Accelerating Problem Resolution: Why It Matters and How to Do It

By Liam McGlynn
Senior Analyst - Enterprise Management Associates

The average organization suffers 61 hours of downtime each year with costs exceeding $1 million per hour for larger enterprises.
And, these same organizations waste the initial 54% of that downtime just trying to determine who should fix it!
Although automated Problem Isolation recovers millions in outage costs by eliminating the detection and identification steps in each incident, few enterprises have achieved these results due to poor architecture and, until recently, a dearth of effective software tools.
Times have changed. The tools are now available for comprehensive and adaptive Problem Isolation. For those who want to save a few million dollars while also improving customer service, you have an alternative to otherwise draconian budget cutting.

Problem Isolation is one of the most challenging tasks in IT management, so it should come as no surprise that the automation of Problem Isolation is a formidable undertaking. However, since outages cost money – LOTS of money – investing in the automated isolation of problems makes sense for many businesses.

In each system outage, the business suffers degraded customer satisfaction (lost revenue opportunities) and decreased staff productivity. In large organizations, the impact can exceed $1 million per hour and in virtually every organization, IT management lists Availability as a primary objective. Despite this imperative, EMA research (EMA Research Report, Data Center Automation: Delivering Fast, Efficient, and Reliable IT Services) shows that the average enterprise suffers more than 61 hours of downtime each year (99.3% availability).

Organizations can reduce downtime by as much as 40% by automatically isolating 80% of the problems. This extends far beyond ITIL v3's Event Management. For successful Problem Isolation, the keys are discovery, dependency mapping, event collection, event correlation, business impact analysis, process orchestration, and an organic organizational structure that enables continuous service improvement (CSI).

As complicated or arduous as this may sound, solutions exist that simplify the implementation and growth of an adaptive Problem Isolation environment. And organizations that adopt advanced Problem Isolation stand to enjoy significant advantages in operating margin, staffing agility, and customer service levels.

Why Problem Isolation Matters

There are three basic phases in any system outage – *Detection, Identification,* and *Resolution* – and each phase requires a different approach to automation:

- **Detection**, the first phase, can originate from several sources. With automation, one would expect any failing component to generate an alert. The challenge in this type of automation is targeting the conditions that constitute "exceptions" and "warnings." This parallels ITIL v3's Event Management process.
- **Identification**, the second phase, becomes complex and time-consuming in the presence of multiple exceptions, especially when these exceptions originate from various infrastructure components (network, application, end-user monitoring, servers, etc.). This kicks off the Problem Management process.

- **Resolution**, the final phase of an outage (but not the final process in Incident Management), requires diagnosis, analysis, and remediation. This is primarily a Problem Management phase, and can involve Change Management as well.

We found that organizations spend 54% of each outage detecting and identifying. This is an ideal opportunity because the first two phases are much easier to automate than the resolution phase, thus yielding a majority of the benefit. The most tangible benefit of outage reduction is employee productivity. The equation is simple:

$$Cost = \frac{Outage\ Minutes\ \times\ Affected\ Staff\ \times\ Annual\ Wage\ \times\ Percent\ Inefficiency}{52\ \times\ 40\ \times\ 60}$$

For example, our research shows that the average outage lasts 87 minutes and the average wage is \$68,000. If 5,000 employees lost 33% productivity for the duration of the outage, the equation looks like this:

$$Cost = \frac{87\ \times\ 5000\ \times\ 68{,}000\ \times\ .33}{52\ \times\ 40\ \times\ 60} = \frac{9{,}761{,}400{,}000}{124{,}800} = \$78{,}216$$

This approach does not account for reduced revenue potential, customer dissatisfaction, or tarnished goodwill. In large financial enterprises, outage costs can easily exceed \$1 million per hour.

Making it Work

To understand Problem Isolation from an architectural perspective, one must understand its layers of maturity. Figure 1 shows a high-level flow of the ITIL v3 processes and activities that underpin Problem Isolation. The core processes are Event Management and Incident Management with inputs/outputs in Problem Management, Change Management, Knowledge Management, Request Fulfillment (process orchestration), Service Level Management, and Availability Management.

At level 4, events converge into an Operations Bridge (OB). ITIL defines an Operations Bridge as a "physical location where IT services and IT infrastructure are monitored and managed (*Service Operations, 5.2.1 Console Management/Operations Bridge*). Although the degree of monitoring, filtering, and automation varies considerably between technology silos, correlation is a weakness at Level 4 because even simple event streams require extensive technical and organizational (political) collaboration.

Because of this, the collaboration of incident response faces two major obstacles. First, multiple events, especially during a serious outage, generate chaos and consternation. Second, when a company is losing \$10,000 per minute, the reluctance to admit culpability generates political tension, finger-pointing, and long delays in event characterization. Level 4, though an essential foundation for Problem Isolation, does not meaningfully mitigate the impact of serious cross-service outages.
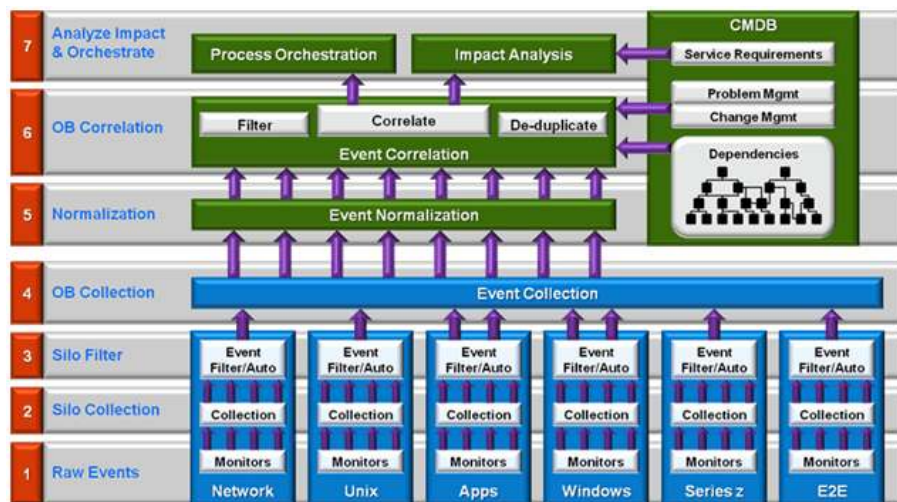

Figure 1. Problem Isolation Layers of Maturity.

Enterprise-wide Problem Isolation does not begin until Level 6 (Ops Bridge Correlation) and requires the underlying

layers of event collection; an advanced, topology-based correlation engine; and architectural components that enable correlation. The supporting architecture is a Configuration Management Database (CMDB) populated with discovered configuration items (CIs) and the interdependencies of those CIs (dependency mapping). Because relational topology is critical to dynamic correlation, the process of discovery and dependency mapping must be comprehensive, accurate, highly automated, and adaptive (change-aware).

The choice of correlation engines requires a bit of digging. Correlation is much more than downstream event filtering. The task of assigning each event a probability of cause demands analysis of problem history as well as algorithms to separate events into multiple classes for rapid analysis. The analysis of problem history typically employs a similarity algorithm in conjunction with problem metadata or "fingerprints." Today, there are perhaps a handful of advanced solutions for enterprise-level correlation.

On an architectural level, correlation must have awareness of problem history, change history, and topological dependencies. Of these, the greatest challenge is topology where an automated tool maps application dependencies across the discovered technology landscape, including database constructs, application programs, web components, dynamically allocated virtual infrastructure, and more.

A topology-based correlation engine processes event streams based on classes of parent-child dependencies. With automated discovery and dependency mapping, the correlation engine remains insulated from the frenetic pace of IT change. This simplifies the administration of correlation rules though correlation, automation, and process orchestration, but still requires considerable effort. However, with the right structure, the effort is both modular and extremely productive.

Automated problem resolution depends on but is not part of Problem Isolation. By automating Problem Isolation, an organization reduces outage duration by more than 50% with very little risk to the enterprise. In one recent survey, 63% of respondents did not want tools to take even some actions automatically. This would suggest that the implementation of automated problem resolution faces major hurdles. However, Problem Isolation is achievable today.

Summary

Though self-healing data centers are still sketches on a whiteboard, the first step is self-diagnosis or automated Problem Isolation. This is a formidable objective because it includes requirements for architecture, advanced correlation algorithms, topology-awareness, adaptive dependency mapping, and integration with ITSM processes like Change, Problem, and Configuration Management.

Despite these challenges, many organizations see Automated Problem Isolation as yielding= consistently high service levels in tandem with dramatic cost reductions.