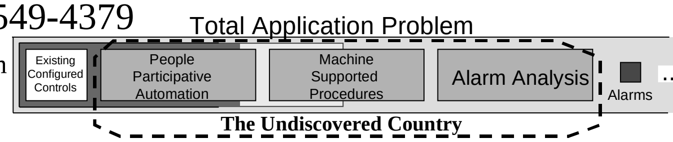
Models and Support for Alarm Operation; Batch vs. Continuous

E. H. Bristol
The Foxboro Co.
Dept. 0318, Bldg. C41-1H
Foxboro MA, 02035
Tel. (508) 549-2019
Fax: (508) 549-4379
email: ebristol@foxboro.com



KEYWORDS

Batch, Sequencing, Alarms, Patterns, Policies, One Word Summaries, Battle Ribbon Trending

ABSTRACT

Alarms are traditionally handled as standard contacts or parameters associated with process sensors, with uniform implementation, but no application related Alarm Management organization or strategy. For continuous processes, where there is a possibility of independent Alarmed events, in different parts of the production, this has been an easy, lazy approach to Alarms, depending on operator experience and discretion. In this context, Alarms are the main connection from the automation to the operator, for addressing abnormal process operation. Alarms of this sort address problems that we have been unable to automate. But they are often as much a source of consternation as help. The Alarm problem is thus twofold: How to make traditional Alarms more useful, and how to provide a more engineered approach to Alarming. For the former, the paper suggests simple Alarm tools requiring only the most basic configuration. These include Trending, automatically generated situation descriptions, causal Alarm analyses, and situation adaptive Alarm operation. But Batch control particularly lends itself to better engineering of the relative role of the operator and automation because the production steps follow a natural progression with expected normal and abnormal consequences. In this case, Alarms and their response can be integrated into the natural sequencing.

INTRODUCTION

Process control alarms are the main automation system vehicle for dealing with operations too abnormal for realistic automated accommodation. These abnormalities will often be unfamiliar to the operations and applications personnel. Some will be very mild, some catastrophic. Large continuous applications may have thousands of potential alarms defined and hundreds active at any point in time, sometimes within one or more causal showers. Traditional alarms, however well designed, are treated as limited sensor attributes, rather than coherent control elements. They are inherently inadequate to support the operator in recognizing and coping with significant abnormal or crisis situations. The push to operate with fewer operators places that much more of a load on the existing ones, exacerbating the problem. The initial tool discussion focuses on improving the analysis and presentation of this traditional alarm information. The later discussion of Batch control discusses how this kind of operation naturally encourages a more rationalized approach to Alarms.

Alarms fall in a number of classes, depending on the extent to which their role is well understood:

1. Alarms reflecting inherent safety or economic limits on the plant, to be avoided, by their very nature.
2. Alarms, for which well-defined corrective operating procedures exist (These alarms may well fall in the first class as well.).
3. Alarms, whose importance and meaning is situation dependent (To the extent to which the situations are detectable and recognized, these alarms may also fall in the first two classes.). Batch by its nature extends this class.
4. Alarms, which generally reflect qualitative constraints, not well formalized. Ideally these alarm limits would be periodically tightened, as operating experience justified it, in an SQC kind of approach.

The first two kinds of alarm are the result of good engineering, and subject to it. They should be included with the orderly automation design, and are improved by better automation[1] and better rationalization of the operator's role. The last two kinds of alarm constitute the greatest problem to orderly, continuous operation. And yet, properly supported, they may offer the greatest opportunity to profit from any operating discretionary capability. The first part of the paper will address Tools designed to better support the operator in coping with large amounts of this kind of Alarm information, most useful in predicting and diagnosing abnormal process states.

Excepting a few alarms with well-defined (but not automated) operating responses, most of the alarm activity will thus be coping with deficiencies in the process automation and operator understanding, and in the adjustment of the alarms themselves. The Alarm Analysis Tools must minimize these deficiencies, but they will function better with better application automation. They presume a general Process Control vantage point, neither Batch nor Continuous.

The Tools are optimized about poorly understood process characteristics, requiring easy sensitivity to unusual behavior, and rapid exploration and diagnosis. Accordingly, they should not be overloaded with configuration or operating features. In particular, the Tools must avoid complex, configured, alarm specific responses more appropriate to well-understood, systematic plant automation. The related configuration of the alarms and their associated displays should be governed by generalized display Policies applicable to groups of alarms rather than by detailed configuration of the individual alarm action. The display approach must support quick, easy, transparent operator action.

Processes with more systematic relations between process events and possible alarms support easier formalization of alarms and the operating role, perhaps with traditional automation, perhaps with Expert Systems. For example, the large continuous plant has all aspects of production running at the same time. The alarms are based on individual process variables. Event/Alarm relationships may be localized to the part of the process in which the causal event may occur. But multiple trains of events may occur at the same time in different parts of the process, merging together in an overall alarm flood or shower. On the other hand, as discussed in the later part of the paper, the events of a batch plant are ordered, sequentially in time, giving rise to greater predictability in any associated alarms. This simplifies operator diagnosis and the development of standard procedures. It also permits alarming of specific events independent of generalized alarming standards.

Batch also emphasizes certain issues not normally recognized in continuous control, in particular the dynamic variation of operational conditions with changing operational phases. The Alarm Tool Category concept is the most basic vehicle for making the operation phase dependent. Most of the accommodation

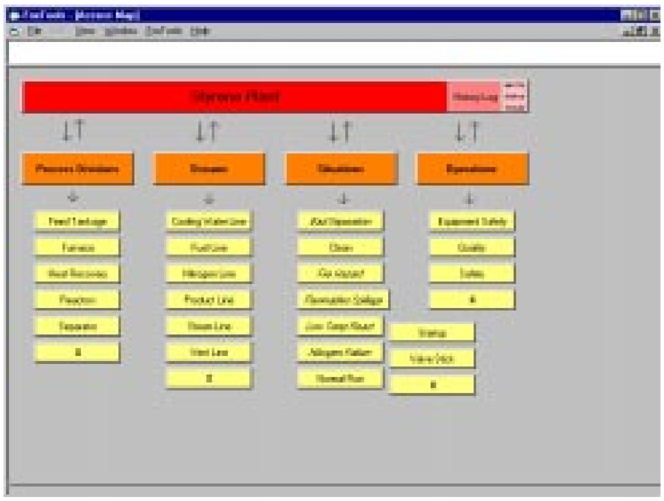
---

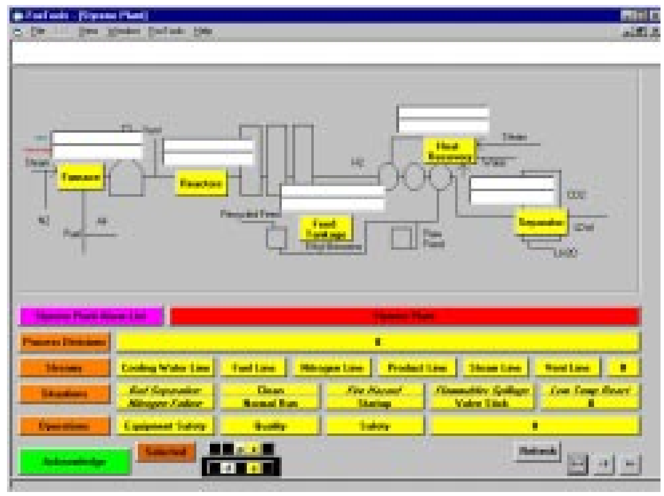
[1] Which requires the invention and commercialization of automation tools designed to make the configured automation more easily understood and implemented.[1-3]

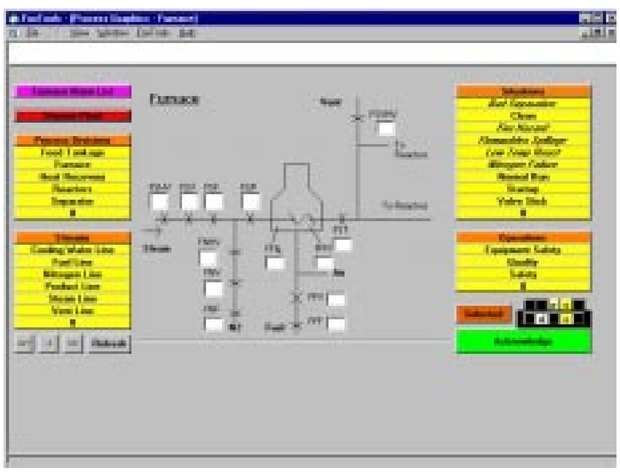
of the Tools to the Batch environment will be through the APIs which allow the construction of Batch system commands or links to control the status of associated Categories from the Batch procedure. Batch will motivate different attitudes for combining the configuration of controls and Categories, and for archiving prior Category configurations.

THE ALARM TOOLS

The accompanying demonstration develops a continuous Styrene Plant example, showing an application based hierarchical organization.[1] The demo expresses a basic set of graphic displays including only alarm related data, which in real life would contain all kinds of information. The demo also shows a number of default displays, which would exist even when a graphic was not configured.







The design concept is based on simple hierarchical Categories of alarms, extended with as many different uses of these Categories as possible. The Access Map includes a Process Divisions based hierarchy. But it also shows other hierarchies as well, based on the process Streams or Situations (in Batch: Phases) or Operations. The resulting multi-hierarchy is intended to be fully user configured: The alarmed process variables are user-defined, as well as the choices of higher level Categories in the complete structure. Each of the demo Categories reflects possible (but not required) user naming and content.

CATEGORIES, CATEGORIZATIONS, AND PATTERNS

The whole system is based on three mechanisms for hierarchically classifying Alarm conditions:

- User defined Categories, which are lists of alarms (listed in the default display for any Access Map button except the Styrene Plant, Feed Tankage, Furnace, Heat Exchanger, Reactor, and Separator buttons with their graphics).[2]

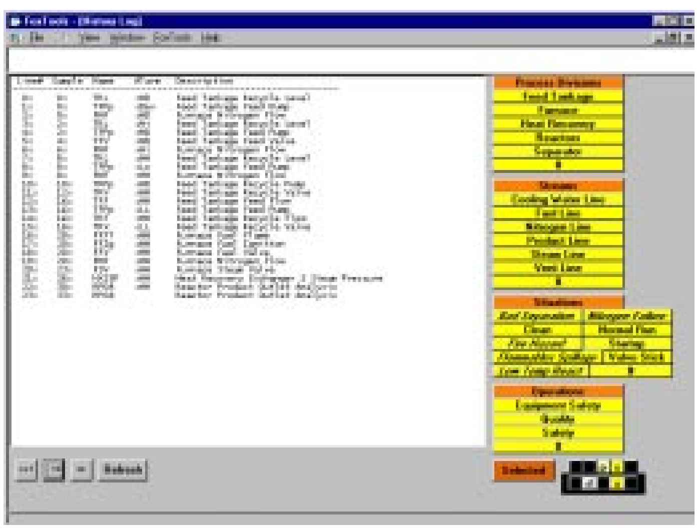
- Categorizations, which are top level nesting Categories, whose subCategories collectively contain all of the application alarms. They represent different ways of completely dividing the process alarm variables. Knowing the Category locations of an alarm in each of the different Categorizations is like knowing the coordinates of a point in space. (The # marked buttons, in the Access Map, correspond to unnamed Categories whose alarms fill out the associated Categorization.) The Categorizations support the hierarchical display access. They also support the Mask/Filter, active in most of the displays (e.g., the Styrene Plant and Furnace graphics above), as will be defined shortly.
- Patterns, which are Categories, whose alarms make up a symptom pattern, and whose order of occurrence, in a pre-defined time window, is presumed significant. They are divided into Cause Alarms, and Main (important) and Secondary Effect Alarms. In the Access Map (and Mask displays), their buttons are distinguished in Italics.

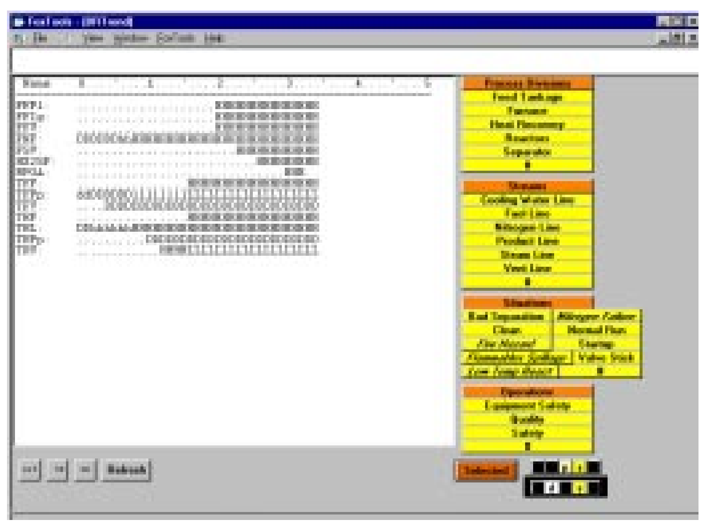
BASIC ALARM RECORDING: LOGGING AND TRENDING; SIDE LOGS

Traditional Alarm display takes three forms: Alarm Panels, Indicators in Graphic Panels or Displays, and Logs. Of these the simplest (below left) is the log, shown in its role as an absolute archival record of events for later analysis. But even here, and particularly as a vehicle for online diagnosis, the log has a lot of defects:

- The events of a given variable are spread among many log entries,
- The log is hard to search for different alarmed variables or events,
- Events for a given variable are hard to correlate.

For its basic archival and post situation analysis purpose, the log may be adequate. But it is hardly an operational tool. For operations, the Tools provide a trending capability, which supports each alarmed variable with its own (in this case letter coded) trend line. The demo codes the six kinds of alarm events





(Deviation, High Deviation, Lo, Hi, Lo Lo, and Hi Hi) with appropriate lower or upper case letters. More

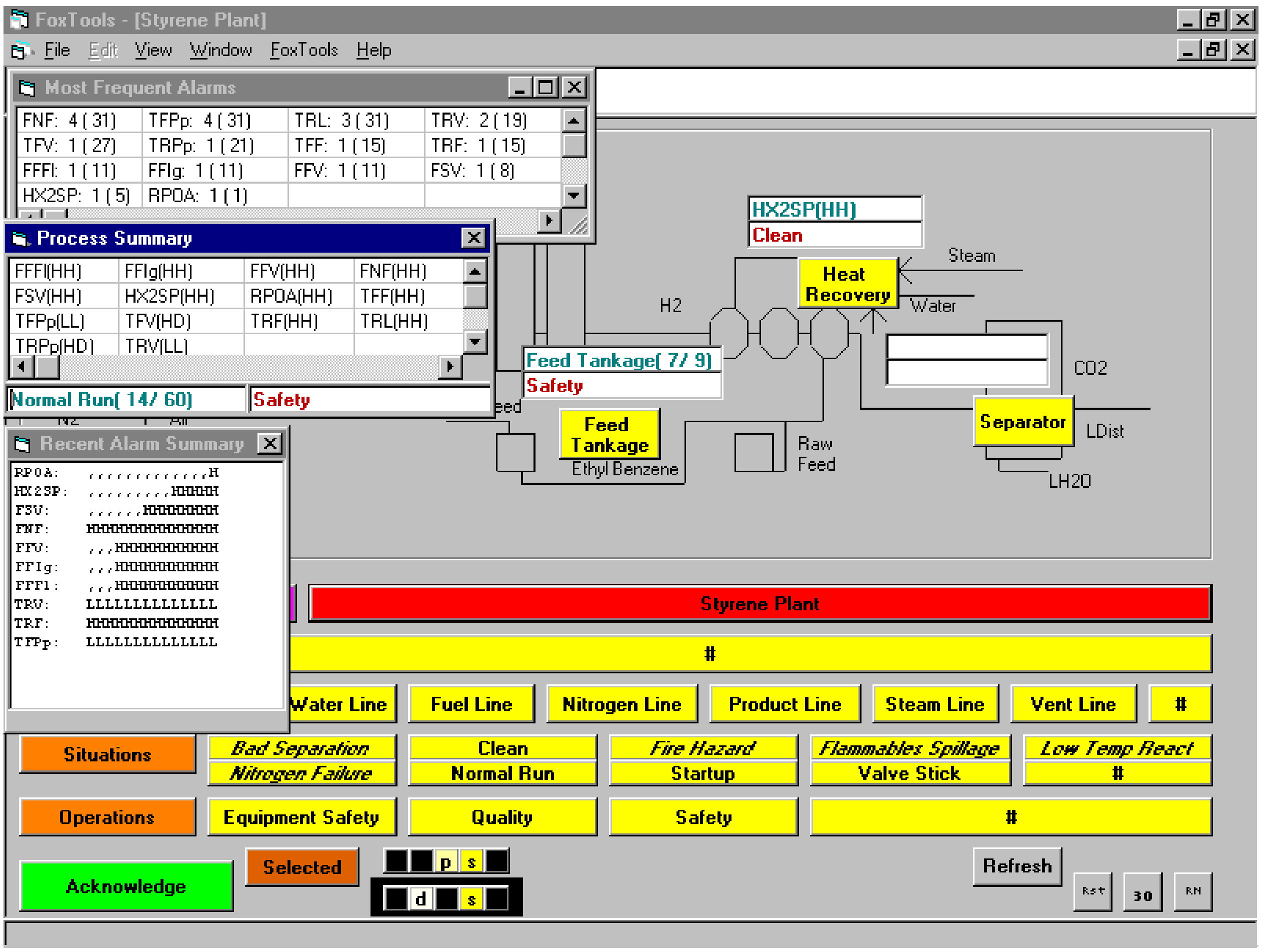
---

[2] The default display is a simple alphabetical listing of the alarms in each Category, called up by the button selection, but a process graphic can be included (as in the case of the (whole) Styrene Plant, Feed Tankage, Furnace, Heat Exchanger, Reactor, or Separator). Each of these graphics has an appropriately labeled Alarm List button, which calls up the corresponding default display.

powerfully, these could be color coded, giving rise to our name for this kind of trending: Battle Ribbon Trending. The color-coding can also trend the acknowledgment history. The display is intended to allow user choice of time window and prediction display as well. The log and trend figures show that the Mask/Filter extends to these displays.

One benefit of a trending (which groups all data about a single variable in a single line) is that its lines can be ordered alphabetically (and include only lines for alarmed variables) for easy search. This automatic ordering can be extended to fit hierarchically grouped tags. The system automatically orders every listing, button array, or other appropriate display, in every figure, consistently in this way.

The demo is based on various standard full-screen Operator displays covering particular classes of information calling for sustained attention. In addition it is handy to have a few simple pull down displays supporting a quick summary capability. The figure shows what we have called Side Logs, displaying different overall alarm views.



From top to bottom three kinds of display are shown:

- A Pareto ordered listing of the Alarms, indicating the frequency of every alarmed variable. Pareto ordering supports a nice alarm discipline encouraging the development of general procedures for addressing or removing frequently occurring alarm conditions.
- An alphabetical listing of all current alarm conditions. This also includes One Word and Priority Summaries (defined in a later section) for the whole plant.

- A trending of the most recently changed alarm conditions.

UNIFORM MASK/FILTER

The Mask/Filter is displayed as a row or column of buttons with their associated Categorization buttons. It allows controlled alarm display with selection of any Category in any Categorization. The alarms displayed in the current (list, graphic, log, or trend) display are those which occur in at least one of the selected/highlighted Categories of each Categorization. Thus the selected alarms occur within the intersection of the different Categorization selections, suppressing all others.

The selection process is designed to make Category selection or deselection easy, whether carried out incrementally or from scratch. The Selected/DeSelected button allows all Categories to be selected or deselected, defining the Selection/DeSelection starting point. Similarly the Categorization buttons allow the Categorization Categories to be all selected or deselected. Collectively the group and individual buttons simplify the Mask/Filter selection.

So far the discussion has addressed Category Selection and DeSelection. The Category selection process can be extended in three more ways:

- Full Selection. This allows the alarms of a selected Category to be fully displayed, without regard to other selections.
- Full DeSelection. This allows the alarms of a deselected Category to be fully masked, without regard to other selections.
- Partial Selection, applicable only to Patterns, as described later.

These modes permit the, always visible, manual or automatic, expression of suppressed or displayed alarms. The two sets of five small buttons, next to the Select/DeSelect button control the choice of available modes. The upper set controls the modes selectable for individual Categories. The lower set controls the modes selectable for the Select/DeSelect and Categorization buttons. The lower set does not allow the change of mode to or from any mode not permitted in the lower buttons. The result is that special conditions, for example an out of service Category, can be imposed without affecting the normal operator use of the Mask/Filter.

ONE WORD AND PRIORITY SUMMARIES

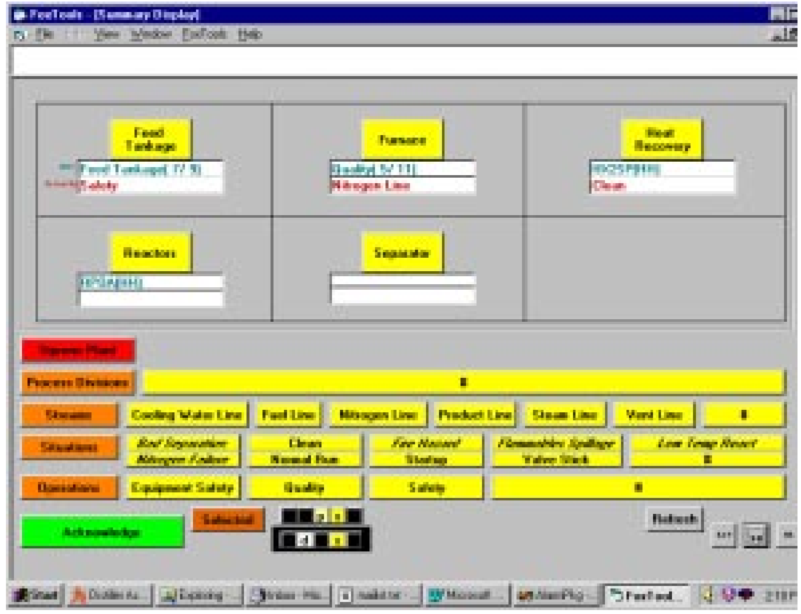
One of the dilemmas of operator display is the meaningful summary of information in alarm indicators representing groups of alarms. The figure below shows a kind of display that might normally be used in an alarm panel, displaying a large number of alarms as labeled alarm lights. In this case, each indicator group summarizes the state of a particular process unit (Category). Each indicator includes button selection of the associated display graphic and One Word and Priority Summary text displays.[3]

A One Word Summary is the name of that other Category whose computed intersection with the group indicator Category contains the fewest listed alarms while containing all of the active alarm conditions of the original Category.[4] It is thus a computed, most-descriptive summary of the Alarm State of the original Category. Where there is only one alarmed variable with active Alarm State in the Category the name of that variable with its state becomes the One Word Summary. In this case, a low level alarm

---

[3] The same information is also displayed in the Styrene Plant plant-wide graphic.

[4] The fraction in parentheses represents the number of active alarmed variables out of the total number of intersection variables.



condition can, when nothing else is happening, percolate up to any higher level display. As more activity takes place, the level of abstraction provided by the One Word Summary goes up, providing the operator with a constant information rate but the best possible description at that level.

One awkward tendency in alarm processing is to configure alarm priorities in neutral categories (e.g. categories 1 through 5). This has the side effect of causing most alarms to be assigned the highest category. A better strategy is to define meaningful, user named Categories of grouped alarms, as described above, and then prioritize the Categories. The Priority Summary is the name of that highest priority Category still containing any of the alarms of the original (e.g. Furnace) Category.

The One Word Summary Category will always fall in a single Categorization. A more powerful use of the underlying concept would make use of intersections between Categories in each Categorization, to further group and localize the alarms of the overall indicator Category. This is not practical in the above displays. However, a special alarm display is provided to allow this form of analysis.

The figure shows three main display elements:

- An alphabetical listing of all alarm variables with their states, similar to that shown later in the default displays.
- A Categorization grouped Category list (similar to the Mask/Filter) that allows the Category membership display of selected alarms.[5]
- The Mask/Filter.

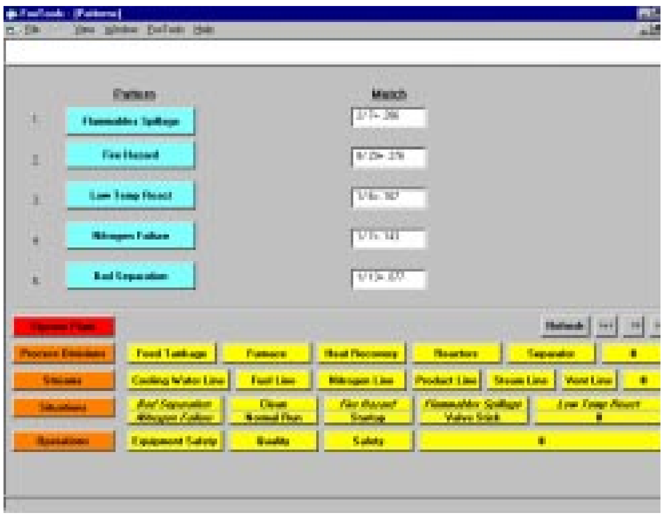
Using the Mask/Filter and Set Buttons, any arrangement of the alarm variables and states can be selected (then highlighted in darker shading). In particular the alarms in any Category can be selected. The Categorization/Category list then indicates (by different shadings), which other Categories (in every Categorization) include all of the selected alarms, and which include only some of them.[6] In this way, the display expands the One Word Summary as an intersection of Categorizations. The display can also be used for a number of other What If explorations.

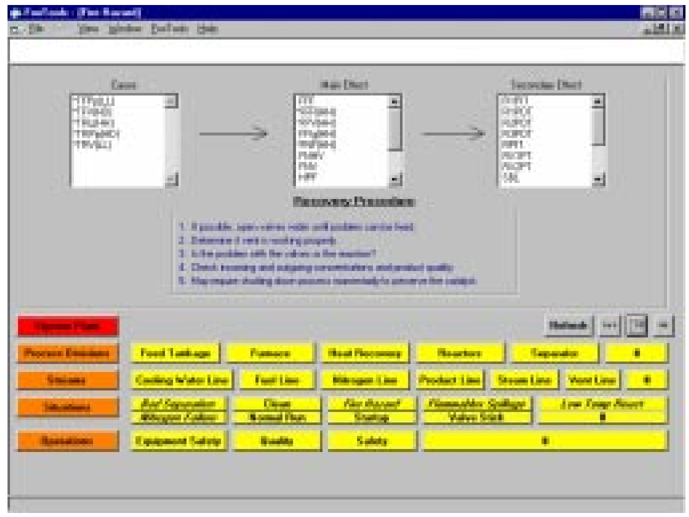
[5] It also shows One Word and Priority Summaries for the selected alarms at the top.

[6] Unlike the One Word Summary display, the included numbers count the total number of alarm states (not just alarmed variables)

PATTERNS

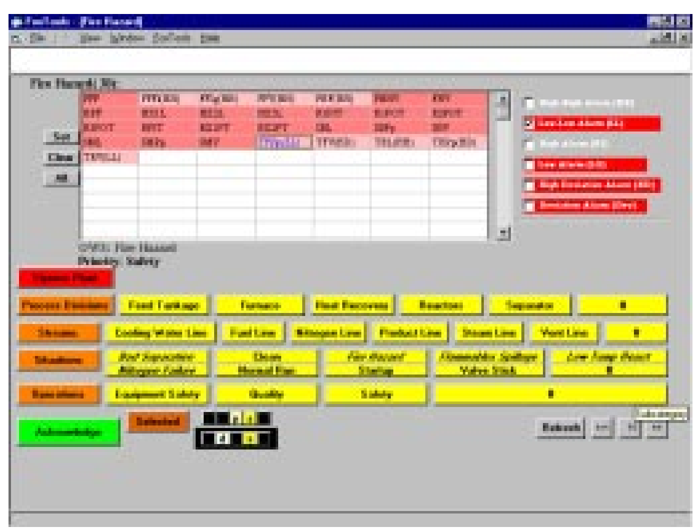
A number of papers, over the years, have proposed showing alarms in terms of recognized sequences of alarm events reflecting causality between the different events.[4,5] Recognition of such a sequence can be used for diagnosis or prediction. The simplest way is to store the sequence and have the system record occurrences of events, which follow that order. The earlier reference recognized that the actual occurring alarms might never follow the sequence directly; moreover there might be several sequences occurring concurrently. Accordingly, a Metric was computed which identified the longest, properly ordered, subsequence occurring within a defined time window. The most likely sequences could then be provided to the operator for his more detailed selection and analysis.





The left-hand figure above shows the resulting ordering with the displayed Pattern Name buttons, number of alarms in the best match sequence, and the normalized (against the total number of Pattern alarms) Metric. The buttons permit selecting a display for a particular Pattern (the right-hand figure), with sequence ordered Alarm variables divided into Cause, and Main (important) and Secondary Effect Alarm variables. Each variable is displayed with its current alarm state. Those alarms, which occurred in the sequence order, as part of the computation of the Metric, are displayed with asterisk. In this form, the display permits operator diagnosis and action on some prescribed recovery procedure.

The Pattern can be used for event prediction as well. In this case the Metric is computed, based only on Cause events. When the number of such Causes exceeds a specified threshold, the remaining Main and Secondary Effect Alarms may then be predicted. The figure below shows the default listed display for the earlier Fire Hazard (as with any other Category). In this case all occurring alarms are displayed with their alarm state. In addition, the remaining alarms are indicated as predicted by the deeper shading. (The right-hand side checkbox display is designed to show the complete Alarm State of any selected alarm variable, with other related information.)



The same shaded prediction is applied to any display including the predicted alarms. The Pattern analysis provides a much simpler substitute for

Expert System application. But it can also be interfaced with an external Expert System, through APIs, to integrate their analyses consistently into the Pattern display. This supports the best of both worlds.

ADAPTIVE ALARMING

Simpler that normal control configuration, these Patterns and Categories can still become too complex for convenient user configuration. There is a need for automatic recognition of usable Alarm situations, Categories, and Patterns. Moreover, the alarm limits themselves can call for situation dependence, making the process of setting limits still more complex than it already is. The Alarm Tools include a number of mechanisms for simplifying all of this:

- Those basic alarms, which do not have inherent limits (as in category 1, above), will be supported by an adapting system, which keeps track of worst case excursions. The engineer can use these to support semi-automatic setting of the limits.
- Categories and Patterns can be supplied with their own set of alarm limits so that whenever the Category or Pattern is selected, the special limits replace the standard ones. In the case of Patterns, the replacement would occur in a timed and sequenced order natural to the Pattern. This allows the Alarm Tools to support situation dependent alarm limits. The Categories and Patterns used this way would be supported by historical records, which allowed the after-the-fact engineer supported adaptation of these limits as above.
- On recognition of a problematic situation the operator could call for a special historian recording action that would record potential Pattern events for later configuration as a Pattern.
- We have defined a special event based auto-correlation strategy, which allowed an engineer to process months of data looking for correlated events constituting potential Patterns.

POLICIES

Currently many vendors support the configuration of alarm characteristics on a variable by variable basis. This is particularly cumbersome for annunciation, acknowledgment, and similar issues, which depend on grouped action for effectiveness. A much better strategy would be to configure such issues for different Categories, as grouped Policies, in the same way that priorities are defined above for the same Categories. The earlier Priority discussion already gives an example of Policy configuration discipline.

BATCH CONTROL ALARMING vs. CONTINUOUS CONTROL ALARMING

The above Tools address alarms in general terms, as related to individual process variables, thorough the simple mechanisms provided by the Categories. They particularly support continuous processes, or the continuous support facilities in Batch processes, by allowing the diagnosis between the many different production activities and failure causality sequences potentially coexisting in time. In the Batch process, each unit or train is naturally operating in a more restricted focus, perhaps in a single phase. Even in the multi-train facility, the alarms of each train correlate with the particular phases running in the train.

In Batch control it is more natural to build an alarm structure clearly integrated with the logic already operating in the control program, and with any normal operating display. These alarms can be seen as special kinds of requests extending the normal operator requests for action. At the same time, the natural program logic lends itself to a higher level of automatic failure reaction, reducing the extent and dependence on operator alarm reaction.

All kinds of processing should be better organizing their collective manual/automated operational support to distinguish different modes or regimes of normal and abnormal operation. These regimes need to be related to clearly distinct kinds of operator display and automation support. The design of Batch control systems should take advantage of their natural characteristics. For example, as relates to the Tools, Batch control already has the Pattern built into the sequencing of the control program. Thus the first part of causal prediction and diagnosis is already inherent in the Batch sequencing. We still need to develop generally recognized regimes of operation, on which we can build standard practices, similar to those developing in the continuous control world.

CONCLUSIONS

Alarms are the main connection between the operator and the automation when addressing operation of the process outside of the normal support of the automation. Such operation, beyond traditional engineered automation, requires tools that allow the operator to effectively call on his working experience. The paper first presents a set of tools, based on the simple configuration of multiple intersecting hierarchies of Categories. These tools allow the operator to control his displays by display selection and information masking. The One-Word Summaries provide summary capability. This permits low level information to percolate up through higher level displays in a controlled level of abstraction with high detail when little is happening and abstraction when more is going on. The Pattern concept supports causal modeling for prediction and diagnosis. Logging and trending access time data. This support exists on top of more organized alarms. Batch control particularly lends itself to these more organized alarms, especially ones indicating exceptional events naturally related to particular phases of the sequencing.

Bibliography

[1] E.H. Bristol, "A Language for Integrated Process Control Application", Retirement Symposium in Honor of Prof. Ted. Williams, Purdue University, West Lafayette, IN, Dec. 5 - 6, ‘94.

[2] E.H. Bristol, "Not a Batch Language; A Control Language", World Batch Forum, San Francisco, May ‘95; also ISA Transactions, Dec. ‘95.

[3] E.H. Bristol, "Redesigned State Logic for an Easier to Use Control Language", World Batch Forum, Toronto, May 13-15, ‘95.

[4] E.H. Bristol and H.L. Wade, "Alarm Analysis Can Diagnose System Faults", Control Engineering, Feb ’76, pp. 47-49.

[5]K.-E. Arzen, "A Graphcet based approach to alarm filtering", Proc. of the IFAC World Congress, San Francisco, `96.