Created:August 1, 1997
Modified:January 13, 1998

Gemini
Controls
Group
Report

Health and Alarms

Steve Wampler, Kim Gillies, and Bret Goodrich

TN-C-G0053/01

This report provides a short clarification on the role of Alarms and system Health in the Gemini Control System.

1.0 Introduction

Recently, some questions have been raised on the differences between the Gemini system 'Health' and the alarm system. This short note attempts to clarify through example the roles of the two, and then presents methods that can be used to implement Health.

2.0 Alarms

An 'alarm' is a signal raised by a specific mechanism when it is not functioning correctly. Alarms signals may be 'MINOR' or 'MAJOR'. There are two ways alarms may be raised:

1. Through EPICS record processing, as when a process variable exceeds a limit (e.g., HI, LO, HIHI, LOLO).
2. Through tests done in C code (device or record support, separate VxWorks tasks, etc.).

A software system may have many mechanisms with many attributes that can raise alarms. Method (2) may cause an EPICS alarm invoking method (1).

3.0 Health

System 'Health' is a status item (SIR record) that describes a system's ability to meet its specification while observing. There are four possible states:

- GOOD -- the system is performing at or better than specification.

- WARNING -- the system is working, but not to specification. Image quality may be degraded as a result.
- BAD -- the system is incapable of observing
- INVALID -- the system's health is unknown (cannot be determined)

Each software system has a single Health status SIR record.

4.0 Comparing Health and Alarms

Note that nothing in either definition directly ties Health and the alarm system. If Health is not GOOD, then the cause must be traceable through that system's engineering screens. (See CICS for an example showing how this can be done.) If an alarm is raised, it may or may not have an effect on Health. It is entirely possible to have a system that has alarms raised, but is still in good health.

For example, consider the M1 control system. The PCS has 154 actuators controlling the shape of the primary mirror. If any one of them is not working correctly (out of limit, frozen, etc.) that should certainly raise an alarm. However, it *may* very well be the case that Health is completely unaffected by having a single actuator working incorrectly, as the mirror figure may not be significantly affected by a single actuators failure. It may also be that the Health is set to WARNING if the figure is slightly degraded, or to BAD if the figure is substantially degraded.

It is also possible that a simple system may be able to determine its health entirely in terms of its records that can produce alarms. This would be the case for a controller that does no self diagnostics, contains no algorithms that use input parameters to produce output parameters, or contains only EPICS records and no C code.

The Gemini system design places the following distinction between Health and alarms: it should be possible for the SSA (System Support Associate, who runs the telescope during observing) to determine the ability of the system to perform observations by looking *only* at the states of the health records for the various subsystems. Alarms are viewed as *entirely* engineering diagnostic tools. *If* the Health of a subsystem is not GOOD, then the SSA may opt to examine the alarm status and certainly alarms should be logged so engineers can track/repair problems. The SSA may examine the condition of a subsystem through the engineering consoles for that subsystem, which should show any outstanding alarms or provide some other indication of the exact problem.

Note that Health is necessarily vague - we expect to refine the computation of Health as we learn more about the relationships between mechanisms and observing performance. However, we are relying upon the developers of systems to provide us with a first approximation. The system developers have the most knowledge about their own systems.

5.0 Implementations

Now, how can Health be implemented? For most systems, there would be some relationship between alarms and system Health (though this relationship may be involved - some alarms may directly alter Health, others might only in some circumstances or conditions, and some may not affect Health at all).

One method is the propagation of alarms into an EPICS genSub record or hierarchical set of records. This method allows alarms to be merged, and a subroutine may be implemented to determine if alarms may change the overall health.

Another method is the EPICS alarm handler (alh). This extension builds alarms into a hierarchical tree. Final or intermediate results may be returned to an EPICS PV such as Health. Two problems this method has are (a) there is no differentiation between Health and alarms, which may not be appropriate for your system, and (b) it requires the use of a Unix system for operation.

Two methods which may be considered in the future are (a) porting the alarm handler into VxWorks and (b) creating a specific alarm record to handle system wide alarms.