Abstract

Operators with supervisory control tasks in large systems, such as chemical process plants or electricity generation plants, are usually supported in their work by an annunciator system. This annunciator system alerts the operator when measured variables exceed their desired limits or when certain discrete events are detected.

The functioning of such annunciator systems is discussed in this paper. The basic principle of conventional annunciator systems is that the system's state, as determined from the measured data, is compared to a reference state. Unacceptable deviations are signalled as alerts. However this principle of state-based annunciation does not provide support for the operator's task of achieving the system goals.

Functional modelling offers the tools to build a new generation of annunciator systems that can "understand" mode changes in a system, and check and report the achievement of system goals instead of conformance to a reference state. The design of such annunciator system leads to less but that are more meaningful alerts. Filtering of alerts, as is done in many annunciator system, becomes unnecessary.

Introduction

Supervision of automatic systems, or supervision combined with control actions, has obtained an important role in many work places. Automated equipment controls and checks a large number of measured variables, called process points. The human operator in these system exercises supervisory control; he or she monitors the process and the automation, adjusts the process setpoints if necessary and does the fault management. Such work situations are no longer only limited to process technology. A few areas in which this supervisory role for the human has invaded are (van Paassen & Wieringa 1997a):

- Power plants and power distribution plants.
- Chemical and technological processes. Many of these processes are run in batches, others run semi-continuously.
- Transport systems, such as aircraft, trains and ships, and their environments, such as Air Traffic Control.
- Patient monitoring systems in hospitals.
- Many industrial manufacturing plants are already automated, and are operated under supervisory control.

Annunciator systems are needed in these cases because the number of variables measured by the data acquisition and control system is far too large to be all checked by the human supervisor, and in most cases only a few of these variables are of interest at any moment. Moreover, the majority of the variables is controlled by
automatic controllers, or only intended as a check on the process. Only when these variables exceed certain limits, the supervisor should take notice of these, determine what the cause of the abnormality is and initiate corrective actions.

The task of the annunciator system is to direct the operator's attention to those parts of the controlled system where operator intervention is needed. Often annunciator systems generate alerts while there is no need for an action of the operator. This occurs frequently during transitions in operating mode. Such alerts are called false alerts. In the following we will discuss why not these alerts, but the method of alert generation is false.

**Conventional Annunciators**

When a system is in a stationary state, detection of anomalies is simple; any change in the process variables can indicate a possible anomaly. This is the basic principle on which conventional annunciator systems are based. Very crudely said, a reference state of the system is defined, and the values that the process variables will have in this state are determined. Then the acceptable deviations from these values are determined, and used to calculate the alert limits.

Once the system has been brought to its reference state, conventional annunciator systems seem to work well for detecting anomalies. However, in a start up or shut down, or during another mode change, the system is usually not in such a reference state. Often many alerts are generated in these cases.

Most annunciator systems function as described above (Paassen & Wieringa 1997), and the operators are used to the current work situation, and even adapt their behaviour to peculiarities in the annunciator system (Kragt & Bonten 1983). When problems occur, such as a flood of too many alerts in certain conditions, or repetitive nuisance alerts, engineers try to remedy the situation by improving the annunciator system. Clusters of alerts that occur in known cases are suppressed and replaced by one alert. The approach taken here is that the system's behaviour is analysed for known and foreseen failure modes (what if the pressure sensor breaks down, etc.), and the alert logic is adjusted so that the most probable cause of a problem is determined and given as the main alert. Another remedy that is often used is the mode-dependent suppression of alerts.

This looks like an acceptable solution, but even when this is done, support from an annunciator system stays sub-optimal. To clarify this, an almost trivially simple example system, given in Figure 1, will be used.
Figure 1: Example system; a vessel with an inflow, outflow and a heater.

The example system contains a tank which can be filled with liquid, and heated with a heater. One can configure a "conventional" low level alert on the tank. When this alert sounds, it tells the operator that the level in the tank is low. A good operator will check the inflow and outflow of the tank, judge whether it was to be expected that the level in the tank would drop, and if this is the case the operator adjusts in- or outflow, or both. The tank may not be heated when empty. Therefore, he will also check whether the heater should be switched off, to prevent damage to the installation.

The crucial point is that when an alert configured in this way, as a check on a measured value, it only tells something about a part of the state of the system. Since the system's goal, and thus the desired state of the system can change over time, the operator has to evaluate the alert in the context of the system's goals. Specifically, an alert:

- Does not convey any information about which of the system's goals are threatened.
- Does not convey any information about which actions are needed to correct a potential problem.
- Occurs only on the basis of the system's state, irrespective of the goals that the operator might have.

In other words, an alert is given, but the operator has to interpret the alert in the context of the state of the system, of his previous actions and of his goals, to determine what the alert really means.

In some cases an alert does indicate which action is required in a certain condition. In the aerospace domain this is often used. Ground Proximity Warning Systems (GPWS) are an example of this, the warning is usually aural warning, such as "too low, pull up, pull up".

The alert given by the GPWS should be compared to a complex alert in an industrial process. Compound logic is used to generate the alert; the configuration of the aircraft is considered. Data about the state of the landing gear (extended or retracted), flap setting and engine power setting are combined with data from the radio altimeter to decide whether an alert has to be given. This prevents a warnings on the approach
to landing. However, false alerts are not rare, in fact, during approaches in mountainous terrain false alerts are fairly common.

What has been done in this case is that the alert logic has been extended to make the alert more compatible with the pilot's intentions. To achieve this a simple guess for the pilot's goal has been made, on the basis of the aircraft's configuration. Then it is checked whether the measured data are compatible with the pilot's goal.

An alternative view

There is a discrepancy between the services offered by an annunciator system, and the task of the operator or pilot. The operator must manage the controlled system so that as many of the systems goals are achieved as possible. For example the airline pilot must try to accomplish a safe, comfortable, economic and timely flight to a certain destination. When it is not possible to achieve all those goals at once, the pilot must make a selection between those goals.

For example when, due to weather conditions it is not possible to safely land at the destination airport, the pilot will make a decision to postpone the landing or go to an alternate destination. The goals of comfort, economy, timeliness and destination are sacrificed for the safety goal.

Determining the goal priority and deciding which goals to pursue and which goals to ignore is one of the most difficult tasks for an operator. The annunciator systems currently in use do not, or at least not explicitly, consider this problem of goal selection and goal achievement.

Annunciator systems help in determining the state of the controlled system, but further analysis and decisions are not supported by the system. Also the other sources of information that are at the operator's disposal, such as the mimic diagram, only give information related to the state of the system. Attempts to present more goal-directed information are often only found in laboratories (Vicente & Rasmussen 1990, Sakuma, Itoh, Yoshikawa & Monta 1995).

In fact, annunciator systems provide information about the systems goals only when the system is in its reference state; when there are no alerts, all goals should be achieved. This is however a one-way causal relationship, when there are alerts, this does not say that there are goals that have not been achieved.

If the state is not close enough to the reference state of the system, so that alerts are given, then it may still be that the goals are achieved. For example the level in the barrel of Figure 1 may high, and the inflow has been shut by the operator to get the level to a normal value. This inflow might deviate too much from the inflow in the reference state, and an alert is generated.

Such an alert is called a "false" alert, but why? An annunciator system correctly indicates that the state of the system is not in accordance with standard state of the process mode. The only thing that makes an alert a "false" alert is the fact that it does not correspond to the operator's goals or to the system's goals. The alert may be irrelevant, because it signals a deviation of the standard state which cannot be corrected at the time, or because that deviation is exactly what the operator intends.
The problem is however, that people tend to rely on annunciator systems for support in their goal-directed behaviour.

If we want to improve annunciator systems, we have to find a way to eliminate “false alerts” and improve the content of the annunciator system's information so that the operator is better aided by the system's output. The solution is that one has to evaluate the goals of a system directly, and generate alerts when these goals are not achieved, rather than generate alerts on the basis of the state of the system.

**Evaluating goal achievement**

**Direct and Indirect goal achievement**

How can the achievement of a goal be evaluated? Sometimes this is simple. Take the example of Figure 1. If the goals of this sub-installation are the delivery of a certain quantity of liquid (goal G1) at a certain temperature (goal G2), then the achievement of these goals can be checked by measuring the outflow of liquid and the temperature of this liquid and comparing those to the desired values. So far there is no difference between a goal-directed approach and a conventional annunciator system.

But what about the other signals we measure from the process, how should they contribute to the alerts? The inflow to the barrel might be low because the pump is broken down, because the level in the barrel has to go down, or because the operator shut the inflow when the level was too high, and simply forgot to open the valve again. Simply setting a limit on the inflow, as might be done for a conventional annunciator system, cannot provide a useful alert about achievement of the system's goal.

It might be clear that the inflow does not serve a goal of its own (such as: there “must” be an inflow of 5 kg/s). Neither does it directly contribute to the outflow goal. It is therefore pointless to define fixed alert limits for the measured inflow.

The inflow of liquid into the barrel is a function, in the definition of the word such as it is used in the discipline of Functional Modelling (Lind 1990, Modarres 1996). A function is the useful behaviour or capability for useful behaviour of a system. Functions are related to activities in the system, and thus to the process occurring in the system, whereas components or sub-systems, the traditional “units” of systems engineering, describe physical properties of a system.

When the operator tries to achieve goals G1 and G2 with the system, the goal of the inflow function is to support yet another function. This other function is the storage of liquid in the barrel (in functional modelling, storage is considered to be function, as well as for example transport or mixing). A certain quantity of liquid has to be available so that the third function, outflow of liquid, can be realised. This last function of the system achieves one of the main goals. We can summarise this in figure 2.

Now it is clear what must be done with the measured inflow. This measured inflow characterises the inflow function, which in turn has a goal. This goal was not chosen by the operator or his boss, but by the designer of the system. Thus the measured inflow can be used to check the achievement of this goal. The criteria to check the
Figure 2: A chain of functions and goals necessary to achieve the goal G1.

achievement of this goal are not always the same, these depend on the state of the storage function.

If the level of the barrel is low, the inflow should be higher than the outflow, and when the level of the barrel is high, the inflow should be lower than the outflow. The inflow function represents the capability to change the level in the barrel.

But achievement of the inflow goal is not equal to achievement of the level goal. When the barrel is empty and the inflow is high, the level goal is not achieved. However we must acknowledge that the inflow is correct for this situation. One has achieved the inflow goal, because the storage function is supported by the high inflow. One can say that the goal of the inflow function is achieved when the goal of the function that it supports, the storage function, will be achieved in the future.

This discussion illustrates that, if we want to provide more useful -- goal oriented -- alerts to the operator, we will have to identify the goals in the system and the functions which the system offers to achieve those goals. Some functions will be in a position to achieve the main goals, other functions are necessary as support. The goals of those functions that are not directly involved in achievement of the main goals have to be identified, and their achievement must be checked. Checking the achievement of these goals can be complex, because the criteria for achievement (the demand on the function) can vary.

Goal selection

The discussion in the previous section ignored one question, namely whether the achievement of the goals G1 and G2 was relevant. Suppose that the liquid flowing out of the barrel goes to a long pipeline, where it would coagulate and form deposits when the temperature becomes too high. Now in order to protect the pipeline, the outflow has to be diverted to a waste drain when the maximum acceptable temperature is exceeded.

How should one express this situation in for example a computer representation of goals and functions? One possibility would be to attach all kinds of conditions to a goal, and consider -- and report to the operator -- its achievement or non-achievement only when these conditions are true. However, there appears to be a simpler principle. In the case of the possible clogging of the pipeline, there is in fact another goal to
consider; “the pipeline should not get clogged”, or more precisely, “the pipeline should not be in danger of getting clogged”. This goal is achieved when liquid that is too hot is not fed into the pipeline.

When the temperature of the liquid is too hot, this goal, lets call it G3 for convenience, is in conflict with G1. If one decides that G3 is more important than G1, then the achievement of G1 is only relevant when this achievement is not in conflict with the achievement of G3. In general, a goal may be considered for achievement only when the functions needed to achieve this goal are not “claimed” by another goal with higher priority.

A more complete set of rules for evaluation of goal achievement and function state is given in (van Paassen & Wieringa 1997b). These rules are given for a system formally described in a Multilevel Flow Model (Lind 1990), and also encompass the description of a transition from one process mode to another. More details on the evaluation of flow function state, and an elaboration of goal concerning the integrity and safety of a system is given in (van Paassen & Wieringa 1997c).

**Goal-based Annunciators**

In functioning the combined human-machine system, the contribution of the annunciator system is one of alerting. We have seen that conventional annunciator systems basically alert for deviations of a reference system state. In an off-normal situation, often multiple measured values exceed their fixed limits, leading to multiple alerts.

An annunciator system based on functional modelling is based on a different principle. The alerts indicate non-achievement of the system's goals; they are not directly linked to the system's state. Apart from the goal achievement, the relevance of the goal at the current point in time can also be derived from the functional model. The alerts that are presented to the operator can thus be limited to those alerts related to goals that are relevant. This means that the following will occur:

Multiple goals may be not achieved, or in danger of being not achieved.

Of those goals, only the ones that are relevant, i.e. the goals that are not in conflict with other goals with a higher priority, are presented as an alert.

In this case, since an alert is always for a goal that is relevant, each alert can be worked on immediately. There are no impediments for not using the system, in fact for not using the available functions, for the achievement of the goals for which an alert is given. For example in critical situations, when the safety goals of the system are threatened, only alerts for these safety goals will be given. Using the functional model, the annunciator system can deduce that the production goals are irrelevant, and not generate alerts for these goals.

An alert that is generated in such a manner can be more than a statement about the system's measured state. Taking the example of clogging of a pipe due to coagulation, the alert can comprise the following information:
- An identification of the goal whose achievement is being endangered. For example: Danger of pipe clogging.
- An explanation of why this is so: Temperature of the liquid is too high and liquid is flowing into the pipe.
- An indication of the required action: Shut down the flow into the pipe. Lower the liquid temperature.
- An indication of the goal conflicts, and as such information about possible side effects of the action: The outflow goal is superseded.

Of course, an experienced operator will know all the major goals and most sub-goals that have to be achieved with the system, and he might want to check the status of specific goals. If a goal is irrelevant, because it is conflicting with a higher-priority goal, this should be reported when information about the goal is requested.

Compared to the conventional, state-based, approach to alert generation, alerts based on a functional model are always relevant, so there is no need for filtering. Moreover the alerts can contain specific information about the required actions to remedy a situation, and on possible consequences for lower-priority goals.

**Implementation issues**

Annunciator system that are based on evaluation of system goals, instead of system state, as outlined above, do not yet exist. A set of basic rules which is required for reasoning about goal achievement and relevance has been developed (van Paassen & Wieringa 1997b, van Paassen & Wieringa 1997c). However these rules, although already specific, have still to be processed into a computer implementation.

A reasoning mechanism alone is not enough. Each annunciator system will need a functional model of the controlled system. An issue that should not be overlooked is the knowledge acquisition needed for the development of such a model. The engineers that have to implement an annunciator system will often not be familiar with a functional modelling formalism such as MFM. They should be aided in specifying the system, for example by a program that can construct the model together with engineers and experts, in an interactive session.

The third issue that has to be addressed is the presentation of alert information to the operator. The information available in the functional model is "richer" than the information given by a conventional annunciator system. It must be investigated how this information must be presented.

**Discussion**

Conventional annunciator systems have shortcomings. In some installations large alert floods are generated, and to improve these systems a considerable engineering effort is needed.

This paper proposes an alternative approach to annunciator systems, which uses goal achievement, and not the compliance to a reference state as its basic principle. This
could lead to annunciator systems that generate less alerts, and the generated alerts would be more compatible with goal-oriented behaviour from the operator.

The development of such an alternative annunciator system is not trivial. The functional modelling theory that is applied is fairly new, and it cannot be expected that the engineers who implement annunciator systems can easily specify the functional models required for the companion. Deployment of this technology will require training and the development of proper support tools.

Acknowledgements

The research into alternatives for annunciator systems has been sponsored by RoHMI (RObust Human Machine Interaction, a European Community Human Capital and Mobility project) and the ABB Research Center in Heidelberg.

References


