Service Virtualization

A faster, more efficient and less costly way to develop and test enterprise-class applications

As cloud and mobile computing gain rapid acceptance, IT departments are expected to provide fast and comprehensive testing of composite applications. By deploying advanced service virtualization, organizations can reduce development time and cost, while ensuring optimum quality from those crucial applications.
Introduction

Organizations across the business and industrial spectrum now recognize the value of as-a-service capabilities.

Service-based software and composite applications deliver significant advantages – giving companies access to new and robust tools, supporting reuse and other efficiencies, and allowing IT units to more quickly and effectively meet the needs of demanding business users.

Yet service-driven technologies also pose a new and unique set of challenges. They are by nature shared and interdependent. When provided by third-party vendors, they may pose security and access concerns. Quality Assurance units often struggle to fully test modern enterprise applications when those services are under development, dedicated to mission-critical production activities, or otherwise unavailable.

In this Orasi white paper, we explore the use of Service Virtualization to address crucial application development and testing requirements in a service-oriented environment. The authors examine current trends in enterprise-class applications, issues related to testing composite apps, and accepted best practices for advanced Service Virtualization.

The paper then reviews the benefits of the virtualization approach, and describes how Orasi can help companies leverage HP Service Virtualization in the enterprise environment.

To better understand the full capabilities of these next-generation mobile application testing technologies, it may help to examine more closely the emerging capabilities of as-a-service cloud-based testing.

Development and testing trends

Composite applications and service-based software systems are quickly becoming key elements in many enterprise IT environments.

Organizations can derive measurable value from composite and service-based solutions, because they support the decoupling of technologies into smaller and more nimble functional units. This key quality allows organizations to reuse functionality, to create and integrate functionality more easily, and to more quickly meet the needs of business users.

Service-oriented and composite applications are typically assembled from existing services, and often span across units within an organization, or even across multiple organizations or geographic regions. Services can be web services, RESTful, messaging, or custom services, and are typically headless with no front-end or user interface, and operate on a machine-to-machine basis.

While Service Oriented Architectures (SOA) and composite services are beneficial, they also present special challenges. Interdependencies among functional elements can create costly delays in development or quality assurance programs, and can affect on-time delivery of IT projects. The differing security and access standards in composite applications add to these lifecycle challenges.

Figure 1: Modern applications present availability, cost, and reliability challenges.
Third-party services typically come with additional per-transaction fees, and may impose limitations on load testing, high-volume functional and performance testing. Third-party services may also be unstable, and are subject to access and change issues when shared with other users.

When a composite service is under development, documentation may be limited to design specifications, and the service itself may offer limited functionality due to implementation schedules or yet-to-be addressed defects. Shared services may offer access only during off-peak hours, typically at night for business-critical systems.

In other situations, the full implementation of a service may not yet exist, or the service may be a component of a production system and thus not available for normal testing.

How SOA quality is handled today

In a service-oriented architecture, quality is directly affected by service availability, cost, and volatility.

Availability is obviously limited during periods of service development, when Quality Assurance does not have the required authentication or authorization permissions, or when they are dedicated to other uses. Functional and performance testing also can have a major impact on the availability of mission-critical services, and for this reason, organizations are working to deploy testing programs that are more time- and resource-efficient.

The cost of a service can be affected by internal expenses, as well as by third-party transaction and licensing costs.

Service quality can be impacted significantly, and negatively, by unexpected changes that may affect performance, functionality, and end user satisfaction. When IT units cannot anticipate or control changes, the organization often experiences inconsistency, delays, and frustration with service delivery and performance.

Those three crucial variables – availability, cost, and volatility – can and do dramatically impact the timeline and budget of a SOA quality assurance effort.

Unfortunately, the traditional manual-intensive approach to SOA quality presents a number of challenges. Most organizations currently utilize techniques such as stubbing, mocking, or emulating to test in the development environment.

Service stubbing allows developers to test components and unit elements of an application before the entire application is completed. Mock objects are typically simplified versions of more complicated or costly resources that allow testers to analyze component resources without relying on the external resource. In the SOA environment, service emulation is an automated, script-based approach that allows testers to evaluate function and performance before services are actually built.

Each of those methods requires at least a certain level of manual input and development skills.

Often, the personnel with the needed development expertise are busy with other activities, or the team does not have ready access to the resources necessary to handle stubs. Stubbing typically makes sense only in situations where the organization has reliable access to the service developer. Thus it is not practical, and often not possible, to manually stub third-party services.

Manual efforts are often hard to maintain, and can typically only be used to cover basic functionality and very simple services. They are not appropriate for handling data or business rules, and do not provide support for service performance and scalability simulations, or for negative tests based on service behavior. While manual methods can in some instances be used to address more complex services or messaging, it can be difficult and time consuming.

The good news is: by leveraging today’s more advanced virtualization technologies, organizations can now utilize a more effective method to develop and test composite applications.

The Service Virtualization Approach

Organizations can now access an emerging generation of more advanced Service Virtualization solutions to better address quality challenges early in the application lifecycle.

By simulating unavailable services in a virtual environment, these technologies allow IT and QA units to address the challenges of developing and testing composite applications. This virtualized approach can be used to perform functional, performance and load testing, integration, and end-to-end application testing. It enables testing earlier in the application lifecycle, and when the actual services themselves have limited or no availability.

Today’s more advanced Service Virtualization solutions address the limitations of previous-generation manual techniques, and can be deployed to reduce the time and cost of developing and testing current-generation applications. With virtualization, organizations can eliminate third party transaction and licensing fees.
Virtualization also relieves the classic pressure points in the application lifecycle, freeing developers to develop and allowing QA to focus on testing. Unavailable services can be accessed in the virtual environment. Limited or constrained services can be fully available, and organizations can now evaluate in a comprehensive way applications that previously could not be fully tested. QA units can leverage Service Virtualization to undertake end-to-end functional and performance testing.

This approach leverages several unique techniques and capabilities – including visual design, learning, service description and sample messaging – to automatically create virtual services. Virtualization also can be used to address data and performance modeling, and scalability for performance testing.

Learning
Virtual services can be created via the learning method, whereby the solution listens on the wire for existing services, then learns the functionality, data, and performance specifics of that application. The solution then creates a virtual service based on that recorded traffic.

Description
Organizations can automatically create virtual services by using descriptions derived from contracts, documentation, sample messages or Web Service Definition Language (WSDL). The description method is appropriate for creating virtual services when existing services are non-existent, or when they have limited availability.

Visual
Virtual services can be created using virtual design tools. In this approach, all functionality is exposed via the design tool, and intuitive guides and tips help accelerate the process. The visual designer method provides very usable virtual services, and can measurably reduce the time needed to produce an initial virtual service.

Data
Modern Service Virtualization solutions can represent learned or imported data. Data can be pulled from external sources, much like “data driving,” but performed on the back-end of the process. QA units also can apply rules to the data, as needed.

Performance Modeling
Once a service is learned, testers can specify exactly how they want the learned service to perform. This is accomplished through a rich GUI.

Today’s robust Service Virtualization solutions support a broad spectrum of technologies, including Java Message Service (JMS), Simple Object Access Protocol (SOAP), Representational State Transfer (REST), Tibco, WebSphere MQ, Windows Communication Foundation (WCF), and Web Service Security (WSS).
Companies and organizations can utilize this emerging generation of Service Virtualization to accomplish a wide range of testing-related objectives.

At its most basic, the approach can be deployed to expose services for parallel development and early functional testing. Simulation models can be exposed as a live service, while ensuring safe access and authentication for testing purposes.

Both data and performance models are easily sustained to reflect any changes in test conditions or operational requirements. Organizations moving to cloud or mobile platforms can leverage virtualization to simulate how new and existing applications will behave in those environments.

Powerful tools deliver new and extended test and development capabilities.

Step-by-step wizards can be used to create functional and performance simulation models. Topology diagrams can be defined and visualized to provide greater insight into boundaries and dependencies in underlying systems, down to the level of remote API calls. Working from the virtualized topology, testers can trigger the creation of functional and performance simulation models. Simulation models also can be stored and invoked from ALM components, thus simplifying the setup and execution of test environments and the analysis of application test results.

Benefits of Service Virtualization

HP Service Virtualization software allows development and testing project teams to access limited or unavailable services in a simulated, virtual environment. Orasi partners with HP to offer services to install, implement, configure, and use the HP Service Virtualization software.

As they move toward cloud and mobile technologies, organizations can leverage HP Service Virtualization to realize measurable strategic advantages.

- Simplify testing and development, by deploying advanced systems that are easy to set up and simple to configure and use. Today’s robust service virtualization technologies provide robust performance and data modeling, and seamless integration with existing HP Application Lifecycle Management infrastructure.
- Enhance application quality and performance by supporting full testing of entire SOA applications earlier in the lifecycle – without waiting for unfinished or undeveloped services. Virtualization also provides broader development protocol support for testing legacy, cloud and mobile applications.
- Reduce the cost of running and managing modern test environments by simulating access to unavailable third-party services for application development or testing. Virtualization reduces the need for hardware and other infrastructure, software licenses, and system maintenance.
- Accelerate processing times and release cycles by giving QA teams 24x365 access to customized virtual services and the reuse of stored services for development and testing of applications. Virtualization reduces wait times and supports time-efficient parallel development and early-cycle functional testing.
- Drive productivity by eliminating the need for QA and development teams to create rough “mock” services. Virtualization provides services that represent performance and functionality of missing or unavailable services, allowing test teams to focus on more crucial, value-building activities.
- Reduce risk by allowing testing earlier in the cycle – thus giving QA teams the ability to identify and correct potential defects when they are easier and less costly to fix. Service virtualization also reduces risk by supporting broader test coverage, and by allowing engineers to fully understand back-end issues and performance before a system is deployed.
Conclusion

Composite and service-oriented applications offer clear cost, access, and productivity advantages.

In a fast-paced development and production environment, however, those as-a-service solutions also pose unique and difficult challenges. As we have seen, these enterprise-class applications may present integration and security issues, additional third-party costs, and access problems that can add time and cost to critical development and testing efforts.

For many quality-oriented organizations, Service Virtualization may be the answer. Virtualization allows QA teams to quickly and efficiently create complete and accurate systems to do end-to-end functional, performance, integration, and load testing. This field-proven approach can yield measurable benefits – reducing development cost and risk, accelerating release schedules, and improving the overall quality of critical enterprise applications.

Orasi has partnered with HP to offer implementation, configuration, and mentoring services to support the HP Service Virtualization platform. By fully understanding the requirements and benefits of virtualization, organizations can optimize access, quality, and value in those mission-critical systems.

About Orasi

An HP Software Specialist Enhanced Partner and authorized Support partner, Orasi resells HP’s test management and automation solutions, is a leading provider of software testing services, and offers mobile testing, security, and cloud-based testing and monitoring solutions. For over 10 years, Orasi has consistently helped customers successfully implement and integrate software testing environments to reduce the cost and risk of software failures. Orasi was HP Software’s 2011 Partner of the Year (US) and was Support Partner of the Year for 2009, 2011 and 2012.

Orasi offers proven solutions for the installation, implementation, configuration, and use of HP Service Virtualization.