Multiple alarms, Major Goals and Implementation.

B. Lublinsky.

Fermi National Accelerator Laboratory*

1. Introduction

It is a very common situation that controlled hardware can work in a variety of different regimes, in which case the control system has to be smart enough to recognize a regime change and reconfigure its alarm system accordingly. For example, when a magnet quench occurs, the refrigerators warm up and the cryogenic control system will generate many alarms. Only a few of them are important and they may be lost in the flood of generated alarms. If the alarm system is smart enough to increase permissible maxima for temperatures and pressures, we will see only what is important. Multiple alarms support usually involves three major parts:

- regime determination support;
- regime switching support;
- multiple alarms support.

This paper discuss implementation of multiple alarms support for the Fermilab cryogenic control system (FRIGes) and the ways of possible utilization of the described approach in the rest of our overall control system, long known as ACNET.

2. The overall structure of multiple alarms support.

The regime determination is the less formalized part of multiple alarms support and can range from a brute force setting to a very precise calculation using the system's parameters. The current implementation allows for a wide range of possible approaches by using the FRIG finite state machine mechanism[1]. This mechanism is already built into the cryogenic control software, which is flexible enough that it is relatively easy to implement an additional finite state machine (FSM) which is dedicated solely to tracking of the cryogenic system parameters and determining the current regime.

The regime switching mechanism is significantly more generic and is implemented in the form of a Virtual Machine Front End (VMFE) that can be used not only by the cryogenic control system but throughout the whole of ACNET. The overall interaction scheme for multiple alarms support is shown in figure 1.

Multiple alarm support itself is included in the FRIG microprocessors internally in the form of keeping several alarm blocks for every particular device having this tracking mechanism and also of allowing alarm block switching based on regime changes.

3. Regime determination and broadcasting.

Several changes have been made to support multiple alarms in the FRIG micro. Firstly, two new types of devices were introduced: a virtual machine control device and a virtual machine reflection device (see figure 1). A virtual machine control device in turn consists of two sub-devices - primary and secondary. The secondary device is used for establishing the link to the VMFE. It supports three database properties: setting (setting of the VMFE device to which it will be connected), reading of setting (reading of the set information back) and basic status (acknowledgment that the link to the VMFE has been established).

The primary device is used for setting of the regime itself. The regime can be set either externally via the network from an application or internally by FRIG software, for example by the dedicated FSM. When the primary device receives a setting it sends a corresponding message (via a proprietary protocol) to the VMFE. Its presence facilitates internal intelligence in FRIG micro, which makes the decisions about regime, or state, changes. The database properties supported for the primary device are setting (setting of the current regime), reading and reading the setting (both reading the current regime) and basic status (as for the

---

* Operated by Universities Research Association under contract with the U.S. Department of Energy

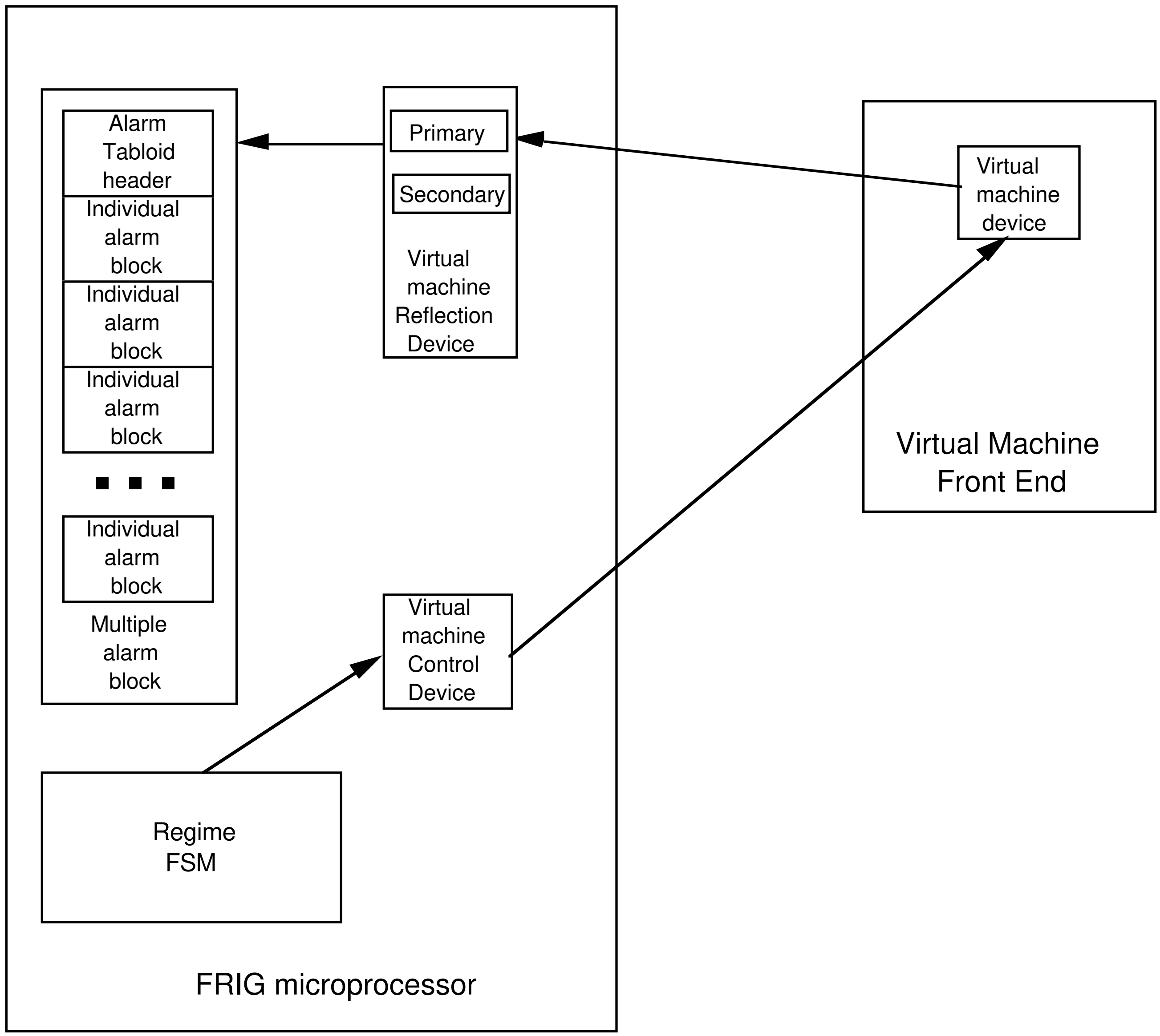

Fig. 1. Overall organization of the multiple alarms support

secondary device). The VMFE, when it receives a regime change message, checks it for correctness and, if it is in fact correct, does the setting to the virtual machine reflection device, which is what actually controls alarm block changing. In the current implementation we are supporting up to 64 virtual machine control devices and up to 64 virtual machine reflection devices. This framework is flexible enough to provide both internal and external regime switching and internode communications.

4. Virtual machine FE.

VMFE is an Open Access Front-end (an ACNET term referring to a node below the consoles in the standard diagram, but where user application code can be inserted) that services requests for state changes of virtual machines. Introduction of this FE might seem as an over-complication of the multiple alarms scheme, but in fact it allows one to solve two very important problems:

- centralization of control of possible states and state transitions for all virtual machines in the control system;
- coordination of regime change between multiple microprocessors.

The first goal is achieved by holding the descriptions of the virtual machines in ACNET devices, owned by the VMFE (usually there is one-to-one correspondence between these devices and virtual machine control devices). Each virtual machine description allows the user to define the set of valid states for this machine and to allow/prohibit state transitions. This significantly simplifies the day-to-day operation of maintaining the current virtual machine definitions.

The second goal is achieved by linking multiple virtual machine reflection devices to one virtual machine device in the VMFE.

5. Microprocessor support for multiple alarms.

Multiple alarm support is achieved by replacement of the usual alarm blocks by alarm tabloids, each containing 16 alarm blocks appropriate for normal devices. Every tabloid has a pointer to the virtual machine reflection device by which it is controlled. This hierarchical scheme is established in the database, by introducing an additional property - PRVMDI (standing for PRoperty Virtual Machine Device Index). Given a VMFE device, it is possible to query it to determine the virtual machine reflection device for the given node, that actually fulfills local regime control. This hierarchy can be downloaded into the microprocessor and queried from a console application, both via a specialized protocol. Each time an alarm scan is performed the value of the virtual machine reflection device is obtained and this value serves as an index into the array of alarm blocks for this particular tabloid. The virtual machine reflection device itself, in this implementation, doesn't have to know about the tabloids it is controlling. This allows one to have connection pointers only in the tabloids themselves and eliminates the necessity of maintaining a linked list of tabloids belonging to each particular virtual machine reflection. The overall regime tracking mechanism is shown in figure 2.

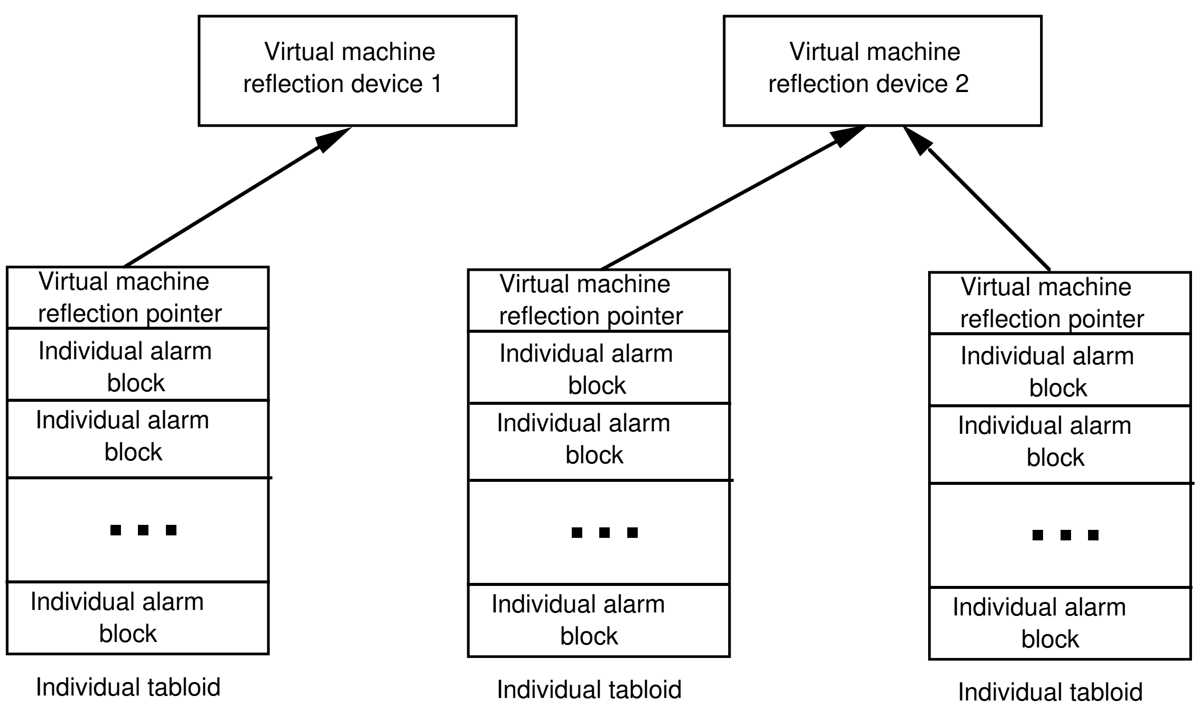

Fig 2. The Tracking mechanism for regime change in alarms support

There are several caveats here. In order not to lose information when a state change occurs, tabloids contain, besides the alarm blocks themselves, the current state value, current GOOD/BAD status, and current "message send to alarm process" flag, thus allowing smooth state transition. Whenever transition occurs the normal alarm processing procedure is followed, beginning with initialization. In the abnormal situations when the virtual machine reflection device requests a state for which the alarm block doesn't exist, the microprocessor will produce an alarm of its own indicating this fact. There is a limited number of such alarms which can be produced, in order to prevent an 'alarm storm' under unusual circumstances.

current alarm block	first alarm block	second alarm block	third alarm block	▪ ▪ ▪

Fig. 3 The external view of the alarm blocks.

The way in which an application sees a multiple alarm block is shown in figure 3. Here "current" stands for the alarm block to which the virtual machine reflection device is pointing. Setting to and reading from this block is possible, independent of what this "current" state is. Similarly the various "physical" blocks may also be referenced.

6. Database support for multiple alarms.

Work had to be done for the ACNET database to support multiple alarms:
- support for the alarm array was implemented;
- the additional property PRVMDI was added.

Support for the alarm array assumes that instead of one alarm block with a particular length, multiple alarm blocks can be associated with every alarm property (analog/digital alarms). The database supports the same "view" of the alarm block tabloid as an external application except that alarm block 0, the "current" block, is not defined in the database.

PRVMDI is that new property introduced especially for multiple alarm support. It is generalized and thus could be used to support 'multiples' of any property. Specifically it consists of two fields; the first of these is the index of the VMFE device, that is making the actual state transition. Using this index an application can query the VMFE about the virtual machine reflection device of interest. The second field is a set of bits corresponding to all possible properties, each of which indicates whether that property will be affected by a VM device transition.

7. Conclusion.

Multiple alarm support is fully operational and has been used in the cryogenic control system at Fermilab for several months. It has proved to be very useful operationally, especially during warm-up and cool-down regimes for the Tevatron. Extension to other ACNET subsystems is under active consideration.

8. Acknowledgments.

Many people participated in design, implementation and testing of multiple alarms. Bob Joshel designed and implemented the Virtual Machine FE. Brian Hendricks, Tim Zingelman, Bob Joshel and Glenn Johnson implemented database support. Alex Martinez of the Cryo Group implemented cryogenic-specific FSM support and thoroughly debugged the whole system.

References

1. B. Lublinsky, J. Firebaugh and J. Smolucha, New Tevatron Cryogenic Control System, "Proceedings of the 1993 Particle Accelerator Conference", V3, p1817.