META Group

META Practice

9 June 2003 Practice 2047

Command Centers: Consolidating Incident Management

Operations Strategies, Service Management Strategies

Glenn O'Donnell

FOCAL POINT

IT infrastructures and applications are notorious for generating a flood of messages about their condition (e.g., “server unavailable,” “performance threshold exceeded”). The sheer volume of information makes fault isolation a daunting task. Management tools have proliferated to sift through these event messages to identify actual root causes worthy of investigation. Some success has been achieved at this root-cause identification, but efforts will be fragmented through 2008, hampering full end-to-end event management. False alarms, spurious transients (extremely short duration), and secondary effects of root causes can all generate event messages that must be suppressed. Dispatching personnel to evaluate and attempt to resolve all unprocessed alarms is wasteful and fosters apathy about such messages. As a result, most messages tend to be ignored until an end user complains. This process can be easily automated to dramatically bolster operational efficiency.

CONTEXT

A robust incident management process, enabled by event management and help desk technologies, can greatly reduce false alarms by intelligently processing incoming events and determining which events truly require attention. Many organizations have implemented some level of incident management, though most efforts are fragmented based on infrastructure technology domains (silos) and little intelligence is built into the processing systems. Organizations should consolidate all domains into a single incident management system with automated event management under the operational directive of the command center (see SMS Delta 1002) and in conjunction with the help desk for incidents initiated by end users.

Incident management consolidation is currently under development in 40% of Global 2000 (G2000) enterprises. By 2006, this will grow to 60%-70%, with 50% of G2000 enterprises successfully centralizing incident management under the jurisdiction of the command center (regulatory and cultural forces will prevent inclusion of the security domain into the unified system). Operational maturity is growing slowly in G2000 organizations. As this trend continues, process integration beyond simple incident management will yield iterative improvements in the entire IT organization (e.g., infrastructure and application refinement based on incident data). By 2006/07, the output of a consolidated incident management process will produce profound enhancements in infrastructure reliability and performance in the 50% of G2000 with consolidated efforts. By 2008/09, this process integration will be so commonplace that infrastructure management performed without such feedback to development will be widely recognized as absurd. This robust cross-process integration is necessary for high operational maturity levels.

META Trend: During 2003-05, new infrastructure and application architectures (e.g., Web services, virtualization, utility computing) and budgetary restrictions will drive additional focus on capacity planning and meta-management efforts (e.g., status aggregation, business views), leading to more integrated meta-management tools (2007). Through 2006, continued cross-boundary management demands (e.g., organizational, informational, technical) will drive process, sourcing, and instrumentation changes.

Command Centers

The command center is a superset of what is commonly known as a network operations center (NOC). Because the incident management process spans more than just network infrastructure, the command center should also encompass more. All IT hardware and software infrastructure components and applications should be monitored for performance and availability by the technology within the command center. It should coordinate closely with the help desk and share technology. Although the command center is best suited for automated incident detection, the help desk is best as an interface to end users and as a conduit for incident escalation.

Incident Management Process Maturity

Mature incident management processes are scarce (10%-12% of G2000 enterprises). The overwhelming majority of organizations

(70% of G2000) are inefficient and fragmented and characterized by frenzied firefighting instead of smooth discipline. These immature organizations must navigate the path to maturity with small steps aligned with an overall strategy toward consolidation and structured operations. Maturity steps and transitions are defined in SMS Delta 1056.

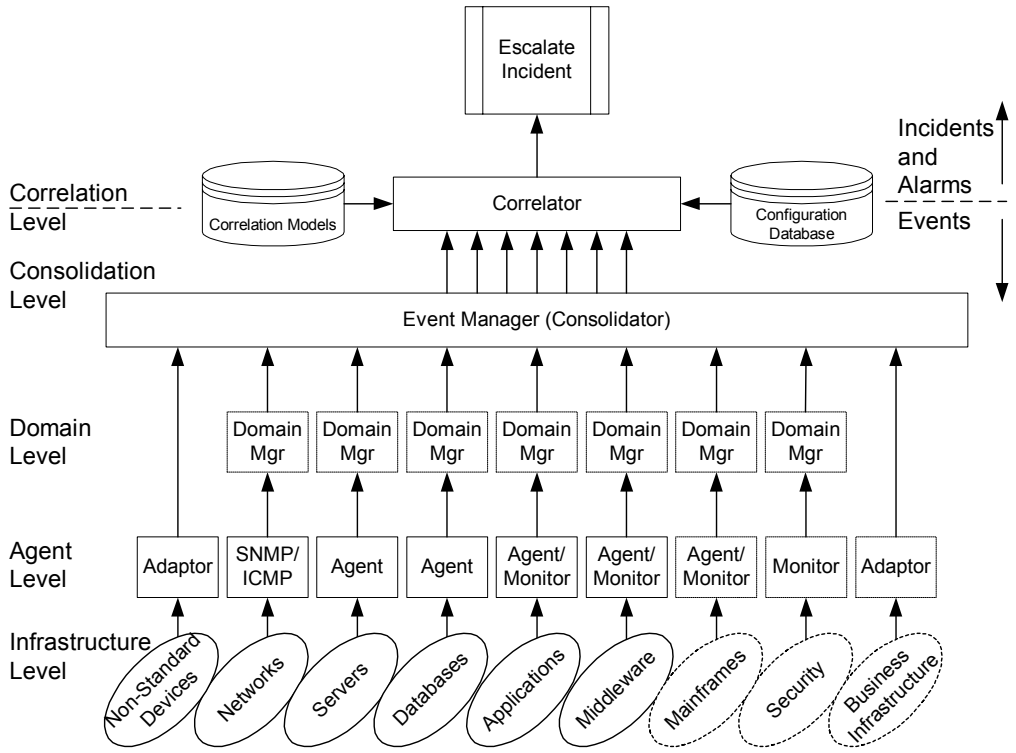
Focus on Applications

Although monitoring is mandatory at both the infrastructure component and application levels, applications are the primary focus of incident management. Only application-level incidents can accurately reflect impact on end users. Fault-tolerant infrastructure is now common, so a failed network switch or server may have no real influence on end-user service levels. Focusing on components is merely a loose inference on end-user effects and is rarely successful at this inference. Direct measurement of the application is much more accurate. When an application-level incident is generated, component-level incidents are examined to determine root cause. Component failures independent of application incidents are still important and require resolution, but the relative importance may be lower if they do not directly affect users.

A Modular Architecture for Incident Management

The incident management system should be developed in a modular fashion for flexibility, scalability, usability, and overall system performance as well as to address the specific needs of each managed technology domain. Figure 1 illustrates this modular concept. Although it reflects management tools, the process itself will follow a similar data flow and modular architecture. No single tool vendor can effectively provide the entire system because of technology limitations, costs, and the installed base of management tools.

Figure 1 — Consolidated Incident Management Modules and Information Flow



Source: META Group

Infrastructure Level

All IT infrastructure and application domains should be monitored and their data included in any incident management, including non-standard devices (e.g., telephone switch, data center environmental conditions, UPS power). In many organizations, mainframes and security remain separate from the rest of IT incident management, with mainframe integration occurring as organizations mature, but security continuing to have only a loose coupling in the longer term. Mainframe operations tend to be more mature, so incorporation of this domain should be approached carefully to avoid disruption. As the incident management process and technology mature, the IT organization can extend the incident management system to non-IT infrastructure (e.g., manufacturing process flows, energy delivery, HVAC, physical building security). Doing so can substantially enhance the IT organization's value to the business, but only more mature operations should attempt such extensions. Credibility within IT domains must be demonstrated before assuming broader responsibilities.

Agent Level

Each monitored technology domain at the infrastructure level requires an independent management module at the agent level. These agents will typically fall into one of four categories:

- ***Embedded agent:*** The most common example is a Simple Network Management Protocol (SNMP) agent embedded within a network switch or router. These agents are usually limited to simple data collection and error detection. More robust analysis requires more processing and local memory, resources that can seldom be justified in commodity devices. One exception is Cisco's IOS, which is rich in management functions and is embedded in higher-end products where the extra cost of processing and memory is more easily justified. We are also seeing growing use of Java Management Extensions (JMXs) as an embedded-agent technology.
- ***Adjunct software agent:*** This is a software module that is loaded on a general-purpose computer (e.g., server, mainframe, client) to monitor performance and availability of the system itself and sometimes other software subsystems (e.g., database, e-mail, job scheduling). The agent's monitoring scope is usually limited to the host system on which it is installed.
- ***Active or passive monitor:*** Monitors can be in the form of an adjunct software agent or an appliance. In the case of an adjunct agent, the agent's scope usually does not include its host, but an external component or service (e.g., application performance, security intrusions). Appliances are growing in use for such management functions (see SMS Delta 1054) because of their simple packaging, low operation cost, and rapid time to value. A common passive monitoring appliance is an RMON probe (see SMS Delta 1081). Newer appliances perform more sophisticated analysis for needs such as end-to-end application performance, external Internet path performance, and security breaches. Active monitors are most often used for measuring application performance and availability by generating synthetic application transactions at periodic intervals.
- ***Adapter:*** An adapter is a device or software component that translates a non-standard data source into a format that can be understood by the management system (e.g., convert analog temperature or gas flow monitors into SNMP-compliant data).

Agents (whether embedded or adjunct) and adapters commonly detect events for a narrow domain (e.g., one server). Monitors are most effective when covering a broader range of measurements (e.g., multiple applications covered for performance management).

In the case of server monitoring, concerns are often raised regarding different operating systems. Unique tools for each operating system class are common (e.g., NetIQ for Windows, BMC Software for Unix, Tivoli for OS/400). If all agents ultimately report into the same event consolidator, concerns about different operating systems are largely irrelevant. That said, economies of scale and operational simplicity are possible if one vendor's product can support all operating systems. Increasingly, "agentless" server monitoring is being employed, further minimizing operating system concerns (see SMS Delta 1143).

Domain Level

Each technology domain (e.g., network, database, server) can optionally use a management consolidator for its own domain. Although these domain managers are not technically necessary in most cases, they can enhance overall management system scalability by delegating some of the load away from the centralized consolidator. Domain managers can perform event filtering and distributed correlation and reduce the number of events fed upstream. If a domain manager were desired for server management, for example, it would be wise to pursue a common vendor across all servers. Otherwise, a different domain manager would likely be needed for each agent vendor. Depending on the domain, heterogeneous domain managers are neither mature nor available at all. Again, agentless management can mitigate this issue.

A domain manager is also a good way to give some control and visibility to domain subject-matter experts (SMEs) without compromising the integrity of the overall system. SMEs will act as the higher tiers for incident escalation, so the data should be accessible for them. Historically, many SMEs already have experience with some of the domain managers. In fact, consolidated systems are often assembled from such tools that were previously under the complete influence of the SMEs. Such continued familiarity and access will soften political hurdles to evolving consolidated management.

Consolidation Level

The event manager (consolidator) is basically a simple function. It receives events from the various domains and feeds them to the correlator, where the system's true intelligence resides. The consolidator can perform limited correlation (e.g., event de-duplication to suppress repetitive events). When evaluating consolidators, clients should seek those that will support integration with a wide variety of data sources. At the very minimum, integration with all anticipated agents and domain managers is essential.

The consolidator need not be a single instance of software. In the very largest of environments, it is desirable to split the consolidator across geographies or business units for scalability or business needs (see SMS Delta 1145). Such functional splits are difficult or impossible with some consolidators. As technology matures, consolidators and correlators are merging into unified event management products, blurring the line between the consolidation level and the root-cause level. The two need to remain conceptually separate, even if the tools themselves are unified. Although it makes sense to distribute event management consolidators, the top-end correlator must remain centralized for optimum benefit.

Correlation Level

The correlator holds the most power in consolidated incident management. It exploits the intelligence within the configuration database and the correlation models to reduce false alarms and escalate only the true root causes of incidents. Without good correlation, consolidated management will yield an unbearable load of escalated events, rendering the system ineffective and likely counterproductive.

Correlation models make use of technology relationship maps (see SMS Delta 1044) to determine the associations between infrastructure components, between components and services, and by alignment in time. Several correlation methods exist (e.g., state diagram, topology, temporal), but no single method is optimum for all correlation needs. Clients should carefully evaluate correlation products to determine their suitability to the desired task and the amount of manual effort necessary to build and maintain models. Mature domains (e.g., network, servers) have several good correlation options that require little or no manual intervention. Emerging domains (e.g., n-tier applications, Web services, business perspectives) still require extensive effort to maintain models. As noted earlier, companies do not typically purchase correlation tools standalone, but rather as a function of an event management consolidation tool.

Accurate configuration information is critical to the success of correlation efforts. Models extract configuration information to build the necessary relationship maps. A good example of this synergy is in the network domain. Network topology is automatically discovered to populate the configuration database. Correlation relationships directly map to topology information, so topology autodiscovery results in automated correlation model development and maintenance. As emerging technology domains mature, more relationship maps will be automated.

Events, Incidents, and Alarms

Messages in the incident management system can be classified as events, incidents, or alarms. Events are low-level messages indicating exception conditions based on defined rules; they can indicate occurrence of a normal condition (e.g., job completion) or they can identify status changes (e.g., new router added). They can be generated asynchronously by infrastructure component triggers (e.g., performance threshold violations) or detected by polling failures. Raw events must be processed to convey significant meaning. An event has context only in the sense that some condition has occurred in a specific component or application transaction. It does not necessarily indicate an actual failure or other problem. That determination is left to the consolidator and correlator.

After the correlator processes events, the resulting outputs are incidents. Incidents have genuine merit as actual problems, if the correlation engine is properly configured and operating as desired. Initially, this configuration will be weak and many incidents will not accurately reflect actual problems. Over time, however, continual development of the correlation engine will maximize the likelihood that an incident is a true incident. Processed incidents, not raw events, should be escalated to SMEs for resolution.

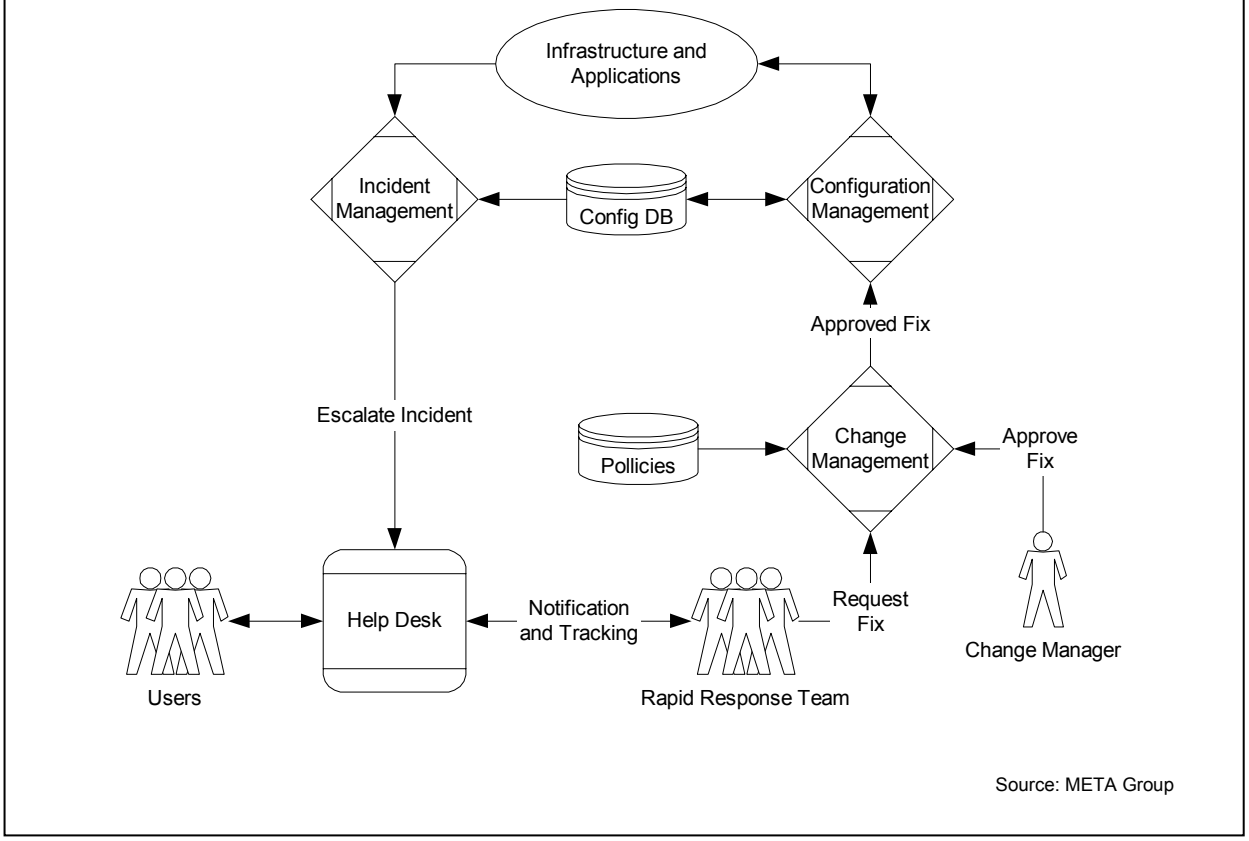
Alarms are high-priority incidents. Incidents reach alarm-level priority based on impact or time. Impact is a configuration attribute of the device, application, or other incident generator, and this attribute is used by the correlator to raise its priority. Priority can also be elevated in the escalation stage (performed by help desk technology). If an incident has gone unresolved beyond a time limit, it is escalated to a higher support tier and its priority is also raised. The incident can reach alarm status after a high time delay.

Events, incidents, and alarms all represent the same condition. Their distinction is in the amount of processing each has received to determine its urgency and its relevance. This distinction is important to the incident management process's development and the architecture and design of the management technology. Not all events become incidents, and not all incidents become alarms.

Escalation, Tracking, and Resolution

When an incident is generated, it is sent to the help desk (service center) for escalation, tracking, and resolution (see Figure 2). The trouble-ticket solutions used by the help desk are tailored for these escalation tasks. This results in a single reporting system for tracking history of all enterprise incidents or alarms. In addition, help desk staff members are not expected to respond to the command center incidents; however, they do now have access to the open incidents and alarms, therefore allowing them more data for communication to their users. By centralizing the escalation and notification at the help desk, it minimizes the number of locations at which rules are created and maintained, therefore streamlining maintenance. The trouble-ticket system will determine the appropriate rapid responders based on the context of the incident and notify them about the situation (see SMS Delta 1092). Usually, this initial escalation should not require human intervention. The automation technology (i.e., the trouble-ticket system) should make these decisions autonomously. Of course, this requires careful planning and configuration of the system, in a cooperative effort between the command center automation team (see SMS Delta 1088) and the help desk automation team.

Figure 2 — Incident Escalation, Tracking, and Resolution



When incident management is consolidated from isolated technology domains, the existing systems will sometimes have implemented their own notification mechanisms to page or e-mail the rapid responders when incidents occur. Sometimes, even the consolidated system designers wish to use such techniques. This practice should be discouraged, because this notification is best performed by the help desk. Leveraging the help desk's notification process encourages a more-streamlined incident management architecture, offers economies of scale, and reduces duplicate notifications and general chaos.

The help desk is also best at tracking the history of the incident and enforcing higher-level escalation if resolution is delayed. The technology can measure operational performance metrics (e.g., mean time to repair [MTTR], percentage of incidents resolved at each support tier, percentage of incidents manually entered by help desk staff vs. those automatically detected by the command center).

Bidirectional integration with the help desk is desirable to enable a sanity check when incidents are resolved. The automated event detection and processing should recognize the resolution, and the trouble-ticket system should track the incident's closure by the appropriate rapid responder. Discrepancies between the two events reflect a process breakdown. Such breakdowns should be investigated to minimize future occurrences.

Iterative Efficiency Improvement and Integration With Problem Management

As time passes, the command center's long-term data can be analyzed for troublesome patterns, driving the organization to be more proactive. These patterns help identify chronic problems that, when resolved, can produce dramatic benefits for the operation. See Figure 3 for a simple example of a top-20 listing of event generators for a given period.

Figure 3 — Sample Listing of Top Event Generators

Events	Class	Mfgr	ID	Location	Percent
843	Router	Cisco	UK047D12R02	London	8.24%
620	Router	Cisco	UK047D12R06	London	6.06%
505	Router	Cisco	US101K01R01	Atlanta	4.93%
481	Unix	Sun	US001A01U22	New York	4.70%
311	Router	Cisco	US001A01R04	New York	3.04%
298	NT	Dell	US101B01N10	Atlanta	2.91%
281	Database	Oracle	US001A01D05	New York	2.75%
277	Router	Cisco	UK047D11R01	London	2.71%
246	Router	Cisco	UK047D12R04	London	2.40%
223	NT	Dell	HK001A02N03	Hong Kong	2.18%
207	Router	Cisco	US041B01R01	Chicago	2.02%
188	NT	IBM	US007A02N04	Denver	1.84%
171	Switch	Cisco	UK047D12S05	London	1.67%
158	Unix	Sun	US001A01U13	New York	1.54%
155	NT	Dell	US101B01N04	Atlanta	1.51%
149	Router	Cisco	US007A02R01	Denver	1.46%
143	Router	Cisco	US015B01R01	Seattle	1.40%
132	Switch	Cisco	JP001A02S02	Tokyo	1.29%
130	NT	Dell	US101B01N06	Atlanta	1.27%
127	Database	Oracle	US001A01D02	New York	1.24%

Source: META Group

Analysis of this list shows some clear patterns. Nine of the top 20 event generators are Cisco routers, representing 32.26% of all events. Obviously, there are problems with Cisco routers in this situation. Resolution of the problems will reduce the total incidents by almost one-third. Performing this exercise periodically will yield significant cumulative improvements in operational efficiency and infrastructure stability. Without this long-term event data collection, improvement attempts are based on mere conjecture, and quantifying true value to the business is impossible.

These longer-term data studies highlight how the incident management process can benefit the problem-management process and vice versa. Deep analysis of long-term data is usually a problem-management task. This task requires the incident management data and provides benefits the entire business.

Other tweaks to the technology and the process itself will also lead to exceptional cumulative advantages. A notable example lies within the correlation models. Iterative reassessment of correlation models will continually fine-tune the models and improve the event-to-incident ratio. The probability that an incident reflects an actual problem condition will eventually exceed 90%. This is noteworthy because initial correlation efforts will produce a probability in the 50%-60% range, still a remarkable gain over the 20% probability without correlation. Reaching the high nineties is unlikely, even in the most mature operations.

Integrating Point Solutions

The cornerstone of an incident management plan will be tool integration. Tool integration has long been an issue with infrastructure and application management, and it will continue to be an issue for the foreseeable future. Some integration is fairly simple, however. At the basic event level, almost all event and incident management products conform to some official and de facto standards. A widely supported baseline integration mechanism is the SNMP trap. Almost all management agents and domain managers can generate traps that feed into the next-higher module of the incident management system, where trap reception is ubiquitous. Other integration points are proprietary remote procedure calls (RPCs), proprietary application programming interfaces (APIs), and emerging XML-based standards from the Distributed Management Task Force (DMTF). The DMTF's Common Information Model (CIM) and the corresponding XML encapsulation (xmlCIM) are attractive for tool integration. CIM adoption has been slow, but growth is accelerating, fueled by security concerns and cross-firewall transport requirements.

Although standards are always preferable, proprietary integration is inevitable. The more popular vendors (e.g., BMC Software, CA, HP OpenView, IBM/Tivoli, NetIQ) enjoy broad support for their integration points. Other vendors must build their management products to easily integrate with these popular packages and other point solutions desired by their customers. The resulting integration will be effective, but limited. Full integration requires significant development effort, even within single-vendor frameworks. To share heterogeneous configuration information, command-center automation teams will need to employ a few staff members with software development expertise in C++, database (especially Oracle and SQL Server), Java, and other Web technologies. Additional expertise or training in management tool APIs will also be necessary.

Bottom Line

Incident management should be consolidated across all technology domains to streamline response to infrastructure and application failures. Efforts should be organized under the jurisdiction of the command center in cooperation with the help desk and technology domain subject-matter experts.

Business Impact: Structure, discipline, and continual improvement in the IT organization's command and control can considerably reduce IT operational costs and quantify business value.