WHITE PAPER

Progress Actional

OPERATIONAL EXCELLENCE FOR DISTRIBUTED, INTERCONNECTED APPLICATIONS

PROGRESS SOFTWARE

TABLE OF CONTENTS

1.0 DISTRIBUTED, INTERCONNECTED APPLICATIONS: BENEFITS AND IT OPERATIONS CHALLENGES

Distributed, interconnected applications are the systems and software approach of choice for the majority of medium and large enterprises.[1] While service-oriented architecture (SOA) was the first widespread manifestation of this approach, other distributed application initiatives, from Web 2.0 and cloud computing, to ESBs, messaging, and business process management (BPM) systems, have been gaining ground. According to the principles of these approaches, IT professionals design, implement, and deploy information systems from components that perform discrete business functions. These components, called "services," can be distributed across geographic and organizational boundaries, can be independently scaled, and can be reconfigured into new business processes as needed. As well as the benefit to business and IT of such flexibility, these initiatives easily allow services to be accessible over the Web, allowing the creation of new on-line businesses and product offerings.

At the same time, IT organizations in every industry sector must meet exceedingly high standards for systems reliability, cost-effective operations, and customer service. Competitive pressures and the needs of both internal and external users allow no room for mistakes in IT operations and applications management. Many are looking to approaches, such as ITIL,[2] specifying best practices for IT management. ITIL defines, for example, the need for the operations and applications groups to provide service-level management, requiring them to meet specific service-level targets (using the broader meaning of "service"), ranging from response time targets for interactive systems to the time it takes to diagnose and correct a problem. Figure 1 shows some sample key performance indicators (KPIs) suggested for a subset of the ITIL management areas.

Figure 1: Sample ITIL Key Performance Indicators[3]

Service-level Management
> Percentage reduction in SLA (service-level agreement) targets missed. > Percentage reduction in the service delivery costs.
Availability Management
> Improvement in the MTBF (mean time between failures). > Percentage reduction in the unavailability of services and components.
Incident Management
> Percentage increase in the incidents fixed before users notice. > Percentage increase in the incidents resolved by first line operatives.

[1]Aberdeen Group, Management and Governance: Planning for an Optimized SOA Application Lifecycle, March 2007. http://www.actional.com/resources/whitepapers/.
[2]Office of Government Commerce, http://www.itil.co.uk/.
[3]Office of Government Commerce, Best Practice for Planning to Implement Service Management, The Stationery Office, Norwich, UK, 2002.

The distributed, heterogeneous, and "loosely-coupled" nature of these applications, as well as new technologies and protocols such as the ESB (enterprise service bus) and SOAP, create new challenges requiring new approaches to the monitoring and problem diagnosis required for IT operations best practices. These, in turn, are essential to assure business transactions execute successfully and support business goals. Yet traditional systems management tools, which are focused on performance and availability at the hardware and operating system level, lack the capabilities to manage service-based, distributed, interconnected systems and applications. For example, consider the following operations management requirements of distributed, interconnected applications, which traditional systems management tools, even with application support and "service-oriented features," are not able to fulfill:

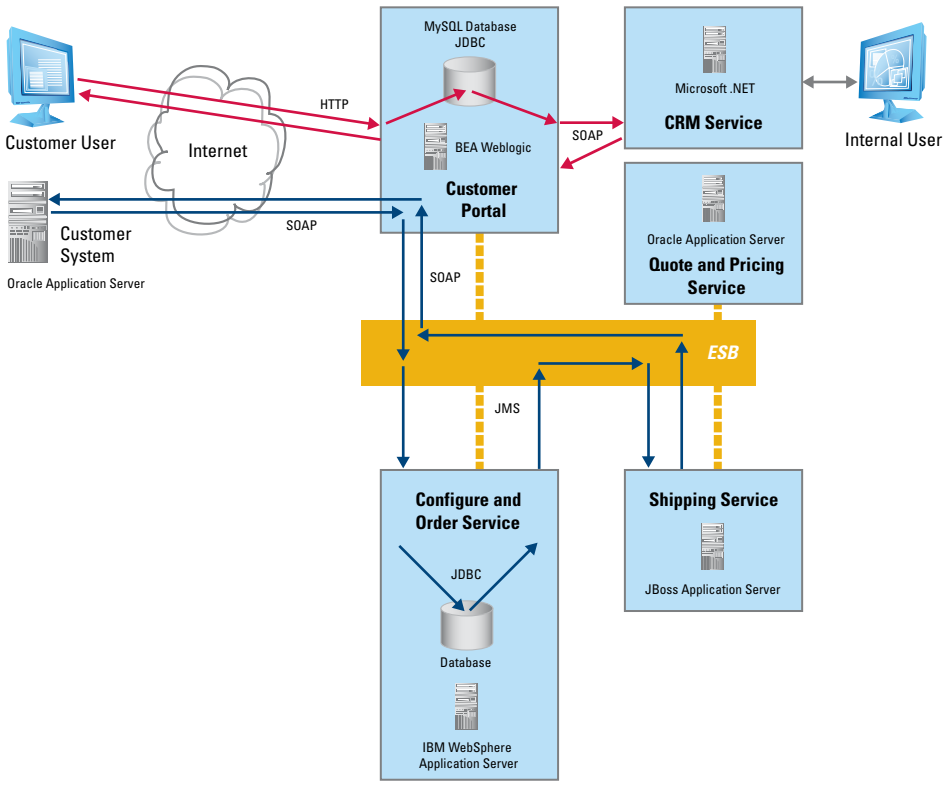
- A hallmark of these interconnected systems is that services are shared—reusing standard business functionality. However, the performance of the shared service may differ according to the context in which it is used. For example, the average response time of a "credit check service" might be 45 msec. when used by an external Web service accessible to other financial institutions, but 10 msec. for an internal requester that calls this service twice as often. In traditional tools, an average response time SLA (service-level agreement) of 30 msec. for the "credit check service" would never indicate that a consumer of the service was having a performance problem. Consequently, there must be the ability to define and monitor a policy with the required response time for a service on a per-consumer basis or as-used basis. Otherwise performance problems can be "lost in the sea of averages."
- Unlike traditional "stovepipe" applications, the shared nature of services and the resulting complex transaction flow can make it difficult to ascertain the cause of a missed service level or a failed transaction. As a result, an operations management system for distributed, interconnected, service-based applications must be able to track the flow of individual transactions. When a service level is missed or a transaction fails, the system must be able to examine each individual step for root cause analysis and problem resolution.

At the same time, it is important that this operations management system integrates with standard systems management tools already in use, so that the operations or application support staff using these tools can be notified of any problems.

Distributed, interconnected systems by their nature are not homogeneous. They span multi-vendor infrastructure components including application servers, database systems, and network devices. Each of these components must be monitored and managed to meet service-level targets for a service request that causes a cascade of requests across the infrastructure components. For example, Figure 2 shows such a system supporting the ordering process for a manufacturing enterprise. The system provides

Internet interfaces to both customers and customer systems through a Web services interface. The example transaction flows are of a customer user employing an external interface to the CRM (customer relationship management) system (flows shown in red) and of a customer system placing an order (flows shown in blue). Each transaction crosses multiple internal services that are running on heterogeneous systems using multiple protocols.

Figure 2: Transaction Flows in a Distributed, Interconnected System[4]



To support the operations management of such a system, a services management system must support each of the following functions across a heterogeneous infrastructure using multiple protocols in a scalable and reliable manner:

Discover > Monitor > Evaluate Policy > Alert > Resolve

[4]Figure derived from A New Service-Oriented Architecture (SOA) Maturity Model, Progress Software Corporation, 2006, http://www.sonicsoftware.com/solutions/service_oriented_architecture/soa_maturity_model/index.ssp.

As explained in the next section of this paper, Progress® Actional® Enterprise uniquely supports each of these functions with the following capabilities built on patented technologies:

> Automatic service discovery

> Non-intrusive, low-latency, runtime monitoring

> Central service-level policy management with distributed policy evaluation

> Policy-based alerting to staff and to other standard management products

> Service network visualization with end-to-end transaction tracing and display

All of these functions are provided in a scalable and reliable distributed architecture with support for heterogeneous environments and for multiple standard protocols.

With Actional Enterprise (hereafter referred to simply as "Actional"), the operations and applications staff can implement best practices and deliver the service levels required by their business, allowing the benefits of service-based, distributed applications to be attained through Actional's ability to monitor service levels, alert on performance and availability issues, and snapshot individual transactions to easily isolate the root cause of problems. Customers report benefits such as:

> Reduction in mean time to resolution (MTTR) by 75% for Web service issues

> Reduction in the number of help desk incidents by 50%

> Automatic detection of problems that were previously reported by customers

2.0 PROGRESS® ACTIONAL® ENTERPRISE OPERATIONS

Actional ensures that IT operations and applications management staff can get the fundamentals of distributed, interconnected operations right. These fundamentals span the range of functionality from service discovery through problem resolution.

Actional includes both a central management server and agents that monitor each individual services network component. The server maintains the repository of service-level policies and distributes them to agents. Furthermore, it maintains the summarized data required for display to the operations staff through a browser-based, highly graphical user interface.

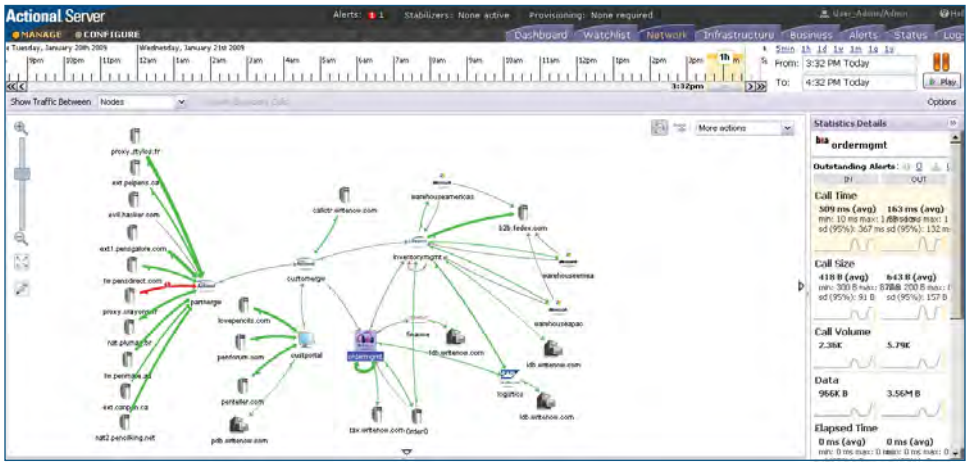
Actional supports each step in operations management for distributed, interconnected applications as follows:

2.1 AUTOMATIC SERVICE DISCOVERY

Automatic service discovery gives an unprecedented level of visibility into a distributed, interconnected system. For each instrumented SOA component, i.e., a component such as an application server with an agent installed, all service-oriented message traffic is monitored. From the messages, the agents can discover the existence of each service used on or invoked from an instrumented component. Requests and responses directly between instrumented and non-instrumented components are also monitored, allowing for services to be discovered even on non-instrumented servers.

Services can be of many different kinds—not only Web services. For example, Figure 3 shows the visualization in Actional corresponding to Figure 2. All of these services and relationships are automatically discovered by the Actional agent and then communicated to the server. No programming, application modification, or pre-discovery configuration of the services in Actional is required.

Figure 3: Automatic Service Discovery and Dependencies



The discovered dependencies allow impact analysis of proposed changes or of service-level problems. Service discovery will also bring to light "rogue services." Rogue services within a service-based application are mismanaged or lost Web services that cause problems for unknowing IT managers who might be using them. A rogue service that does not comply with business policies and/or compliance mandates can result in costly audits or, in the worst case, harmful security breaches.

2.2 NON-INTRUSIVE RUNTIME MONITORING

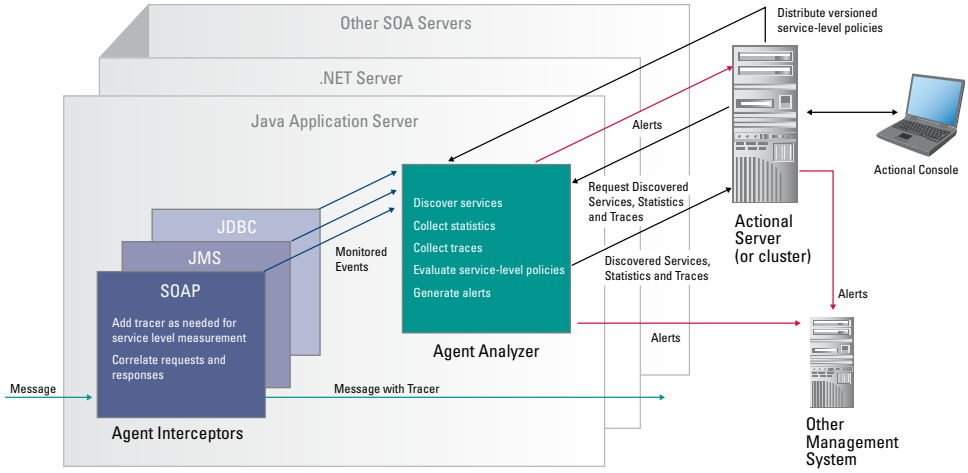
A key strength of Actional is in its runtime monitoring of components. These components can include application servers, database systems, XML network appliances, enterprise service buses (e.g., Progress® Sonic ESB®[5]), and application integration platforms (e.g., SAP NetWeaver). The monitoring of each of these components is done in a way that is completely non-intrusive to the monitored component and the applications that use it.

Monitoring is done in three stages, as shown in Figure 4. The Actional agent on an individual network node consists of interceptors for each component being monitored and an analyzer. The interceptor is written using the native "plug-in" capability for the specific component being monitored. The interceptor copies "on-the-fly" messages or function calls (such as for JDBC database access) to an in-memory buffer the minimum data it needs to monitor the component. The elapsed time to perform an intercept is typically in the 10's of microseconds. The analyzer processes the data collected by the interceptors to discover services, test for service-level policy violations, and collect aggregate statistics. Policy violations are communicated immediately to the Actional server as alerts while aggregate statistics are communicated to the Actional server on a periodic, user-specified basis every few minutes.

Transaction tracing is also enabled by the Actional agent. For transactions that are subject to policy enforcement, a non-impacting "tracer" is added by the interceptor to the transaction request. This allows subsequent tracing of the transaction flow through the rest of the network. Tracing information is collected by the interceptors, correlated as requests and responses, and then communicated to the server to allow transaction tracing across components.

[5]Sonic ESB: An Architecture and Lifecycle Definition, Progress Software Corporation, 2006, http://www.sonicsoftware.com/products/sonic_esb/architecture_definition.

Figure 4: Non-intrusive Monitoring, Policy Management, and Alerting



SOA Network nodes wih Actional agents

2.3 POLICY MANAGEMENT

As shown in Figure 4, Actional implements a "best of both worlds" strategy of centralized policy management combined with distributed event monitoring and policy evaluation. This allows for management of policy across a services network from a single console while limiting network traffic between the agents and the central server and provides an architecture that has no single point of failure. Even if communications are temporarily lost between the Actional agent and server, the agent continues to collect data while monitoring and evaluating policies. In addition, the Actional server can be clustered for high availability, and alerts to other connected management systems can be sent from either the agent or the server.

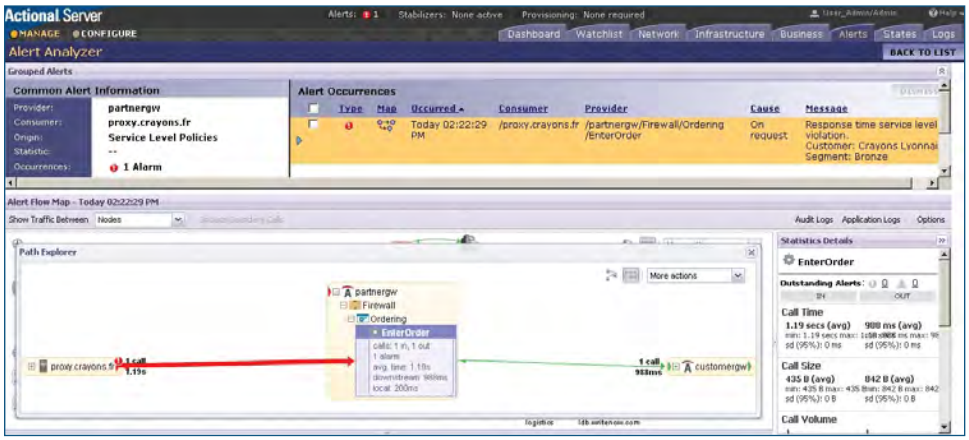
Service-level policies are created with easy-to-use wizards in the Actional server console. Policies can be set to look at a variety of behaviors such as response time, data throughput, and frequency of faults. Policies can specify required service levels on either a per-transaction (e.g., maximum response time) or aggregate statistic (e.g., average response time) basis. Policies are then "pushed" from the server to each agent. Each policy is versioned for easy tracking across each network component and for roll-back, if needed.

2.4 POLICY-BASED ALERTING

When a service-level policy is not met, it is critical that IT staff is notified to resolve the problem. The staff notified could be operations staff, applications support staff, or both. The Actional agent sends alerts to the server, which are displayed in the console and can be sent by email to support staff or to common systems management products such as to HP OpenView (by SNMP).

Alerts show up in the Actional server console in the Alert Analyzer, as shown in Figure 5. This image shows the case of a response time target that was not met. The Alert Analyzer shows the cause of the alert, a message sent with the alert, and an alert flow map. For individual transactions that violate a policy, the flow map shows the end-to-end flow of the transaction for use in problem determination.

Figure 5: Actional Alert Analyzer



2.5 PROBLEM RESOLUTION

When an alert notification is received telling of a service-level target that was missed or is in danger of being missed, the operations or application support staff uses Actional to pinpoint the source of the problem. A key to problem resolution with Actional is to use the service network visualization and/or the alert flow map to pinpoint the source of the problem.

For example, the alert flow map in Figure 5 shows a call from an external customer to a customer gateway application server in which the 7 second response time was exceeded. The response time shown of 11.4 seconds is traced through the gateway

(taking .2 seconds) to the EnterOrder operation in the Ordering service, which took 11.2 seconds. There were no additional service requests from the EnterOrder operation, so this is the source of the slow response—and the target of additional research and then remediation.

Actional supports each of the capabilities outlined above for systems that are:

> ***Heterogeneous***—for multiple types of components, application servers, databases, appliances, ESBs, and application integration platforms supporting multiple technology platforms including Java EE, .NET, and open source.

> ***Multi-protocol***—including protocols and interfaces such as Axis, SOAP, REST, POX, EJB, JMS, Servlets (HTTP), Jakarta, HTTP client, ADO.Net, RMI, and JDBC.

An operations management system for distributed, interconnected applications is only "enterprise-ready" if it has a distributed architecture itself like Actional's, keeping policy enforcement at the point of no-impact monitoring, so that it is:

> ***Usable***—The Actional server provides centralized, service-level policy definition, distribution, and roll-back across a distributed network as well as alert management and problem determination.

> ***Practical***—Agents are architected to cause virtually no impact on monitored components while at the same time able to autonomously evaluate service-level policy and perform transaction tracing.

> ***Scalable***—A single Actional server can manage tens of thousands of services on 1,000+ managed systems.

> ***Reliable***—The network of Actional servers and agents has no single point of failure.

3.0 ADVANCING DISTRIBUTED, INTERCONNECTED APPLICATIONS WITH ACTIONAL

Distributed, interconnected approaches to computing provide the impetus for the development of applications designed and deployed as service components. Managing the operations of such an infrastructure requires the ability to perform each of the following functions:

Discover > Monitor > Evaluate Policy > Alert > Resolve

Actional provides these functions in a manner that supports best practices in meeting and exceeding target for key performance indicators (KPIs) such as:

> Percentage reduction in SLA targets missed

> Percentage reduction in the MTBF (mean time between failures)

> Percentage reduction in MTTR (mean time to resolution)

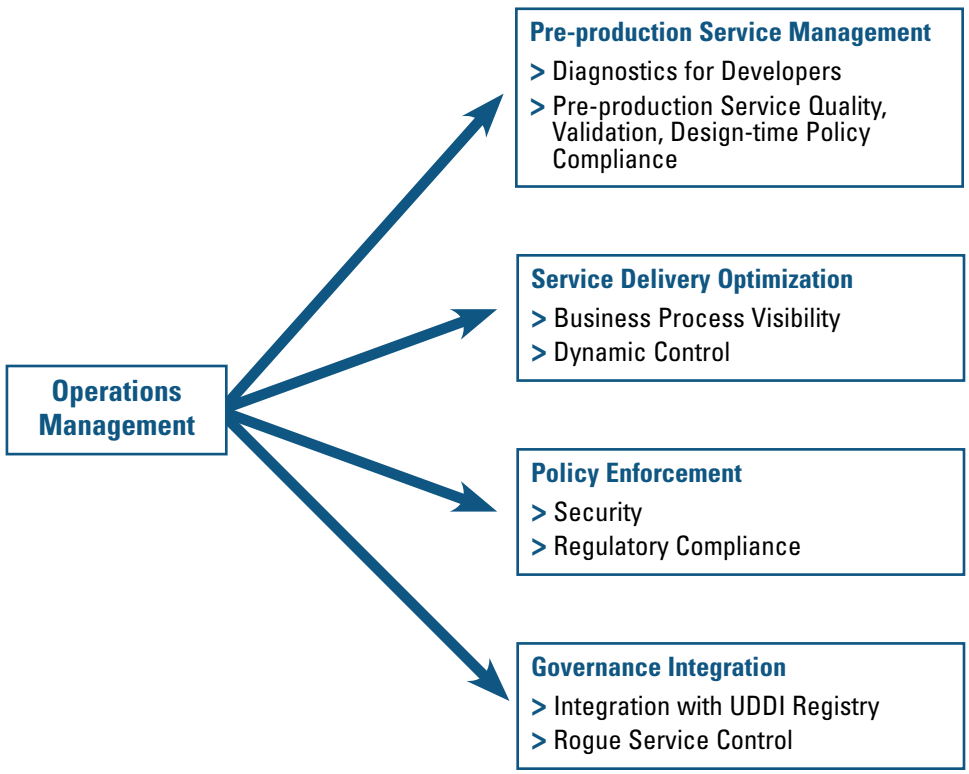
> Percentage reduction in service delivery costs

> Percentage increase in incidents fixed before users notice

Actional allows for the use of distributed, interconnected applications by an enterprise in a manner which maintains performance and availability of a heterogeneous infrastructure while meeting and exceeding operational KPIs. These benefits are valuable from the development and QA stages of services development through integration and production. Using Actional early in the services lifecycle helps assure that KPIs will be met in production.

Actional provides capabilities beyond the core services operations management described so far in this paper. As shown in Figure 6, additional capabilities built into Actional Enterprise provide other important management capabilities as well.

Figure 6: Actional Capabilities beyond Operations Management



- Actional pre-production tools help application pre-production teams, from developers, testers, and QA personnel to architects and analysts, simplify XML-oriented tasks to more quickly build, test, validate, and deliver high-quality Web services.
- Actional tools for continuous service optimization enable IT to align operations with business needs by ensuring quality of service for customers and other end users. It provides both business insight into distributed, service-based operations for making and prioritizing decisions and integrated runtime controls for continuously optimizing business outcomes.
- Actional tools for policy enforcement provide centralized creation and management of business (e.g., SLAs), security, and compliance policies, while ensuring distributed policy enforcement. Actional empowers experts to author policies once, apply them consistently across the services network, and guarantee complete coverage while reducing cost and risk.
- Actional Governance Integration Module integrates with third-party UDDI governance tools, such as Oracle AquaLogic Enterprise Registry Repository (formerly BEA ALER), Software AG's CentraSite Governance Edition, Fujitsu's and Software AG's CentraSite Enterprise Edition, and HP Systinet. With this integration it can control rogue services, pending review and approval; share policy information; and upload performance statistics and dependencies collected by Actional into the registry or repository.

For more information on how Progress Actional sets the standard for services management, see www.progress.com/actional, call +1 781 999 7100 (+44 (0) 1753 217001 in Europe), or send email to eval@progress.com (info-emea@progress.com in Europe).

PROGRESS SOFTWARE

Worldwide Headquarters
Progress Software Corporation, 14 Oak Park, Bedford, MA 01730 USA
Tel: +1 781 280-4000 Fax: +1 781 280-4095
www.progress.com

For regional international office locations and contact information, please refer to www.progress.com/worldwide