Alarm Reduction and Root Cause Analysis for Nuclear Power Plant Control Rooms

Jan Eric Larsson[1], Bengt Öhman[1], Christer Nihlwing[2], Håkon Jokstad[2], Liv Iren
Kristianssen[2], Jon Kvalem[2], and Morten Lind[3]
1) GoalArt, Tunavägen 39C, 22363 Lund, Sweden
2) IFE, OS alle 13, 1777 Halden, Norway
3) Ørsted-DTU, Technical University of Denmark, DK-2800, Kongens Lyngby, Denmark
1) Phone +46 46 192640, Fax +46 46 192641, E-mail janeric@goalart.com

Abstract

In modern control rooms there are often problems with information overload, which means that the operators at certain times may receive more information than they are able to understand and use in their work tasks. The most serious information overload occurs in two types of situations. The first is when the operating state of the plant or process changes, which often gives raise to a large shower of alarms and events, related to the change. Such an alarm shower is expected, but can be dangerous, because it may hide other alarms related to real and unexpected faults. The second, even more dangerous problem occurs when a fault causes several consequential faults, leading to a so-called alarm cascade. Because the alarms in the cascade seldom arrive in correct time order, it can be very difficult to analyze such a cascade, and the information overload occurs in exactly the moment when a potentially dangerous situation starts.

In an ongoing project, GoalArt and IFE are cooperating in testing and evaluating GoalArt's methods for alarm reduction and root cause analysis. The testing comprises two specific algorithms, root cause analysis and state-based alarm priority. The GoalArt system will be integrated with the HAMBO simulator so that operators can evaluate the algorithms on-line.

1. INTRODUCTION

Alarm problems are well known in nuclear and other industries. Already in the seventies, US nuclear regulatory guides gave clear requirements on alarm reduction, identification of primary alarms, and special treatment of alarms of high priority. The requirements were motivated, but at that time, available control technology could not meet them. Since then, technology has improved, but alarm problems have proven to be more persistent than expected. Several different technologies have been tried during different time periods, for example, fault trees, rule-based expert systems, fuzzy logic, neural networks, and simply writing if-then-else logic in the code of control systems. However, the main problem is still here today. The research community has not been able to provide generic diagnostic methods suitable for industrial, commercially viable control system implementation. The situation is well known. Operators are constantly bothered with stray alarms and overloaded with thousands of nearly simultaneous alarms during plant transients. There is hope for improvement, though. There are feasible requirements, see, for example, [4], and research efforts in the eighties and nineties have provided useful results.

This paper describes a pilot project between IFE and GoalArt. The purpose of the project is to test the suitability of GoalArt's alarm handling technologies for control rooms of nuclear power plants, in a large-scale test using the Hambo simulator, at the MTO laboratory at IFE in Halden, Norway. We describe the basic concept of multilevel flow models (MFM) and MFM-based diagnostics. A main MFM concept is that a knowledge base can be built in MFM graphs, easily extracted from the existing plant documentation. No complex rule bases are needed, as all necessary information is handled automatically. MFM development started in the late seventies and has reached industrial application in the beginning of this century, see [2-3, 7-12].

The methods used in this project were previously tested during 2002, in a project between GoalArt, IFE, and the three Swedish nuclear power plants Barsebäck, Oskarshamn, and Ringhals. In this project, it was concluded that the algorithms could provide root cause analysis and state-based alarm prioritization and suppression, in a nuclear power plant setting. Tests were performed with a system comprising some 1000 components and 300 status signals, and with data from a nuclear power plant simulator, see [16].

In the current project, we are working on a larger scale test. The idea is to create a system that will cover the entire Hambo simulator, that is, around 8000 status signals, and to perform tests and evaluations on this system integrated in the Hambo experimental simulator environment. This will provide us with important data on how well the algorithms can handle larger systems and fault situations, how the new information can be presented, and also with more reliable operator tests concerning the usefulness of the new information.

2. ALARM PROBLEMS AND INFORMATION OVERLOAD

Alarm problems have been known since the introduction of control room technology in the sixties. In spite of this, they have been more difficult to solve than expected, and are still present today. We propose that there are two reasons for this.

- There are several different types of alarm problems, and each type of problem needs its own solution. So far, many efforts to solve alarm problems have used only one method or technology, and therefore failed to solve all different types of problems.
- So far, there has been a lack of methods to solve some of the more difficult problems in a technically and commercially viable way. Notably, the problem of consequential alarm cascades needs an efficient method for root cause analysis, and so far, all proposed solutions have demanded a prohibitively large amount of knowledge engineering work.

So when there are alarm problems in a control room, they often consist of a mixture of several different kinds of problems. Some common types of alarm-related problems are:

- The alarm system may be badly designed and the alarm points may be badly placed. In today's computer-based control systems, it is quite easy to add another alarm point, and sometimes there are simply too many unnecessary alarms in the system. The solution here is to perform an alarm system revision and remove unnecessary alarms.
- Alarm limits and other parameters may be wrongly tuned, so that spurious alarms are generated because of noise, etc. For example, if a limit is too tight, signal or process noise may cause the alarm to be activated a large number of times, when there is really nothing wrong, just because the signal is close to the limit and the noise pushes it over. The solution to these kinds of problems is to tune the alarm parameters, limits, and filters, based on either process knowledge or historical trend data.
- Alarms are often designed for a certain operating state, such as, for example, 100% production, while they may be irrelevant for other states, for example, stand-by or emergency shutdown. It is a well-known phenomenon that alarm showers tend to appear during state changes, such as startup and shutdown, and often alarms are

generated from equipment, which is switched off and not in operation. In themselves, such alarms are expected and easily understood, but the problem is that they may hide other alarms, from faults occurring during the state changes. If a fault occurs during a startup, the operators may not see the corresponding alarms, because they were drowned out and lost in the large shower of “usual” alarms. The solution to this problem is to make the alarm generation or presentation state-sensitive, and to suppress those alarms that are irrelevant to the current state. A potentially efficient method for accomplishing this is tested in the current project.

- When there is a fault in a process, it usually causes several consequential faults. If all faults are monitored by the alarm system, a small original fault is usually followed not by one alarm, but often by tens or hundreds of alarms, a so-called alarm cascade. Because the alarms in the cascade seldom arrive in time order, it is often very difficult to analyze and understand the fault situation. The solution to this problem is to apply a root cause analysis to the fault situation, to find out the original fault and the causal chain of events. An efficient method for this is tested in the current project.
- Another type of problem relates to the presentation of alarms in the human-machine interface. If too many alarms are presented, the operators will be overloaded by the information and unable to utilize the information provided. If, on the other hand, to few alarms are presented, while others are suppressed, the operators may not have the right information needed to successfully analyze and understand the situation. The solution here is to design the alarm presentation system so that the more important information is easily available and other information can be hidden or shown according to the current needs of the operators.
- Finally, there may also be alarm problems related to the organization. For example, there should be clear and well-documented decisions about the alarm policy of the company, about the roles and responsibilities of different people and groups, and well-established routines for operation and handling of the plant in fault situations. The solution here consists in creating an alarm philosophy or policy for the organization, and education, training, and problem handling in the group of people operating and maintaining the plant and control system.

Again, the main conclusion to be drawn from the above is that there are a number of different types of alarm-related problems, and each type of problem demands a separate solution. For example, an alarm revision and alarm system redesign may improve the average alarm load, but it will not help against consequential alarm cascades. A new alarm presentation may help the operators to notice a problem quicker, but if there are too many spurious or false alarms, there will still be a problem.

In the current project, we are focusing on three different problems:

- Handling consequential alarm cascades by MFM-based root cause analysis.
- Reducing alarm showers induced by state-changes, using an algorithm called state-based alarm priority.
- Designing new solutions for the alarm presentation based on the new analysis capabilities that the two above-mentioned algorithms could provide to the human-machine interface.

2.1 Consequential Alarm Cascades

The most difficult alarm problem concerns consequential alarm cascades and root cause analysis. In a process, a fault usually leads to several consequential faults. This means that a single fault can create a large number of alarms, a so-called alarm cascade. Because of timing effects, alarm limit tuning, and physical properties of the process, the root cause seldom appears first. Alarm cascades can be very difficult to analyze, which often means that operators cannot easily figure out what is actually wrong with the process.

Alarm cascades are a well-known phenomenon in most control room settings, for example, in nuclear and petrochemical industry, in power grid control rooms, in conventional power plants, process industry, etc. They are a main source of trouble in complex fault situations, since they appear exactly at the most critical moments, and tend to incapacitate alarm systems by producing a large number of alarms.

Alarm cascades can be very large. Major power grid outages are usually preceded by alarm cascades of tens of thousands of alarms, lasting from a few minutes to a couple of hours. In a large power plant, an upset often causes cascades of a few hundreds of alarms, blocking the use of the alarm system for some 1-10 minutes after the upset. Small plants, such, as, for example, a local power plant, will have alarm cascades of 10-100 alarms at a plant trip.

A most difficult complication is the case of multiple, independent faults occurring at the same time, that is, a fault situation consisting of several independent root causes, combined with all the consequences of the different root causes. In such situations, the analysis of the causality is often very difficult, and we believe that it is very easy to commit errors in this analysis.

2.2 State-Dependent Alarms

Some alarms are only relevant for specific operating states. In other states, the alarms are irrelevant and disturbing. Typical examples of this are alarm bursts that appear during startup and shutdown, and alarms from switched off equipment.

Most alarm points have been designed and placed in the process to monitor a specific value, which is important for the process in a certain operating state. For example, a circulation pump is important for the plant during production of thermal energy. But the relevance of the alarm depends on the current operating state. In a stand-by situation, for example, no circulation may be necessary, and the pump is not, and should not be, operating. In a simple alarm system, without suppression logic, this has the effect that there will be several alarms that are activated in states where they are irrelevant. For example, the pump in the example above, may give a low-flow alarm in the stand-by state.

There are two drawbacks of irrelevant alarms:

- Some alarms may actually be missing the real fault situation. For example, in one state, it may be important that a valve is open, while in another state, it may be vital that it is closed. If so, a simple, non-logic alarm will be errouneous in one of the states.
- Even if the alarms produced are not erroneous, but just irrelevant, they will fill up alarm lists and block the human-machine interface. There is a clear risk that, if the operators acknowledge the “usual” list of irrelevant alarms, they might miss to discover a relevant fault that occurred in the same time peridod. A large number of irrelevant alarms may cause the operators’ trust in the alarm system to deteriorate.

The solution to the state-dependency of alarms is to equip the alarm system with a state-estimating algorithm and to assign different priorities to each alarm depending on the current state. The priority should include complete suppression for irrelevant states.

3. MULTILEVEL FLOW MODELS

Multilevel flow models (MFM) are graphical models of goals and functions of technical systems. The goals describe the purposes of a system or subsystem, and the functions describe the capabilities of the system in terms of flows of mass, energy, and information. MFM also describes the relations between the goals and the functions that achieve those goals, and between functions and the subgoals, which provide conditions for these functions. MFM was invented by Morten Lind at the Technical University of Denmark, see [11-13]. Several new algorithms and implementations have been contributed by Jan Eric Larsson at Lund Institute of Technology, see [7-10]. Other developments have been made by Akio Gofuku, see [2-3].

MFM provides a good basis for diagnostic algorithms. The work of Larsson [7] describes four algorithms based on MFM. Measurement validation checks consistency between redundant sensor values, and can discover flow leaks, sensor failures, and other measurement errors. The alarm analysis algorithm analyses any (multiple) fault situation and can tell which faults are primary and which faults that may be consequences of the primary ones. The fault diagnosis uses sensor values and queries to the operator to discover the faults of the target system. The explanation generation algorithm uses the states discovered by the fault diagnosis to produce explanations and remedies in pseudo-natural language. Other algorithms have been developed later. The failure mode analysis uses MFM with added timing information to predict the consequences of failures. It can be used both during the design phase of a plant and in real-time during actual operation. The fuzzy alarm analysis works in a way similar to the discrete alarm analysis, but is based on fuzzy logic, which makes it more robust when faced with noisy signals close to decision boundaries, see Dahlstrand [1]. MFM has also been the basis for validation of consistency in alarm patterns and for FMEA-like consequence analysis, see Öhman [17].

The method is well suited for describing power plants, chemical and petrochemical processes, pulp and paper, power distribution, local heating grids, gas, and several other similar process types.

3.1 An Example of an MFM Model

MFM has been thoroughly explained in Lind [11] and Larsson [7-9]. Here a small example will be given, to show the basics of MFM modeling. We will use a part of the main circulation system of a nuclear power plant. A much simplified process graph, from an example in the master’s project Ingström [5], is shown in Figure 1.

In this system, reactor tank water flows from the downcomer, via the valve V1, to the pump. After the pump, the water flows through the two parallel valves V2 and V3, back to the moderator tank. The pump is cooled by water. There is also a need for a frequency converter for the power to the pump, since the pump is frequency-controlled. Finally, the frequency converter must also be cooled. The purpose of the main water circulation is to control (moderate) the flow of neutrons in the reactor, and to cool it at the same time.

The goals of this simple system are: “maintain desired water flow through the moderator tank,” “cool the pump,” “provide electrical energy with the correct frequency,” and “cool the frequency transformer.”

The functions of the system are, among others, the downcomer's ability to provide water, the pump's ability to transport water, and the heat exchanger's ability to transport heat. An MFM model of this system is shown in Figure 1.

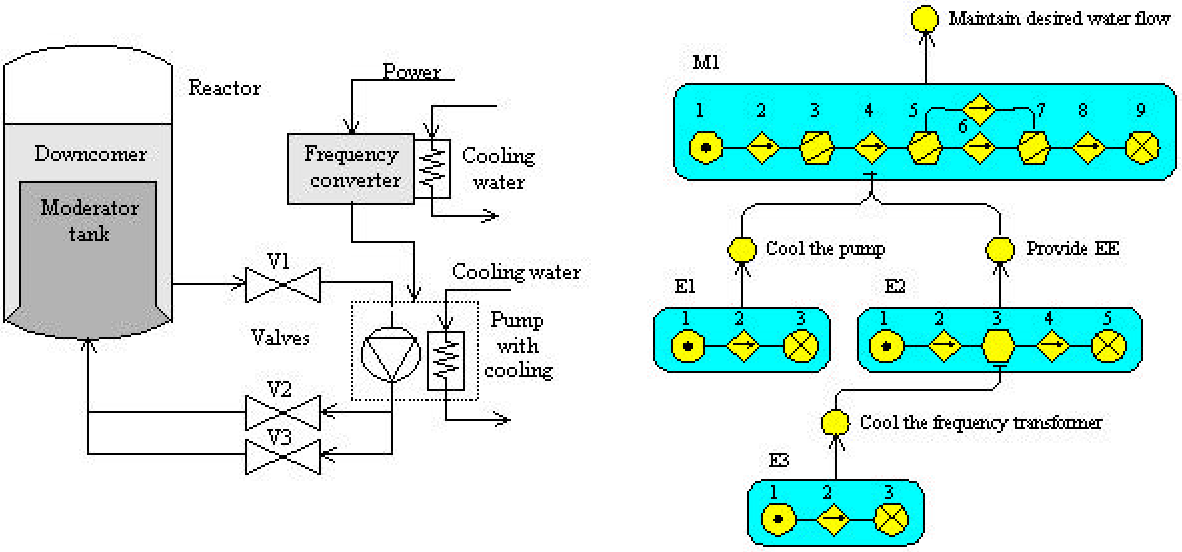

Figure 1. A process graph of the main recirculation system of a nuclear power plant, and an MFM model of the same system.

In the MFM model, there are four flows. The flow network M1 describes the water flow from the downcomer to the moderator tank. The network E1 describes the transport of thermal energy from the pump to the cooling water. The network E2 describes the flow of electrical energy from the supply, via the frequency transformer, to the pump. Finally, the network E3 describes the flow of thermal energy from the frequency transformer to the cooling water. Thus, M1 is a model of a mass flow, and E1 to E3 are models of energy flows. In the network M1 the functions are, from left to right: 1) a source of water, realized by the downcomer; 2) a transport, realized by the valve V1; 3) a balance, realized by the pipe between V1 and the pump; 4) another transport, realized by the pump; 5) another balance, realized by the forking pipe between the pump and the two parallel valves V2 and V3; 6) two transports, realized by the valves V2 and V3; 7) a balance, realized by the pipe sections between V2 and V3, and the moderator tank; 8) a transport, realized by the pipe that runs into the moderator tank; and finally, 9) a sink, realized by the moderator tank. The networks E1 to E3 contain energy flow functions describing the flows of electrical and thermal energy.

It should be noted that MFM describes how different flows enable each other. In the simple example in Figure 1, it can be seen that the cooling water flow E3 is necessary for the proper function of the frequency converter, and that the cooling water flow E1 and the electrical flow E2 are needed to keep the main water flow operating.

It is also important to observe that the given example is a very small toy example. MFM is designed specifically to handle large models with thousands of objects or more.

3.2 Advantages of MFM Algorithms

The algorithms described in Larsson [7] are based on discrete logic where the “sensor” values are low, normal, or high, and the resulting values are consistent or inconsistent, working or failed, primary or consequential, etc. In other words, MFM uses a linguistic interpretation of logic variables, just as do rule-based expert systems and systems based on fuzzy logic. In addition, the MFM algorithms all operate by searching in fixed graphs. We have aimed at always producing algorithms that can handle the full MFM syntax, including closed loops in both the flows and the means-end dimension, as well as every kind of multiple fault situation. In addition, these complex cases should be handled by search methods of linear or sublinear complexity. So far, all of our presented methods fulfill these requirements. Together with the discrete logic, explicit means-end concepts, and graphical nature of MFM, this gives several advantages:

- The explicit description of goals and functions gives a small semantic gap between the diagnostic task formulation and the knowledge representation.
- The graphical representation provides strong support for knowledge base overview and consistency, and there is no need for a specialized knowledge engineering tool.
- The high level of abstraction makes knowledge acquisition, knowledge engineering, and knowledge base validation and support considerably easier than with standard rule-based systems or fuzzy logic systems.
- The graphical nature of the models allows the algorithms to have good real-time properties, such as an easily computed worst-case time, low memory demands, and high efficiency.
- The high level of abstraction allows the algorithms to be very fast. A worst-case fault diagnosis on the smaller nuclear example, takes about 4 miliseconds on an 800 MHz PC.

These advantages have been observed in practice, during the previous nuclear project, see [16].

3.3 Root Cause Analysis

An MFM-based algorithm called alarm analysis calculates the root cause (or causes) of any alarm cascade, and thereby helps the operators to understand the current fault situation. The algorithm can handle all theoretically possible combinations of root causes and consequential faults. It reads real-time data from the existing control system and uses multilevel flow models (MFM) for its analysis.

There are two advantages of the MFM-based root cause analysis, which we hope to prove in the ongoping project:

- The algorithm can handle all theoretically possible input combinations, including multiple independent root faults, and circular dependencies. The inputs are limited to low faults, normal states, and high faults. If this limitation is taken into account, and the MFM model gives an accurate description of the physical process, all potential fault scenarios will be correctly analyzed by the algorithm, including those scenarios never seen before.
- Building an MFM model is a relatively easy task, compared with producing a system with similar capabilities using fault trees and a rule-based expert system. The previous nuclear model comprises some 1000 components and took 1-2 man-months to build. In a previous project, a system of similar size, based on fault trees, took around 10 man-years to build, see [14]. We state no exactly fairness in this comparison, but we believe that MFM modeling is quite efficient.

3.4 State-Based Alarm priority

An algorithm called state-based alarm priority (SBAP) reads real-time data from the existing control system, tracks its current operating state, and gives each alarm a dynamic priority depending on the current operating state. This includes complete suppression of irrelevant alarms. The method is not based on MFM, but uses an Excel-like editor for graphical input of alarm priority data, see Figure 2.

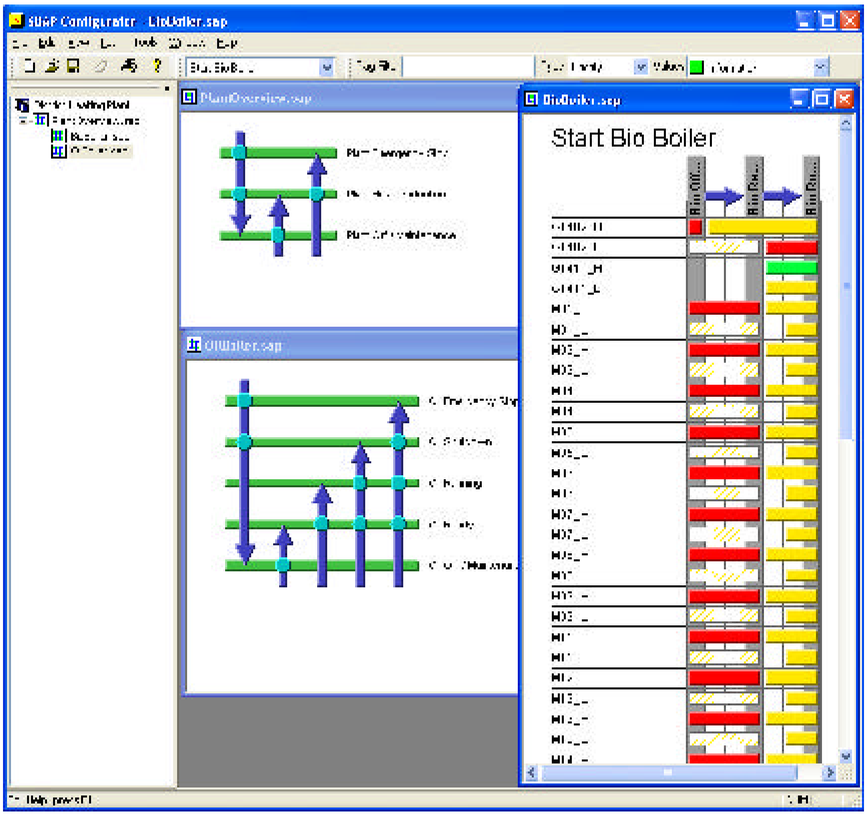

Figure 2. Screen layout from SBAP Configurator.

The runtime part of SBAP estimates the current operating state of a plant or sub-process, and assigns a priority to each active alarm, including complete suppression if the alarm is irrelevant in the current situation. The main advantages of this method and tool are:

- It is a fairly easy task to use the graphical building tool to enter all the state-dependent priority information. In the project, we hope to be able to find out whether using this environment is more efficient than writing logic expressions directly in code.
- The graphical tool also gives a good overview of the varying priorities in successive operating states. In this way, it becomes easier to validate the entered knowledge.

4. HUMAN- MACHINE INTERFACE

The current project involves test and evaluation of the new algorithms for root cause analysis and state-based alarm priority. But in order to use this new functionality, and to test and evaluate it in the project, we also need to present the new information in an efficient way. The new information available is:

- Knowledge on root causes and consequences, that is, the entire causal chain of alarms and events.
- A dynamic priority, varying with plant state, and including a priority value of “irrelevant”,” where such alarms should be suppressed.

In the previous project, a new type of alarm list was tried out, see [16]. It contains four separate lists, one for alarms with critical priority (root cause *or* consequential), one for root cause alarms, one for consequestial alarms, and one for expected but missing alarms (which is also calculated by SBAP), see Figure 3.

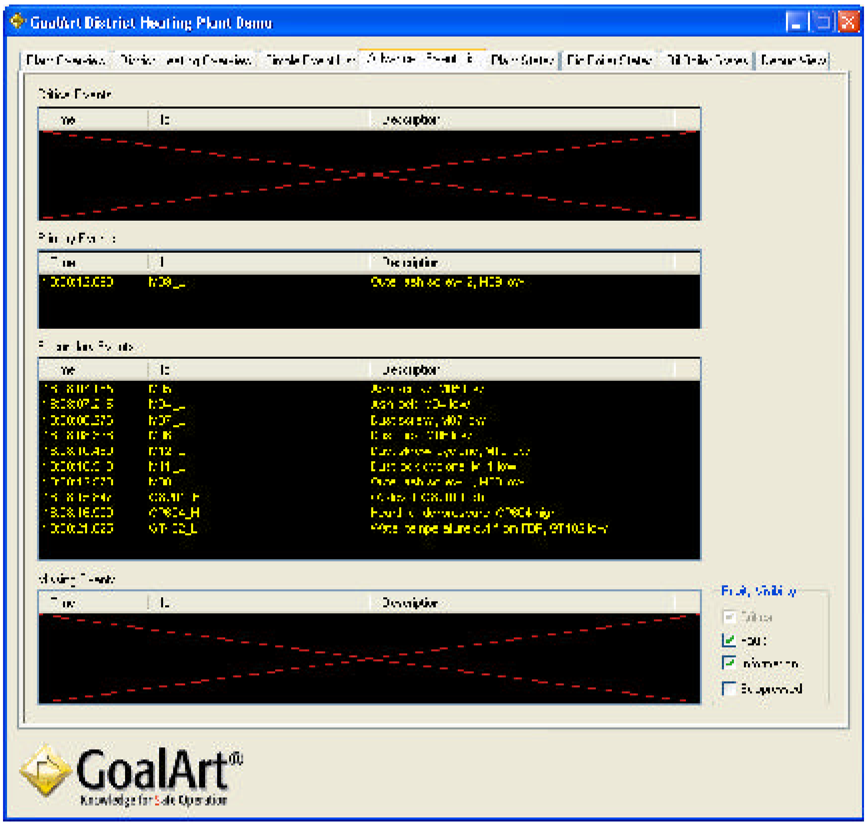
GoalArt District Heating Plant Demo
Description
Description
Description
Description
GoalArt®
Knowledge for Safe Operation

Figure 3. Alarm list with four sub-lists, showing critical alarms, root causes, consequences, and missing but expected alarms.

In the current project, we intend to develop and test several new ways of presenting the new information available.

5. TASKS OF ALARM SYSTEMS

An alarm system should provide operators and maintenance personnel with structured information. This simple statement means that the operator should be able to extract exactly the information required for the particular state of the plant operation. In case of a plant transient, the operator will probably get an alarm cascade of several hundred alarms, and will have no time to study the complete alarm list. The information to be extracted concerns the following:

- What initiated the transient?
- Which parts of the plant are still available?
- Are there high priority alarms, which must be dealt with immediately?

The situation will be still more complicated if the plant moves into unstable conditions. The initiating condition temporarily disappears, and the initiating alarm burst will periodically be replaced by other bursts, not apparently connected to each other. The plant protection system will further complicate the alarm situation, especially if faulty sensors initiated it. All those scenarios show that alarm situations have their own dynamics, which must be recognized and addressed accordingly.

The first action of the operators is normally to find out exactly what has happened and what parts of the plant are available. If that process would take too long time, the operator team leader responsible for safety will decide to take the plant over to a fail-safe state of operation, through reduction of the load, load rejection, or through emergency stop of the plant. An unnecessary stop of the plant means lost revenues, especially for emergency stop requiring at least 24-hours to start again.

The primary task of the improved alarm system will be to provide direct answers for the three queries listed above. The answers should be presented in a clear graphical or verbal formulation, allowing operators to take the right decisions in an orderly way, in an early stage of the event development. The presentation will address the dynamic aspects of the alarm cascades mentioned above.

6. THE HAMBO SIMULATOR

The HAMBO simulator is an experimental simulator located at IFE in Halden, and the simulator is used for performing human performance experiments, as well as testing of operator interfaces, alarm systems, and other operator support systems. The simulator's reference plant is the Forsmark 3 nuclear power plant in Sweden, and the simulator comprises around 8000 status signals. The aim of the project is to cover all of these signals, using both MFM-based root cause analysis and state-based alarm priority. The project also includes development and testing of new ways of alarm presentation, using the extra information available from the root cause analysis and state estimation. Concerning the existing alarm system of HAMBO, see [6, 15].

7. CONCLUSIONS

The aim of the alarm project is to reach a full integration of the alarm analysis and state-based alarm priority in the HAMBO simulator, so that the new algorithms can be tested under realistic circumstances on-line with real operator crews. In this way, one will be able to obtain measures of the usefulness and efficiency of the alarm reduction and situation assessment provided by the specific alarm algorithms.

8. REFERENCES

[1] F. Dahlstrand, *Methods for Alarm Reduction with Multilevel Flow Models of Industrial Processes,* Licentiate thesis, Department of Information Technology, Lund University, Lund, Sweden, 2000.

[2] A. Gofuku, T. Ohi, and K. Ito, “Qualitative Reasoning of the Effects of a Counteraction Based on a Functional Model,” Proceedings of CSEPC 2004, Sendai, Japan, 2004.

[3] A. Gofuku and Y. Tanaka, “A Combination of Qualitative Reasoning and Numerical Simulation to Support Operator Decisions in Anomalous Situations,” Proceedings of the 3rd IJCAI Workshop on Engineering Problems for Qualitative Reasoning, 1997.

[4] IEC 62241/CD “Nuclear Power Plants – Main Control Room – Alarm Functions and Presentation”, CD 2001.

[5] D. Ingström, “MFM Modeling and Alarm Analysis of the Barsebäck Nuclear Power Plant,” Master’s thesis, Department of Information Technology, Lund Institute of Technology, Lund, 1998.

[6] T. Karlsson, “The Alarm System for the HAMBO BWR Simulator”, Halden Report, HWR702, 2000.

[7] J. E. Larsson, *Knowledge-Based Methods for Control Systems*, Doctor’s thesis, TFRT–1040, Department of Automatic Control, Lund Institute of Technology, Lund, 1992.

[8] J. E. Larsson, “Diagnostic Reasoning Strategies for Means-End Models,” Automatica, Vol. 30, No. 5, pp. 775–787, 1994.

[9] J. E. Larsson, “Diagnosis Based on Explicit Means-End Models,” Artificial Intelligence, Vol. 80, No. 1, pp. 29–93, 1996.

[10] J. E. Larsson, “Diagnosis Reasoning Based on Explicit Means-End Models: Experiences and Future Prospects,” Knowledge-Based Systems, Vol. 15, No. 1-2, pp. 103–110, 2002.

[11] M. Lind, “Representing Goals and Functions of Complex Systems — An Introduction to Multilevel Flow Modeling,” Technical report, 90–D–38, Institute of Automatic Control Systems, Technical University of Denmark, Lyngby, 1990.

[12] M. Lind, ”Modeling Goals and Functions of Complex Industrial Plant,” Applied Artificial Intelligence, Vol. 8, No. 2, 1994.

[13] M. Lind, “Plant Modeling for Human Supervisory Control,” Transactions of the Institute of Measurement and Control, Vol. 21, No. 4-5, pp. 171-180, 1999.

[14] A. B. Long, “Technical Assessment of Disturbance Analysis Systems,” Nuclear Safety, Vol. 21, No. 38, 1980.

[15] D. Roverso, “ALLADIN Run-Time and HAMMLAB Applications”, Halden Report, HWR691, 2000.

[16] J. Tuszynski, J. E. Larsson, C. Nihlwing, B. Öhman, and A. Calzada, “A Pilot Project on Alarm Reduction and Presentation Based on Multilevel Flow Models,” Proceedings of the Enlarged Halden Programme Group Meeting, HPR-358, Storefjell, Gol, Norway, 2002.

[17] B. Öhman, *Real-Time Diagnosis of Industrial Processes Using Multilevel Flow Models*, Licentiate thesis, Department of Information Technology, Lund University, Lund, Sweden, 2001.