout* (a deenergizing of an electric feeder) on the primary side has a uniquely different effect than a secondary side tripout. Thus, transformer clusters have their own set of rules.

Implementing Causal Propagation

Cluster analysis and *causal propagation* provide a complete description of the system's state (the *scenario*) at any instant of time. Causal propagation is implemented as an incident declaration (fact assertion) in the

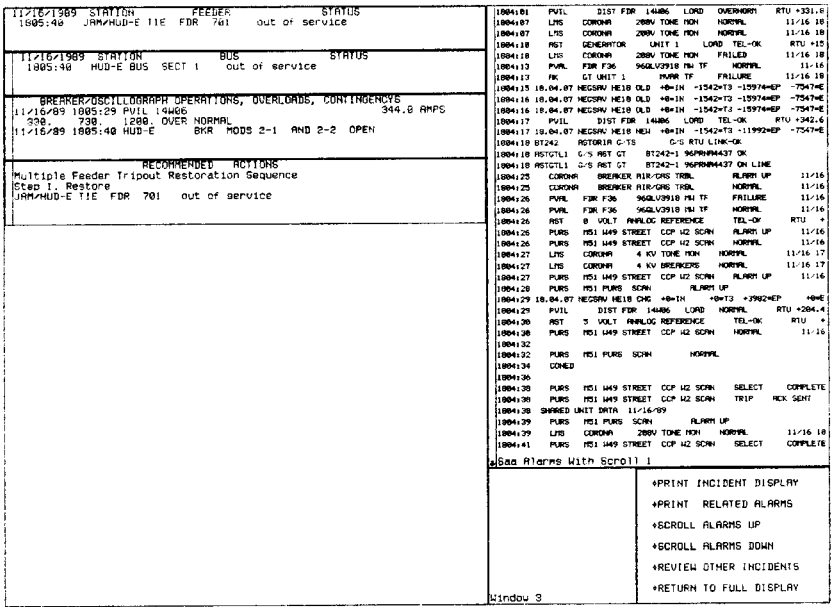


Figure 7. Alarm Advisor Example Recommendation.

adjacent clusters; for example:

If only one switch group in a cluster “c” is closed (all others are open),

and the closed switch group is not isolated from the adjacent cluster (it is seen as a source by its neighbor),

then that switch group is isolated within “c”

and assert an incident in the adjacent cluster.

Implemented in a data-directed architecture, *incident declarations* trigger the initiation of cluster analysis rules for these adjacent clusters. The principal method of reducing search (stopping propagation through the clusters) is *analog confirmation*, that is, when the SAA cluster analysis rules infer conclusions that are supported by analog values; thus,

If analog measurements are inconsistent with the conclusions (analog denial),

then create an incident in all adjacent clusters;

otherwise do not create incidents in the adjacent clusters.

This rule does not mean that analog measurements are accurate. Manual discrete states can be inaccurate, or telemetered discretes can be indeterminate. The point is that if the analogs agree with the conclusions, there is no reason to continue propagation. Otherwise, SAA will continue analyzing neighboring clusters until there is agreement.

Recommending Actions

Recommended actions have been implemented according to Con Edison's operating procedure manual. These actions include, for example, breaker failure recognition and action, transformer spare bank identification, and multiple feeder tripout procedures (figure 7).

Performance and Size of the Alarm Advisor

When an SAA run-time system is initialized, a separate filter process is forked, running at highest priority to read the alarm buffers at the same rate as produced by SOCCS (maximum of 200 alarms per two-second scan). SAA's clock is synchronized with the SOCCS clock. After filtering nuisance alarms, the analysis of an incident is immediately displayed on the operator's workstation; the recommendations are intentionally delayed 20 seconds to allow the electric system to stabilize. SAA contains priorities to ensure that critical activities (for example, reading SOCCS buffers) occur before less critical activities (for example, servicing operator requests for statistical summaries). The use of priorities was sufficient to achieve real-time performance in this application.

When the topology of the electric network is modified (for example, adding a transformer or altering feeder connections), a new SAA system (with a new physical network and other changes) needs to be created. The total number of facts input to SAA from the SOCCS topology points is about 7,800 (3,900 entities and 3,900 discretes). SAA includes about 250 rules to create as many as 2,000 objects (for example, representing clusters); 100,000 facts; and 75 distinct relations. The SAA delivery (run-time) system contains about 70 rules and 3,000 facts and uses about 40 relations. Twenty hash tables are created with a total of 25,000 entries.

Innovations

Innovations of SAA include the first deployed real-time expert system in energy management; one of the first deployed expert systems that meets real-time performance criteria in a time-critical application; and the use of physical, functional, and temporal models to provide a complete and systematic analysis of alarms.

Criteria for Successful Deployment

Three criteria were critical in ensuring SAA's deployment and continued use: First, all information presented by SAA is correct; that is, all data presented, all analyses performed, and all recommendations suggested must be correct. Second, SAA does not lose any alarm data; that

1988	
September	Project initiation
September–October	Functional specification
October–December	High-level design
November	Online data collection
December–January 1989	Kernel expert system
1989	
February–June	Extensive enhancements
June	SAA beta version operational online
July–September	Extensive testing and system extensions
September–November	Final production system online testing
November	Acceptance testing; full deployment

Figure 8. SAA Development and Deployment Time Line.

is, all alarms are processed and analyzed. Nuisance alarms are correctly identified and filtered. Third, SAA response time meets the criteria of Con Edison operators.

Payoff

The major benefits of SAA to Con Edison include (1) a significant reduction in alarms seen by operators (by suppressing nuisance alarms and consolidating alarms); (2) enhanced operator confidence in their incident responses (because of SAA's analysis and recommended actions); and the standardization of operator actions according to Con Edison procedures. In addition, there were some unexpected benefits: SAA identifies inconsistencies in the SOCCS database and generates reports useful to operators and management, for example, statistical summaries, logging reports, and outage reports. SAA also identifies and prioritizes maintenance tasks (focuses maintenance on most severe problem areas identified by SAA).

Development and Deployment Time Line

The development and deployment time line for SAA is shown in figure 8. SAA has been in operation since June 1989. The development life cycle, including the beta period and final production testing, was 15 calender months and involved approximately one full-time Inference person for this entire period. The total individual involvement of the user community was approximately 4 to 5 person-months.