Chapter 3
Every Need to be Alarmed

Ed Young
Young Consulting, Australia

ABSTRACT

Demand for contemporary IT systems to support chronic availability, expansive integration and extensibility has never been greater. Distributed infrastructures and particularly, the advent of Service Oriented Architecture (SOA) introduce new challenges for meeting these demands. Despite architectural conventions to prescribe a common structure and simplified approach, these systems are becoming more complex, heterogeneous and critical. Comprehensive System Management is no longer a luxury. Faults and potential failures have to be identified, isolated and addressed, and ideally pre-emptively. Our front-line indicators are alarms.

INTRODUCTION

Definition: Alarm

- A device that signals the occurrence of some undesirable event
- An automatic signal...warning of danger

(WordNet Search – 3.0, (Search term: ‘alarm’), 2008)

An operational management alarm is raised by a component whose state satisfies pre-determined conditions within a monitored system.

Law: Second Law of Thermodynamics

Krafzig et al. (2005) observe that the second law of thermodynamics, any closed system cannot increase its internal order by itself, has application in solution design. It also rings true for system monitoring. Any internal instruments geared towards ordering the system will ultimately increase overall disorder (entropy) therefore it must be externally defined and regulated. These are the roles of the Solution Architect and system alarming (monitoring) respectively.
Every Need to be Alarmed

ALARMING STRATEGY

Good system architecture demands attention is paid to capacity, availability, security (see Security), reusability and flexibility amongst others. Single points of failure need to be identified and eradicated through redundancy and balanced resource allocation. Despite these efforts, there will always be potential for a system to fail. An alarming strategy serves to identify potential weaknesses and mitigate their consequences.

The prevalence of SOA (see Service-Oriented Architecture (SOA)), distributed architectures in general, utilising loose coupling and dislocated services while promising little or no disruption to service makes failure analysis and mitigation more important than ever.

It is not possible to develop a comprehensive alarming strategy without a detailed knowledge of the system to be monitored and a thorough appreciation of what alarming Instruments and approaches are available.

The strategy must:

- Evaluate the consequences of the alarm being raised to determine severity (see Severity Descriptions)
- Identify severity thresholds (See Instruments of Measure). Use stress and volume testing for calibration early.
- Determine what should happen should when a particular alarm is raised, who should be informed and how
- Identify reporting requirements (see Reporting)

This is particularly important for bespoke elements of the system as they required specialised attention to accommodate any non-standard behaviour.

It is important to visualise the system and apply the alarming strategy to it.

When identifying elements of the system to be alarmed, consider the six general architectural stack elements, how they interact with each other and potentially, how they interface with external services. Juxtapose each element as if matryoshka dolls each contained within the other and consider their relationship to each other.

Rules

These rules are used to model the alarm system interactions:

- **Calculation Rules** are used to determine the importance of a node to the system
- **Propagation Rules** are used to determine how an alarm impacts a particular node or service.

ALARMS

Types of Alarms

Operational alarms can take many forms. It is possible to group them loosely into two types; agent-server and agent-less.