# STATE-OF-HEALTH SOFTWARE FOR THE AUTOMATED RADIOXENON SAMPLER/ANALYZER

Tom R. Heimbigner, Theodore W. Bowyer, James C. Hayes, Charles W. Hubbard, Justin I. McIntyre, Mark E. Panisko, Michael D. Ripplinger, and Reynold Suarez

Pacific Northwest National Laboratory

Sponsored by National Nuclear Security Administration
Office of Nonproliferation Research and Engineering
Office of Defense Nuclear Nonproliferation

Contract No. DE-AC06-76RLO 1830

## ABSTRACT

The Automated Radioxenon Analyzer/Sampler (ARSA) is a complex gas-collection and analysis system that requires constant online monitoring of system operations and overall system health. The software-control system records and monitors and over 100 different system sensors (temperature, pressures, voltages, etc.). A real-time record of the system state allows the system to monitor for unsafe conditions and maintain the system in a safe state regardless of external or internal failures (vacuum pump, valve or power failures, and runaway temperatures are a few examples). Another function of real-time monitoring allows the user to troubleshoot the system when a problem arises, should a minor sensor or a major system failure occur. This paper will outline the general scheme used by the state-of-health program to monitor and assess the system, the graphical user interface programs, and the alert message system, and give specific examples of proper system performance and some systems failures.

## OBJECTIVE

In this paper we describe the development and application of a state-of-health monitoring system. This paper is directed primarily toward the ARSA described by Hayes et al (1999). However, the concepts presented are applicable to nearly any automated system, as outlined in CTBT/WGB/TL-11/5/Rev. 4 (1999), where verification of proper operation, recognition of system failure, identification of a failed component, adjustment of critical timing, and safety-related issues are a concern. A complete description of international noble gas testing can be found in Auer et al., (2004). We have found that the capability of easily monitoring a system's state-of-health is as critical in the development phase as it is once the system has been deployed. The ARSA state-of-health monitoring system is made up of three main components (1) sensor information, (2) event log data, and (3) alerts.

## RESEARCH ACCOMPLISHED

### Description of the ARSA system

The ARSA is a relatively complicated instrument operating autonomously under computer control. It uses Opto22 hardware to perform the necessary I/O. There are an excess of 80 digital outputs used to control valves, apply power to various pumps and compressors, reset various components, and shuttle calibration sources in and out of the system's nuclear detector. Approximately 17 digital input signals are used to monitor source transfer and safety-related sensors such as heater over-temperature signals. The 10 analog output signals are used to adjust three mass-flow controllers as well as to control the system's seven heaters. Finally, there are an additional 44 input signals used to monitor dew points, $CO_2$ readings, mass flow, pressures, temperatures, current, voltages, and a thermo conductivity detector (TCD).

The ARSA system includes an uninterruptible power supply (UPS) that provides power to critical components. This particular UPS can be interrogated to obtain additional information that is critical to proper system operation (input/output power, battery status, load power, and temperature). Early detection of an input power failure is crucial to allow time for critical operations to be performed and the execution of an orderly shutdown to mitigate unsafe operating conditions.

### Components of the ARSA State-of-Health Monitoring System

1. Locally Maintained State-of-Health File

One of the primary goals of any state-of-health monitoring system is to be able assess the "state" of the system at any given time. This provides a historical perspective of the system, allowing a user to review in detail how the system was performing over some previously defined time interval. Furthermore, it allows the user to form comparisons between the day-to-day operations to see if performance is degrading with time. More importantly, however, is having the ability to diagnose the root cause for any decrease in performance that may have been recognized. With a system as complicated as the ARSA, this is not a trivial task. Software solutions providing automated analysis or predictive failure can be extremely complicated. Typically, the analysis of a state-of-health file is performed manually by an experienced operator who is intimately familiar with the system.

The ARSA software architecture implements a client/server model with each server having a responsibility for maintaining one dedicated task. The ARSA state-of-health server's sole responsibly is to interrogate the other servers in the system for the necessary sensor information and write this information to a local binary file. We refer to the ARSA state-of-health file as a "ringfile". The implication here is that after some predetermined time, the sensor data will be overwritten as more and more state-of-health records are accumulated. The ARSA "ringfile" will normally only maintain the previous 13 days of sensor data, sampled at two-minute intervals. Each state-of-health record in the file is time stamped. Because all normal operations performed during the ARSA process control are usually separated by many minutes, or even hours, the 2 minute sampling interval is usually more that sufficient. However, in the development stages of the system, some timing critical information is not yet known and it is important that the developers have the ability to increase the sampling interval to say, 10 seconds, therefore, a finer-grained history is available. This is easily accomplished by a simple command to the state-of-health server.

2. Locally Maintained Event Log

Although the state-of-health file conveys a wealth of information about all sensors in the system, additional information is sometimes required in order to fully understand a particular reading. During system development and testing, the users will often override the normal control system. Although it is true that the state-of-health sensor readings will reflect any changes the user had made, it may not be obvious from the sensor data alone what actually occurred. Similarly, a condition may have existed on an unattended ARSA that forced it to take some action that is outside the norm of everyday operation. Here again, it may not be obvious from the sensor data alone exactly what transpired up to that time. Consequently, the ARSA employs a second software server whose sole responsibility is to maintain a binary Event Log. Whenever any change is made to the system or the control process changes state, an event record is written to this file. Each event record message includes a timestamp, the originator of the message, a priority that was assigned to the message, and a brief description of the message. The ability to review this Event Log is invaluable in assessing the total state-of-health of the system.

3. The ARSA Safety Server

The real-time analysis component of the ARSA state-of-health monitoring software is provided by an additional server called the safety server. This server, unlike the state-of-health and Event Log servers, regularly performs an analysis of the system's health. The sole purpose of this server is ensuring that the system is protected should an unsafe condition arise. It is the only server that has the capability of terminating all other servers. Unlike the state-of-health server, the safety server only interrogates those sensors that are involved in maintaining the safety of the system.

The ARSA must adhere to some very rigid operating parameters in order to preclude any unsafe condition from occurring. To further complicate the problem, many of these parameters are continually changing as the ARSA sequences through the collection, separation, and archiving processes. A few of these parameters are referred to as "global" parameters and are not dependent on the state of the system. Global parameters are always checked. Other conditions, such as the temperatures of the seven charcoal traps, are very dependent on the state of the control system. Traps that are being regenerated or eluted can safely maintain higher temperatures than if they were in the active phase. One final condition, the loss of primary AC power, is also considered. Another unique feature of these parameters is that they have a time-variable condition associated with them. Various perturbations in the system, such as valves opening and closing, can occasionally cause momentary spikes in the sensor data. These spikes are normal and do not necessarily reflect an unsafe condition. The time variable condition ensures that no action is taken unless the condition being tested is out of bounds for the duration of the variable.

If the safety server ever detects an unsafe condition, it will immediately take the required action to ensure that the system is brought back to a safe condition. All servers will be terminated and an alert message placed in the e-mail queue. Additionally, a DO.NOT.RUN file is generated. If for some reason the system is rebooted after a safety shutdown, the ARSA will not start until the condition is corrected and this file has been deleted.

4. The E-mail Interface

Many of the ARSA systems will run unattended. Although there are mechanisms for remotely accessing the system directly (TELNET, FTP, Phindows, or the GUI), this is not the norm. In normal operation, eight files are routinely e-mailed to addresses defined in a configuration file each day. These files include six detector files, an abbreviated state-of-health file, and all records in the Event log that were entered that day. In addition to these files, alert messages may also be e-mailed.

The e-mailed state-of-health file contains only a subset of all the sensors in the normal file. Additionally, this file has been converted to ASCII format that is more suitable for parsing by database applications and Excel. Normally, only the most important analog sensors are included. If analysis of this file determines there is a problem with the system, the entire binary file can be retrieved for a more complete analysis. The e-mailed event log file is also converted to ASCII format.

Whenever a major event occurs on the ARSA, an alert message is e-mailed to the recipients listed in a configuration file. The ARSA will always e-mail an alert message during startup. Once the ARSA system has started, alerts are only sent if the safety server detects a problem with the system. In most cases, these alerts are severe enough that a shutdown of the system is required. The following is an alert message sent by the safety server indicating that a pressure sensor (PS-6) has exceeded its allowable range. Note that the severity of the error is FATAL, indicating that normal operations can no longer continue and that the system will be shut down. Additionally, the alert message contains the date and time that the failure was identified. At this point, we only know that there was a problem with this particular system (USX99). A more thorough analysis of the state-of-health file would have to be performed to determine the root cause of the problem (Figure 1).

```
BEGIN     IMS1.0
MSG_TYPE  DATA
MSG_ID    6859  USX99
DATA_TYPE ALERT
USX99 SYSTEM     2003/06/26 03:41:40
FATAL           (2003/06/26 03:41:40)
a-PS-6 out of range(-40.0-105.0),
value: 115.1, Global


2003/06/26 03:41:40
STOP
```

**Figure 1. Example Alert Message from the ARSA.**

<u>**The System Data Viewer**</u>

A Visual Basic application, the System Data Viewer, was developed by Pacific Northwest National Laboratory (PNNL) to aid in the analysis of the binary files generated by the ARSA systems. This application has provisions for viewing the binary state-of-health and Event Log files. The state-of-health viewer and the Event Log viewer are both structured in a way that allows the user to easily navigate through the files.

When viewing state-of-health data, the user can select any combination of analog and digital sensors for simultaneous display. It is this capability that allows a user, who is familiar with the operations of the ARSA, to evaluate the health of the system. With the inherent zoom capability, the user can examine specific regions of the data in more detail (Figure 2).

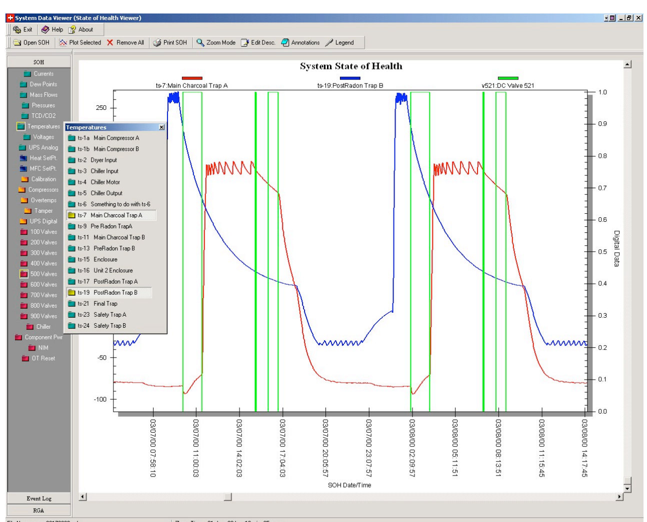


**Figure 2. State-of-Health Viewer Sensor Selection.**

The Event Log viewer displays all the associated records in the file. When displayed, each record is color coded according to the priority of the message. Additional options allow the user to display only messages matching a specific priority. A limited search capability allows users to easily find records such as those that may have been generated by the ARSA safety server (Figure 3).

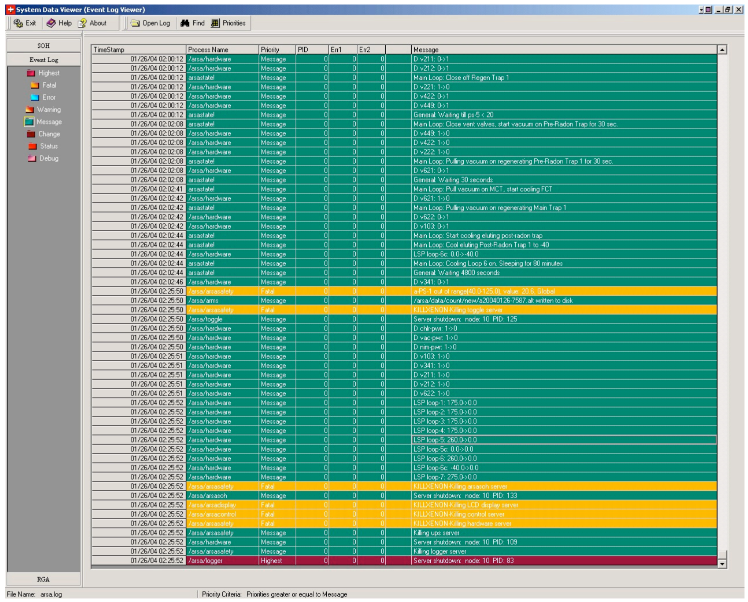

**Figure 3. Event Log View with Priority Highlighting.**

## Examples of Normal Operation

In most instances, when the ARSA is operating normally, state-of-health data are used primarily to verify that system performance is not degrading with time. In many cases, the operator simply may want to obtain a baseline for future comparisons. This is particularly important after the system is initially installed. The ARSA is designed with two distinct processing chains. These are typically referred to as the A and B sides. The purpose of having the two sides is to allow one side to be regenerated while the other side is actively collecting Radioxenon. Ideally, both sides would exhibit the same characteristics. Unfortunately, this is not always the case. There are many reasons why this may not be the case. Subtle differences in the sensors, sensor location, or small details in the manufacturing can all have an affect.

The importance of obtaining a baseline is demonstrated in the Figure 4 below. The temperatures and pressures for both of the main charcoal traps are graphed simultaneously. We recognize immediately that there is very good agreement between the temperature readings for trap A (TS-11, cyan) and trap B (TS-11, green). Both have approximately the same maximum and minimum value and behave similarly over the same time interval. However, when examining the corresponding pressures for the two traps over the same time interval, we notice a distinct difference in the profiles. There is a peak in pressure sensor PS-5 (blue) that does not appear in pressure sensor PS-4 (red). It turns out that this difference in pressure sensors readings has always existed and does not present a problem. If we had not recognized that this peak was present initially, we might suspect that something had occurred on the system and would unnecessarily investigate this anomaly in further detail.

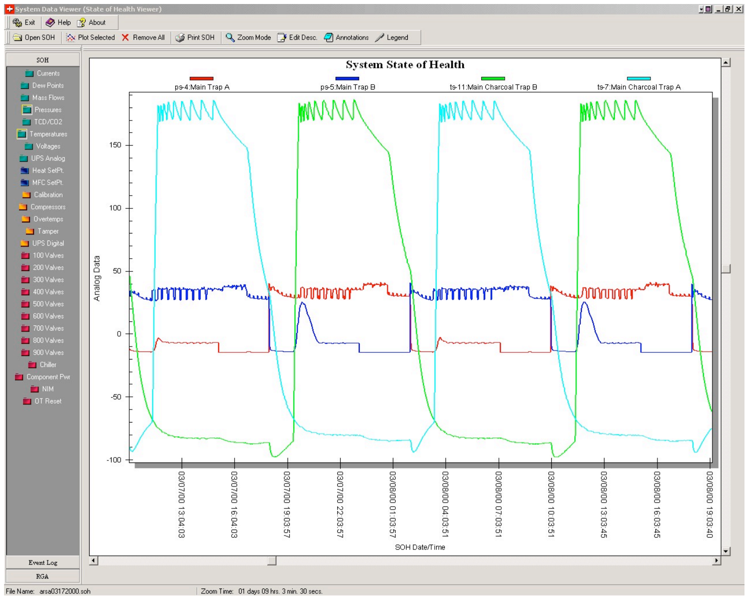


**Figure 4. Normal Sensor Readings with Baseline Requirement.**

## Abnormal Event Resulting in a System Shutdown

While the ARSA system was deployed in Frieburg, Germany, the 24-V power to all the DC valves would occasionally drop out for brief periods of time. Normally, the ARSA would survive these momentary dropouts without incident. The dropouts occurred infrequently, and the nature of the process was such that many of these dropouts occurred at a time when there was minimal impact on the system. On occasion, however, the dropout would coincide with a time in the process when the effects of the dropout were much more pronounced.

In Figure 5 we see multiple dropouts in the 24-V valve supply (red) but see very little perturbation of the dry process pressure sensor, PS-6, shown in blue. The duration of these dropouts is on the order of 1 to 4 minutes. The ARSA survives these dropouts without any problem, and no more dropouts occur for another 9 days.

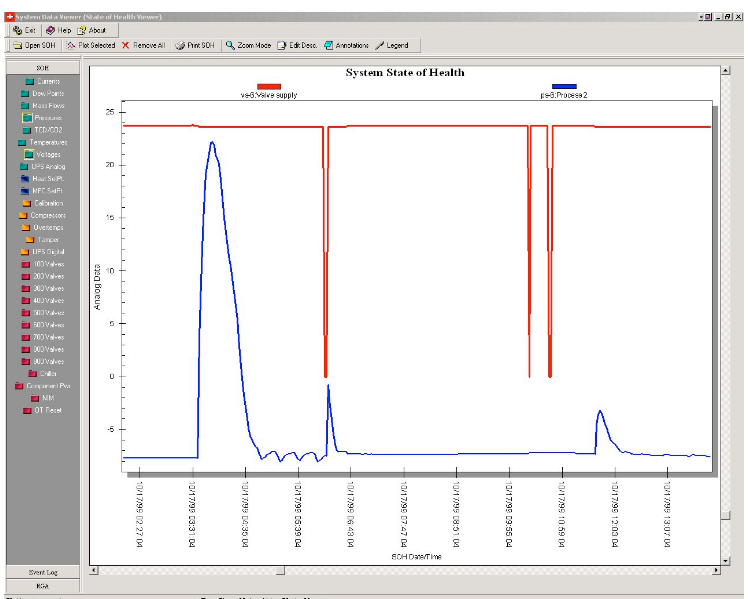


**Figure 5. Valve Power Dropouts Unrecognized by the ARSA**

Nine days later, another dropout in the valve power supply occurs and remains for many minutes. This time the effect is far more dramatic. In Figure 6 below we see that as soon as the dropout occurred, PS-6 immediately rises, and the sensor reading saturates at its maximum valve. The dropout does not recover this time, and the pressure reading remains high for many minutes. The ARSA safety server, recognizing that PS-6 has exceeded its maximum allowable value for the allowable time defined in the configuration file, will initiate the sending of an e-mail alert message and execute an orderly shutdown of the system.

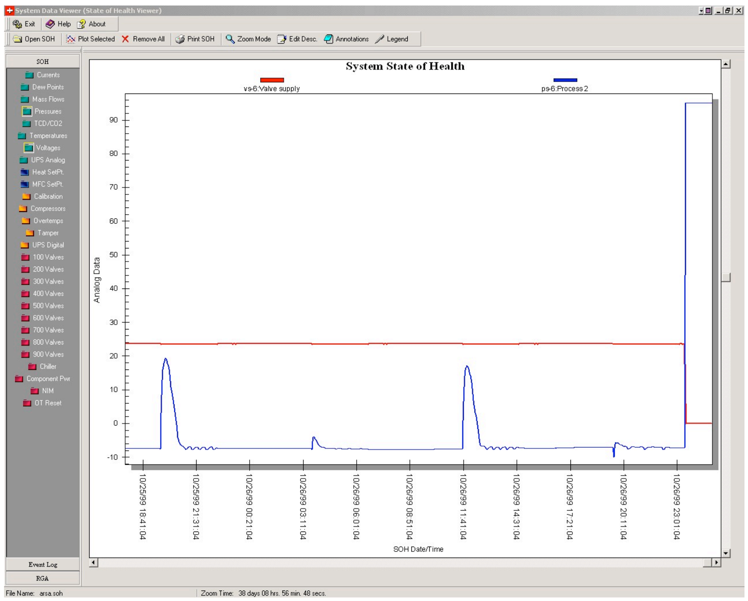


**Figure 6. Valve Power Dropout Resulting in System Shutdown.**

## <u>Sample of an Orderly System Shutdown</u>

When a condition is recognized by the safety server that requires a shutdown of the system, a series of very ordered actions occur. Each action taken is recorded in the Event Log. Figure 7 displays the actions of a typical safety shutdown.

The first line highlighted in orange demonstrates that the safety server has recognized that PS-1 is out of range and has determined that the severity of this message should be FATAL. The following line shows that an alert file (.alt) has been written to the disk. It is this file that will ultimately be emailed by the system. After the alert message is written, the safety server first kills the toggle server so that it is no longer able to cycle the dryer valves. As the toggle server closes, it writes its own message to the log file. The safety server then closes all open valves in the system through an interface to the hardware server. Additionally, the safety server ensures that all heater-control loops have been turned off and that the outputs to the heaters are set to zero. Safety then terminates all other servers and writes the DONOT.RUN file (Figure 7).

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 01/26/04 02:02:46 | /arsa/hardware | Message | 0 | 0 | 0 | | D v341: 0->1 |
| 01/26/04 02:25:50 | /arsa/arsasafety | Fatal | 0 | 0 | 0 | | a-PS-1 out of range(40.0-125.0), value: 20.6, Global |
| 01/26/04 02:25:50 | /arsa/arms | Message | 0 | 0 | 0 | | /arsa/data/count/new/a20040126-7587.alt written to disk |
| 01/26/04 02:25:50 | /arsa/arsasafety | Fatal | 0 | 0 | 0 | | KILLXENON-Killing toggle server |
| 01/26/04 02:25:50 | /arsa/toggle | Message | 0 | 0 | 0 | | Server shutdown: node: 10 PID: 125 |
| 01/26/04 02:25:50 | /arsa/hardware | Message | 0 | 0 | 0 | | D chlr-pwr: 1->0 |
| 01/26/04 02:25:50 | /arsa/hardware | Message | 0 | 0 | 0 | | D vac-pwr: 1->0 |
| 01/26/04 02:25:50 | /arsa/hardware | Message | 0 | 0 | 0 | | D nim-pwr: 1->0 |
| 01/26/04 02:25:51 | /arsa/hardware | Message | 0 | 0 | 0 | | D v103: 1->0 |
| 01/26/04 02:25:51 | /arsa/hardware | Message | 0 | 0 | 0 | | D v341: 1->0 |
| 01/26/04 02:25:51 | /arsa/hardware | Message | 0 | 0 | 0 | | D v211: 1->0 |
| 01/26/04 02:25:51 | /arsa/hardware | Message | 0 | 0 | 0 | | D v212: 1->0 |
| 01/26/04 02:25:51 | /arsa/hardware | Message | 0 | 0 | 0 | | D v622: 1->0 |
| 01/26/04 02:25:52 | /arsa/hardware | Message | 0 | 0 | 0 | | LSP loop-1: 175.0->0.0 |
| 01/26/04 02:25:52 | /arsa/hardware | Message | 0 | 0 | 0 | | LSP loop-2: 175.0->0.0 |
| 01/26/04 02:25:52 | /arsa/hardware | Message | 0 | 0 | 0 | | LSP loop-3: 175.0->0.0 |
| 01/26/04 02:25:52 | /arsa/hardware | Message | 0 | 0 | 0 | | LSP loop-4: 175.0->0.0 |
| 01/26/04 02:25:52 | /arsa/hardware | Message | 0 | 0 | 0 | | LSP loop-5: 260.0->0.0 |
| 01/26/04 02:25:52 | /arsa/hardware | Message | 0 | 0 | 0 | | LSP loop-5c: 0.0->0.0 |
| 01/26/04 02:25:52 | /arsa/hardware | Message | 0 | 0 | 0 | | LSP loop-6: 260.0->0.0 |
| 01/26/04 02:25:52 | /arsa/hardware | Message | 0 | 0 | 0 | | LSP loop-6c: -40.0->0.0 |
| 01/26/04 02:25:52 | /arsa/hardware | Message | 0 | 0 | 0 | | LSP loop-7: 275.0->0.0 |
| 01/26/04 02:25:52 | /arsa/arsasafety | Fatal | 0 | 0 | 0 | | KILLXENON-Killing arsasoh server |
| 01/26/04 02:25:52 | /arsa/arsasoh | Message | 0 | 0 | 0 | | Server shutdown: node: 10 PID: 133 |
| 01/26/04 02:25:52 | /arsa/arsadisplay | Fatal | 0 | 0 | 0 | | KILLXENON-Killing LCD display server |
| 01/26/04 02:25:52 | /arsa/arsacontrol | Fatal | 0 | 0 | 0 | | KILLXENON-Killing control server |
| 01/26/04 02:25:52 | /arsa/arsasafety | Fatal | 0 | 0 | 0 | | KILLXENON-Killing hardware server |
| 01/26/04 02:25:52 | /arsa/arsasafety | Message | 0 | 0 | 0 | | Killing ups server |
| 01/26/04 02:25:52 | /arsa/hardware | Message | 0 | 0 | 0 | | Server shutdown: node: 10 PID: 109 |
| 01/26/04 02:25:52 | /arsa/arsasafety | Message | 0 | 0 | 0 | | Killing logger server |
| 01/26/04 02:25:52 | /arsa/logger | Highest | 0 | 0 | 0 | | Server shutdown: node: 10 PID: 83 |

**Figure 7. Sequence of Safety Server Initiated Shutdown.**

## Recognizing Long-Term Component Degradation

Historically, most component failures in the ARSA have been sudden, catastrophic failures. Compressors suddenly failed to start, heaters have burned out, valves have quit operating, and temperature sensors have failed. For these types of failures, there is very little, if any, warning of an impending failure. Even though current sensors and temperature sensors were attached to the compressors, these sensors did not exhibit abnormal readings before their failure.

There have been some instances, however, where the state-of-health sensor data could have forewarned of an impending failure had it been interpreted correctly. Initially, the state-of-health data from the ARSA was not being stored in a database. The normal access to state-of-health data was through the e-mailed files (daily files that represented only a subset of all sensors) or the binary state-of-health file that had to be manually (by way of FTP) from the system. Because these files represented a relatively short time interval and could only be interpreted accurately by an experienced user, it was very easy to overlook slowly varying changes in a sensor.

The ARSA diaphragm vacuum pumps were a common failure point. Usually, the diaphragm in the pump would first develop a very small tear. This tear would cause a very small, almost imperceptible, rise in the baseline of the vacuum pressure sensor. As the tear continued to develop, the vacuum baseline would continue to increase. At some point, a catastrophic failure in the diaphragm would occur. This very slow rise in the baseline reading was certainly not evident when a single e-mailed file was plotted. The time frame was just too short. In order to effectively examine this data, many emailed files would have to be concatenated together to produce a much longer time span.

## A Unique Problem

While the ARSA was in Germany, a unique problem began occurring on the system. For no apparent reason, the system would simply reboot. Our first inclination was to examine the state-of-health data coming from the UPS, surmising that we may periodically be losing primary AC power and that the UPS was failing during the switch to battery power. However, up to the point of the reboot, all UPS indicators were well within normal operating ranges. The UPS indicated that its battery was fully charged and that it had in excess of 10 minutes of reserve at the current power draw. Additionally, both the line input power and the line output power were operating in normal ranges. Conversations with the personnel at the Institute for Atmospheric Research (IAR) in Frieburg verified that there was not a power failure at the times these problems surfaced. Examination of the event log initially did not help in diagnosing the problem. At the point of failure, there were absolutely no records in the event log that would indicate a problem. After the reboot, the messages immediately before the "System Started" message were normal, everyday messages. Basically, the system was behaving as if it didn't have a UPS at all and as if someone had just disconnected the primary AC from the wall.

Compounding the problem was the fact that it happened infrequently. What we hadn't initially noticed was that although the shutdown was infrequent, it was happening regularly. Although we were receiving e-mailed alert messages that the system had restarted, there was no reason to suspect that these message were related in time. Finally, someone happened to recognize that these alert startup messages were occurring at almost exactly a two-week interval. We wondered what possible component of the system would do anything on a two-week interval. The ARSA control process was fully understood and could not be the problem because it operated strictly on eight-hour cycles. The only conclusion that could be drawn was that the UPS was the source of the problem even though there had never been a loss of primary AC power.

Closer examination of the UPS manual indicated that it was configured to perform a self-test at two-week intervals. When this test is performed, the UPS will switch to battery power even though line input power is available. Although all indicators of the battery status were good, the battery had obviously failed. When current was demanded from the battery, the output AC voltage could not be sustained and the computer would reboot. Although it took a great deal of time to diagnose this problem, it could not have been done without using the combined capabilities of the state-of-health viewer, the Event Log viewer, and the alert messaging system.

## CONCLUSIONS AND RECOMMENDATIONS

The suite of applications and programs developed for the ARSA state-of-health monitoring system has proven to be invaluable in both the development and deployment phases of the project. With few exceptions, problems that have developed on the system are easily diagnosed by a careful examination of the files and alert messages generated by the system. The full implementation of a state-of-health database and the development of automated analysis routines (with predictive capabilities) would fully complete the ARSA state-of-health monitoring system.

## REFERENCES

Auer, M., A. Axelsso, X. Blanchard, T.W. Bowyer, G. Brachet, I. Bulowski, Y. Dubasov, K. Elmgren, J.P. Fontaine, W. Harms, J.C. Hayes, T.R. Heimbigner, J.I. McIntyre, M.E. Panisko, Y. Popov, A. Ringbom, H. Sartorius, S. Schmid, J. Schulze, C. Schlosser, T. Taffary, W. Weiss, and B. Wernsperger, (2004); Intercomparison experiments of systems for the measurement of xenon radionuclides in the atmosphere, Applied Radiation and Isotopes 60 863–877.

Hayes, James C., Keith H. Abel, Theodore W. Bowyer, Tom R. Heimbigner, Mark E. Panisko, Paul L. Reeder, Justin I. McIntyre, Robert C. Thompson, Lindsay C. Todd, and Ray A. Warner, (1999), in *Proceedings of the 21st Seismic Research Symposium: Technologies for Monitoring the Comprehensive Nuclear-Test-Ban Treaty*, LA-UR-99-4700, Vol. I, p. 249.

Operational Manual for Radionuclide and the International Exchange of Radionuclide Data (1999), Preparatory Commission for the Comprehensive Nuclear-Test-Ban-Treaty Organization, CTBT/WGB/TL-11/5/Rev. 4.