# Alert/Notification Requirements

Tom Vermeulen

29 April 2004

This document is available on the Web at: http://software.cfht.hawaii.edu/statserv/notification

# Contents

# 1 Introduction

## 1.1 Purpose

As we move forward on identifying the requirements for WIRCam, one key requirement is the ability to monitor subsystem parameters and send out alerts based upon predefined error or warning conditions. With MegaPrime, these alerts were hard-coded in the fipcom agent software. This has worked pretty well with MegaPrime, but it has several drawbacks. First, the rules are hard-coded in the C code of the agent. This makes it more difficult to perform iterative fine-tuning of the rules. If a rule needs to be changed, someone in the software group needs to be informed and change the agent. Meanwhile, everyone using MegaPrime needs to hope that there were no side-affects to the agent introduced by the change. At this point, there are several rules which have been determined to be less than optimal, but a determination has been made that some of the changes may not be worth the risk of changing the fipcom agent.

Another issue with the MegaPrime model is that the fipcom agent must be running in order for the alerts to be available. Because all of the current MegaPrime alerts are based on fipcom data this may not be a major issue. However, plant and instrument control data is now being added to the Status Server by sources which are not director-based agents. The best example of this would be the Allen-Bradley PLC status and control information. As a result, it appears that much of the critical state and status information, upon which notification rules would be based, will be available in the Status Server. However, there isn't the dependency that an agent running under director must execute the notification rules.

At this point, I don't believe that the current mechanism used with MegaPrime is the best choice for WIRCam instrument, or plant control related notifications. The purpose of this document is to list a first set of requirements for a notification system based on information contained in the Status Server. These requirements were defined in a meeting held with Will Rambold, Rohendra Atapattu, Tom Benedict, Grant Matsushige and myself. At this point, there may be more requirements, which still need to be identified. The goal of this document is to highlight the requirements which have already been defined and, through the distribution and review of the document, perhaps identify additional requirements.

## 1.2 Scope

Unless otherwise noted, the requirements identified in this document are intended to be implemented in the first release of a notification system. However, release requirements may dictate the priority and staging of functionality.

# 2 Notification Triggering Requirements

This section will discuss the requirements for the rules which will define whether a notification will be sent.

## 2.1 Data used within the triggering rules shall be available in the Status Server

The rules used to define the notifications will be based on data which is currently in the Status Server. The notification system will not be required to source data outside of the Status Server. If external data is considered necessary to be part of a notification rule, it must first be pushed to the Status Server. For example, if Allen-Bradley PLC information is used within a notification rule, it must first be made available to the Status Server. The notification system will not communicate directly with the PLC device.

## 2.2 A simple and well described syntax must be available to define notification rules

One of the goals of the notification system is to make it easy for individuals, who wish to receive notifications, to define and adjust the rules which trigger the notifications themselves.

A likely mechanism for defining the rules is a C-style regular expression syntax. For example, here is a simple rule which might trigger if there is a potential problem with the vacuum system for MegaPrime.

```
((/i/megacam/fipvar/cryoErrors/vacuumCritical == ‘‘TRUE’’) ||
 (/i/megacam/fipvar/cryoErrors/vacuumOutOfRange == ‘‘TRUE’’) ||
 (/i/megacam/fipvar/cryoErrors/vacuumTooHigh == ‘‘TRUE’’) ||
 (/i/megacam/fipvar/cryoErrors/vacuumTooLow == ‘‘TRUE’’));
```

I would anticipate that many triggering rules will be more complex than this example.

## 2.3 It must be possible to specify a time-based rule conditions

It must be possible to define rules which will be triggered only after a parameter or set of parameters is at the same value for a given duration of time. For example, there are cases where a subsystem will be turned on and some settling will occur before a parameter becomes stable. Hopefully this can be illustrated more clearly with a picture. Figure 1 shows the case where some time may be required for a parameter to settle within its operating threshold. In this example, the notification rule could be set up to only trigger after the on switch has been in the “ON” state for at least 15 minutes.

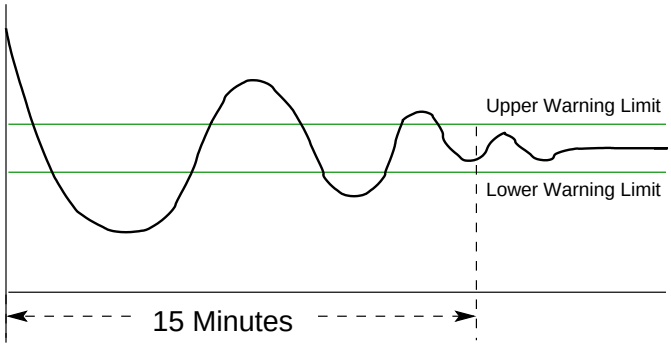


Figure 1: Parameter Settling Example

## 2.4 It must be possible to trigger rules on both valid and invalid data

The data in the Status Server is usually valid if the supplying client is actively providing information. However, if the client were to go away, or if the data goes stale (lifetime exceeded), the data will be considered “EXPIRED”. An example of this is the datalogger. The datalogger updates the Status Server every 10 seconds. If a new set of updates is not received within 20 seconds, the data will go expired. It must be possible to specify rules which trigger on data changing between a valid and invalid state and vice-versa.

## 2.5 It must be possible to suppress repeat notifications for a period of time

If a rule is triggered, it must be possible to suppress a repeat notification based on the same rule for a predefined period of time. For example, if a notification is generate on the following rule.

```
(/i/megacam/fipvar/cryoErrors/vacuumOutOfRange == ‘‘TRUE’’);
```

It may be possible that the vacuum system is right on the threshold of going out of range. If the “vacuumOutOfRange” variable begins to oscillate between “TRUE” and “FALSE”, it must be possible to suppress multiple notifications each time the variable changes to a “TRUE” for a predefined period of time.

# 3 Notification type and message content

## 3.1 It must be possible to send email notifications

At this point, the only required mechanism for notifications has been identified to be email. Many of the current CFHT pagers are numeric message only pagers. However, there is talk of replacing some of these pagers with either email pagers or cell phones with text messaging capabilities. This would certainly improve the ability for someone to receive critical notifications without needing to explicitly check email on a CFHT account.

While the notification system can apply rules based on Status Server data and initiate email notifications, there are also several external dependencies. As part of the design and implementation of the system, these factors must be taken into account.

- Dependable server to send out email notifications. At this point, the mail server at CFHT used to send email is hokua, which is in Waimea.
- For email addresses outside of the CFHT network, a reliable connection to the Internet must be available. At times the UH router/connection has been less than reliable. :(
- Reliable email connectivity and support from target devices. This is dependent upon both system reliability and coverage (in the case of remote devices).

# 4 Message content

## 4.1 Message content must contain a short subject line and Status Server data

It must be possible to specify both both the subject line and the content of the message. Based on earlier discussions, it may be sufficient to have the content of the message be a list of Status Server variables. The message body would then contain the name value pairs of the Status Server variables. An example is as follows:

```
Subject: MP Vacuum Warning

Body:
/i/megacam/fipvar/cryoErrors:
vacuumCritical = FALSE
vacuumOutOfRange = TRUE
vacuumTooHigh = TRUE
vacuumTooLow = FALSE
```

Because some of the target email devices may have text size limitations, it may be premature to define a rigid message content until more detail is known about the target devices.

# 5 Specification of notification recipients

## 5.1 It must be possible to specify either single or multiple email addresses as notification recipients

For the existing MegaPrime cryogenic system notifications, an email alias was established (alert.megacryo@cfht.hawaii.edu) and all notifications are sent to this alias. It is likely that similar aliases will be set up for WIRCam or other notification categories. However, the system should allow the ability to send emails to multiple addresses as well.

# 6 Notification management

This section covers the infrastructure required to enable users to manage the notification rules. It is very likely that some of the GUI windows and forms will be available after the core notification engine has been developed and tested.

## 6.1 Users must have the ability to create, update, view and delete notification rules

The system must be open and accessible such that users can create, modify or remove notification rules. While the system will be provided by the software group, the management of the rules themselves will be a self managed process available to anyone within CFHT.

The software required to view and manage the rules must be available on both Windows and Unix platforms.

## 6.2 Modified rules must be immediately available

A modification of notification rules should not require a restart of the notification process. Instead, the notification engine should dynamically detect and apply rule changes on the fly.

## 6.3 A dynamically updated list of active alerts must be available

When an alert condition is met, it must be available in a list of active alerts until the alert condition is no longer active. An example of how this could be done is to have a directory in the Status Server hold all alerts which are currently active. If the alert condition is no longer met, the alert will be removed from the directory. It would then be possible to monitor this directory and see the updates dynamically.

## 6.4 Remote request and retrieval of active alerts must be available

Once an alert condition has been triggered, it may not be easily apparent if the issue has been resolved remotely. For example, if a problem occurs it may cause a chain reaction of several notifications being triggered. As work is done to resolve issues, it would be useful to query the system and retrieve the current alert status. This query would ideally be available from a remote device. For example, if a specially coded email is sent to specific email address at CFHT, a process under that account could filter the request and send back an email with the contents of the active alert list.

# 7 Document Change Log

| Version | Date | Comments |
|---|---|---|
| 0.1 | April 29, 2004 | First release for review. |