

Managing mission- critical applications with Verax NMS

A service assurance approach



Table of contents

Abstract	3
1. Introduction	4
2. Handling information chaos – a service-oriented approach	4
2.1. A service assurance approach.....	5
3. Verax NMS implementation	8
3.1. Business view aspects	8
3.2. Rules-based automation.....	9
3.3. Metrics and KPIs	10
3.4. What if there are other monitoring systems already in place?.....	11
3.5. Summary	11
4. What about the cloud?	12
5. Summary	12

Abstract

The objective is to demonstrate how complex, mission critical application systems can be managed with Verax NMS using an IT service assurance approach, rather than traditional monitoring.

We strongly believe that the service approach is required as corporate IT is getting more complex and new services and applications are deployed. The objective is to minimize service downtimes, prevent issues before they affect the service, increase efficiency by allocation of resources to the most important tasks and increase customer satisfaction, which altogether inevitably maximize the long-term profitability of an enterprise.

Intended audience

This paper is intended for users and vendors of enterprise mission critical systems. These systems may include fraud detection in telecom or banking, on-line sales, e-commerce and others, especially those that are revenue-generation or revenue-protection focused and require 24/7, on-line uptimes.

1. Introduction

In today's world, every aspect of a modern company's activity is supported by IT systems. As the business becomes more and more complex to survive in the competitive world, so does the IT. Large, complex, heterogeneous IT systems are like people. They have moods, likes, dislikes, good and bad periods. In other words, they need to be taken care of.

This is where the system administrator comes into the picture. A good administrator is a Guru who "knows what to do" if or when something happens. And he or she needs information. Lots of it. This information is needed to "get a feeling" of the present condition of the system. Good administrators are hard to find and their attention has to be split across many tasks as the companies are struggling with staff shortages and cost cuts.

That's where Verax Systems' value proposition kicks in: what we propose is an **IT service assurance** approach rather than a traditional network management, as eventually, it simplifies and lowers the cost of management of complex applications, data centres and services.

Service assurance (SA) is an all-encompassing paradigm around minimizing service downtimes, prevention of issues (rather than dealing with failure aftermath) and increasing customer satisfaction which altogether inevitably maximize the long-term profitability of an enterprise.

2. Handling information chaos – a service-oriented approach

Let's take a look at what is required to make a hypothetical application run – in other words, what makes up an IT service, such as for instance a telecommunications billing system.

Hardware

The obvious choices here are computers (servers, workstations); however, there is a number of supporting systems such as:

- Uninterruptible Power Supplies (**UPSes**).
- **Interconnecting networking gear** such as switches, cabling, fiber-channel connections, etc.
- **Backup infrastructure** (essential for most applications, although its failure does not affect the service) ranging from physical tape drives to backup software agents.
- **Network infrastructure**, for instance: edge routers, backup links, load balancers, firewalls, etc. These can range from simple passive elements to complex pieces of hardware such as deep packet inspection (DPI) probes.
- RAID arrays and SAN storage devices.

Software

Today's applications typically require use other software packages in order to run, such as:

- **Database servers** (which themselves can be redundant and distributed across multiple hosts) e.g. Oracle RAC configurations. For large services there might also be a master slave duplication of MySQL, even across various types of operating systems and hosts. A popular configuration is to use MySQL as a "caching" database with a master copy persistent in an enterprise-grade DBMS like Oracle.
- **File storage**, for instance for static content such as images or documents. This storage itself may be distributed over a network, in case of SAN.

- **Application servers** and transaction monitors such as Java J2EE servers, which can distribute processing across a number of hosts – it's worth noting that these themselves can be distributed. For instance, in massive telecom systems requiring a lot of processing (such as Billing, Revenue Assurance or Fraud Management) it is often a case that Sun SPARC Solaris is used for the central node and PC Linux nodes can be used for workers.
- **HTTP servers** such as Apache. These servers can be organized in a number of logical blocks such as static content farm, multimedia, dynamic content, etc.
- **Virtualization infrastructure and terminal servers.**

The application itself

On top of standard, infrastructure monitoring, there are certain aspects that are application-specific and require deep knowledge of the application. For example:

- **Own application logs or events.** For instance, a telecom billing application may report CDRs (call data records) that it is not able to parse or rate. These may appear due to switch software updates or other factors (e.g. new number pools added). It is also important to mention, that in this case the problem is completely outside the application or associated infrastructure.
- **Child/background processes** for performing background tasks. For instance, an application may spawn processes that also require monitoring (e.g. data aggregation in the background).
- **Resource consumption characteristics**, such as typical memory consumption levels, volume of occupied disk storage, number of transactions made on the database, number of connections used and others.

Gartner's study highlights the most important factors which makes application monitoring far more complex than monitoring of physical infrastructure. According to the research these are:

- Evolution of applications' architecture – changes in development trends such as agile development resulting in increased modularity of an applications architecture.
- Distributed applications environment – as a compensation for the centralized physical infrastructure.
- Difficulty of discovering the boundaries between one application and another, and the boundaries between application and infrastructure.

User experience factor

This factor is unfortunately very often neglected by organizations. The application may be perfectly working in terms of NMS sensors, however it does not actually mean that the users are working comfortably. From the **user's perspective** it is important that, for instance, HTTP response times are short, files are being served quickly or client GUI does not consume all the workstation's resources.

Of course, the list above is quite simple and does not include all the possibilities.

2.1. A service assurance approach

Traditional approach

There are many interfaces to manage the elements listed in section 2, including SNMP, WMI, JMX, command line scripts and others. The traditional approach is concentrated on placing sensors and setting alarm thresholds on critical points, ranging from simple ICMP ping response time to complex JMX memory consumption parameters. Also hardware and software elements usually support some form of alarming about network conditions, such as SNMP traps or remote syslog events.

Unfortunately, the traditional approach does not scale – this is becoming evident when the managed network and software is growing in scale. **It is not a problem to collect the information, the problem is to use it properly and ignore bits that are not relevant.**

The key problems to monitoring are:

- The alarms are raised regardless of whether the failure affects the service or not. Let's assume a scenario in which static HTML content is served from a farm of separated, load-balanced servers. Single server down, raises an alarm, whereas the failure has a relatively small impact on the service (depending on a number servers in the farm).
- The above causes **flood of information** – it is impossible to understand the dependencies and perform impact analysis. It is important to aggregate the information into **events** that carry important information to the system administrator, e.g. "the system is down due to network switch failure", rather than report a number of individual outages.
- **Sensible notifications** – defining a reasonable notification, prioritization and escalation policy is just as important as event correlation: too many notifications via e-mails or SMSes usually get ignored and the whole mechanism becomes useless.
- **No logical dependencies** - usually NMS systems can discover how individual elements are connected on the hardware level, but this is not sufficient. For instance, an application (service) may have a problem if its master database is down, but an outage of one of the web servers in a farm does not affect the service availability.
- No service-based metric calculations and limited trend-analysis.

On top of these, software packages that are monitoring-focused, have limited element configuration capabilities. Whereas it is relatively easy to reboot a hanging process or machine, it is more complex to adjust Oracle table spaces, or put a WiFi hotspot in maintenance mode and there are few NMS systems that support similar functionalities.

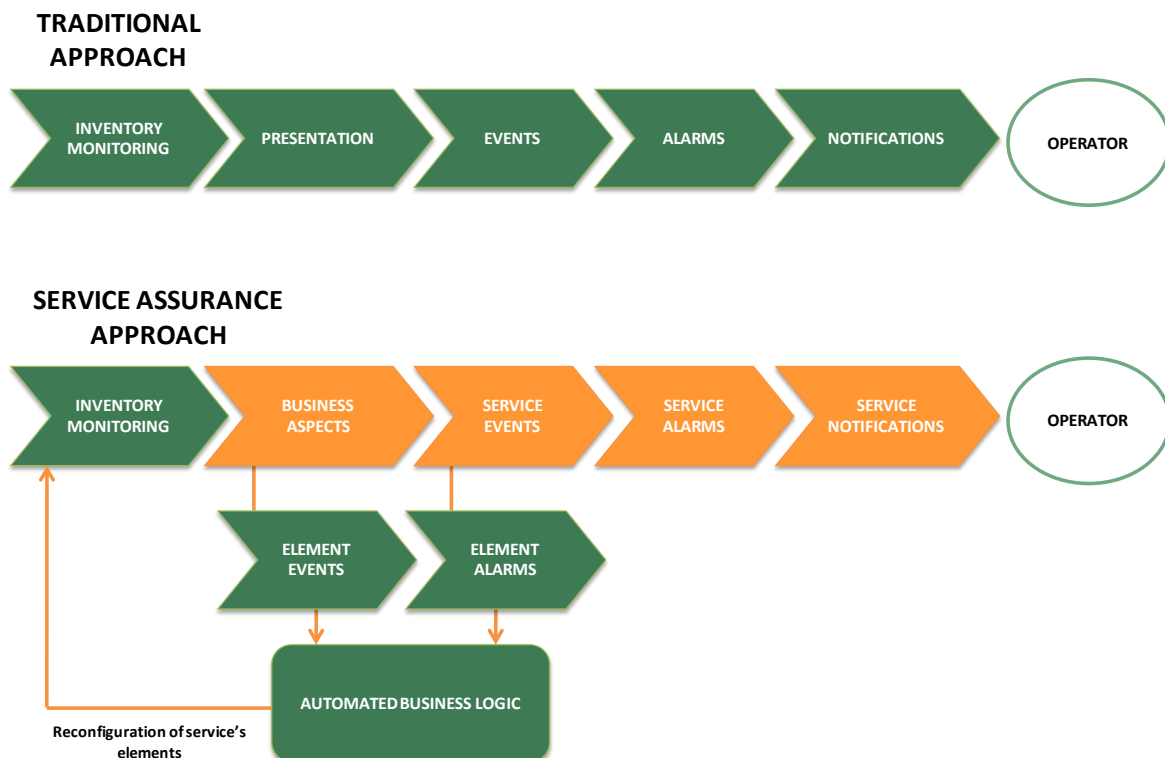


Figure 1: Traditional vs. service based management.

Solution: The service-oriented approach

The application services are now based on complex architectures, using hardware and software elements linked by many different integration technologies. Ensuring the desired level of availability and operational effectiveness of mission critical applications services requires a completely different approach.

According to Forrester research it is crucial for network managers to see entire path of a service across the network, systems, applications and databases, which all are participating in delivering a particular service. The same physical network is used to provide multiple services. Service assurance solution have to provide visibility into network traffic flows as well as application performance across multiple components supporting services.

Traditional monitoring solutions are becoming insufficient in delivering complex service monitoring needs of service providers. Recent research points out that the major factor behind this is “componentization” of the infrastructure – a common situation where tools are used to monitor objects grouped in independent domains from a single management vantage point which makes harder for management group to see entire business service picture. As a consequence of this fact, it is highly desired for network management systems to evolve from resource-centric to more service-centric.

A service-oriented approach is based on two principles:

- Monitoring services on a whole, rather monitoring of individual elements (applications, systems, hardware) and
- Applying automated procedures in case of problems.

Imagine a large data center of a bank – rather than seeing it as a massively complex bundle of interconnected elements, one can see it as a means to provide a number of services such as core banking, internet banking and others. These services are dependent on each other or on an element. A single element may be required for one or more services to function properly. On the other hand, even a number of faulty elements may not indicate a failure to the entire system.

In a way, the service perspective offers a higher dimension compared to the traditional, “flat” view.

Viewing systems from a high-level, service-based perspective allows for:

- More effective allocation of resources.
- Prioritization of problems (rather than seeing a flood of equally important outages).
- Root-cause analyses.

Verax NMS implements the service approach on top of the standard one in the form of a number of means, such as user-defined business view aspects, automated business rules and others described in the subsequent sections. It is also an important step towards ITIL for organizations, as that standard is process and service based.

3. Verax NMS implementation

The subsequent sections describe how Verax NMS implements functionality required to provide a service assurance approach.

3.1. Business view aspects

The service assurance approach is focused on understanding the dynamic business context of each delivered service and being able to indicate which infrastructure components are used in delivering the service. The key element behind implementing service-orientation within Verax NMS is a **business aspect** (or business view). The business aspect is a container/grouping of elements that make up an entire service. Business aspects can be nested, and particular services can be dependent on each other (i.e. status from an underlying service is propagated to the top – greatest severity rule applies in this case).

Rather than being just a logical grouping, a business aspect within the Verax NMS is an entity on its own that is characterized by:

- **Flexibility of containment** – the business aspect can contain any status-aware objects such as hosts, interfaces, applications and others. It can also contain (be dependent on) another business aspect.
- **Custom status calculation rules** – the status of an entire business aspect can be custom-calculated using business rules. For instance, a static web content farm, a fault-tolerant virtualization environment or a distributed database (e.g. Oracle RAC) can potentially have a critical hardware problem (e.g. a machine is down), but from the business standpoint the entire service is in a non-critical state as it is still provided (perhaps the performance is downgraded). Custom status rules can be flexibly defined to meet particular environment needs and requirements.
- A business aspect is itself a **managed object**, therefore it can have its own events, i.e. events for contained elements and custom events for the business aspect. Also, **custom alarms** can be generated for a business aspect. Both custom events and alarms are typically a result of correlation and as such carry a lot of information to the system administrators.

Like other aspects, business aspects in the Verax NMS have rich visualization capabilities such as lists, trees, geographical maps, backgrounds and others.

programming language. In order to speed up the process, the new plugin can be derived from the SMMP MIB-2 standard plugin.

3.3. Metrics and KPIs

Even the most basic course on management mentions that a process should be measured and controlled and introduces the concept of a Key Performance Indicator. Verax NMS has rich capabilities to provide both on-line and off-line metrics. **On-line** metrics consist of mini dashboards and summaries based on the currently displayed view (this feature is available for most of the views). On line metrics include uptime percentages, averages, counts and others. An exemplary mini dashboard view is presented in the figure below.

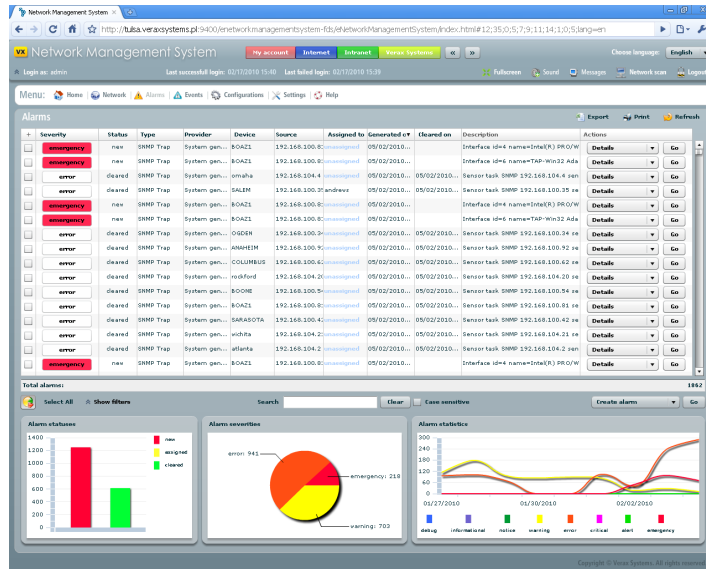


Figure 3: Verax NMS mini-dashboard for element alarms.

Off-line metrics can be built using the business reports feature of the NMS. Verax NMS uses a built-in Jasper Reports engine working under supervision of a built-in process manager. Users can design their own reports and configure their schedules and retention times. The reports can also be manually generated from the administrative console.

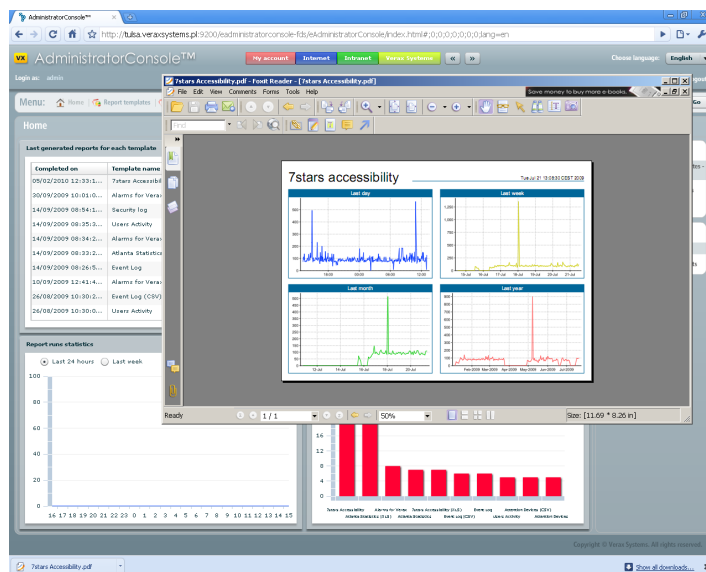


Figure 4: NMS Business reports.

If further data warehousing and OLAP features are required, the NMS can be easily integrated with BI tools such as Verax KPI Dashboard (<http://www.veraxsystems.com/en/products/kpidashboard>).

3.4. What if there are other monitoring systems already in place?

One of the most frequently asked questions is how to introduce the service assurance approach to managing IT, especially when significant investments into traditional NMS systems have been made. One of the possible migration strategies is to isolate critical areas of activity and introduce a management subsystem that would be responsible for reporting to the master system already in place (the umbrella system). Such a scenario is presented in the figure below.

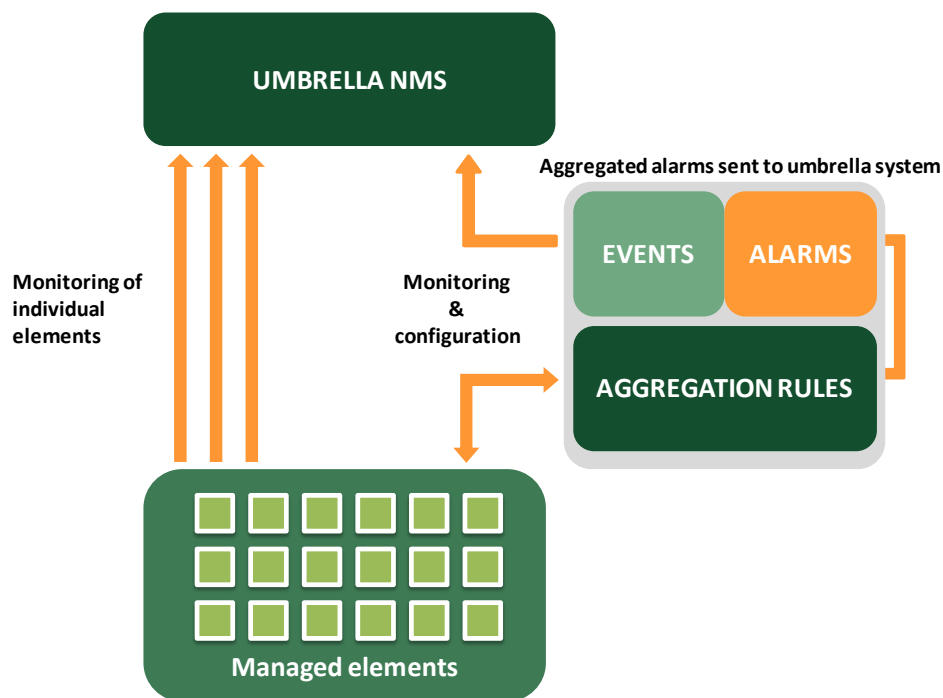


Figure 5: Using Verax NMS as a service monitoring solution reporting to an umbrella system.

In the scenario above, the Verax NMS is used to provide management and automation layer to the heterogeneous database environment (the database farm). The underlying system provides business aspect views, correlation and creation of events, generation of alarms. These aggregated alarms and events are propagated to the umbrella system as SNMP trap or SNMP inform notifications (for manager-manager communication), simplifying the overall management process.

Both underlying and umbrella systems can co-exist in parallel.

3.5. Summary

Thanks to the service assurance approach, Verax NMS can automate the information flow from the bottom up:

- **Capture, process and filter information** about the service and create **meaningful events**.
- **Provide automated, ITIL-like procedures** to improve the organization's efficiency and prevent problems without human intervention.
- **Measure performance metrics** allowing to improve the management process.
- **Report upwards** to higher level management systems.

Altogether, these factors lead to significant reduction of IT ownership costs.

4. What about the cloud?

Cloud computing is definitely a hot topic and more and more organizations are moving towards cloud-hosted systems. In the cloud scenario, the service assurance and SLA compliance becomes even more important as the (partial) responsibility is transferred to the cloud provider.

Cloud providers can offer very good infrastructure uptimes, however the application aspects and user experience aspects (as described in section 2) are outside providers' knowledge. This gives an opportunity to the application vendor to focus on advanced business metrics (such as user experience) and leave the traditional NMS-type monitoring to the cloud provider.

5. Summary

A service assurance approach in general, and Verax NMS in particular, can bring significant benefits to companies, including:

- **Reduction of issue resolution costs** by preventing problems before they have an impact on the service(s) provided.
- Reduction **of downtime** costs by detecting potential problems before they affect the service.
- IT service improvement and increase of **customer satisfaction** by reducing the time required for problem resolution. Please note that the customer may either be internal (e.g. other departments or organization units) or external (e.g. bank users).
- **Shortening of service downtime** through quicker problem analysis via event correlations, notifications and automated business logic.
- Automation of repeatable tasks for improved **operational efficiency**.
- **Collection of service level KPIs and ability to demonstrate SLA-compliance**.

For more information about the Verax NMS product, please visit our website:
<http://www.veraxsystems.com/en/products/nms>.

References

1. Gartner RAS Core Research Note G00173116, "Magic Quadrant for Application Performance Monitoring", 18 February 2010.
2. Forrester Consulting, "Emergence Of Service Assurance For Enterprise IT", 4 Aug 2010.
3. Forrester Consulting Report, The Network Management Software Market, 6 June 2007.