

# Effective Management of SOA Applications with Semantic Modeling

*An Oracle White Paper*  
*November 2008*

# Effective Management of SOA Applications with Semantic Modeling

Executive Overview .....	3
Introduction .....	3
A Closer Look at SOA Applications using Process Orchestration .....	3
Performance Management of Oracle SOA Suite.....	5
Managing Oracle SOA Suite Applications with Oracle Composite Application Monitor and Modeler .....	7
Intelligent Metric Mapping Removes Dependence on Experts.....	8
Rapid Results and Significant Cost Savings .....	8
Improve Insight with Workflow Visualization.....	9
Built-In Delay Analysis Isolates Trouble Spots.....	9
Conclusion.....	10

# Effective Management of Enterprise Integration Applications with Semantic Modeling

**With the continuing evolution of composite applications, connecting application silos has become a key objective of enterprise IT strategy. To achieve this objective, many large IT organizations have implemented complex integration solutions.**

## EXECUTIVE OVERVIEW

This white paper focuses on the challenges and solutions associated with managing integration applications that use process orchestration. As a commercial example, it highlights the Oracle Composite Application Monitor and Modeler application management solution from Oracle, which uses semantic modeling to manage integration applications running on Oracle SOA Suite and Oracle WebLogic Integration.

## INTRODUCTION

With the continuing evolution of composite applications, connecting application silos has become a key objective of enterprise IT strategy. To achieve this objective, many large IT organizations have implemented complex integration solutions, ranging from enterprise application integration to business process management, to Enterprise Service Bus platforms. These technology sets form an enterprise integration infrastructure supporting the creation of distributed applications that can deliver orchestration of complex end-to-end business processes and services.

These integration environments enable new complex business applications and services to be developed in a much shorter time and with fewer resources. Because of the nature of process orchestration, these applications tend to run the business processes that are the most critical to the enterprise, usually requiring reliable support for numerous users—and *always* with hefty service-level requirements.

Increasingly, the productivity and financial benefits delivered by these applications are making them an indispensable part of their owners' day-to-day business operations. These mission-critical applications handle complex business processes, ranging from processes that orchestrate financial transactions to processes that prepare insurance quotes. Maintaining high application availability and performance is vital to the owners of these applications. Unfortunately, managing this class of applications is no simple task.

## A CLOSER LOOK AT SOA APPLICATIONS USING PROCESS ORCHESTRATION

Understanding the challenges of managing distributed integration applications that use process orchestration means first understanding the complexity associated with

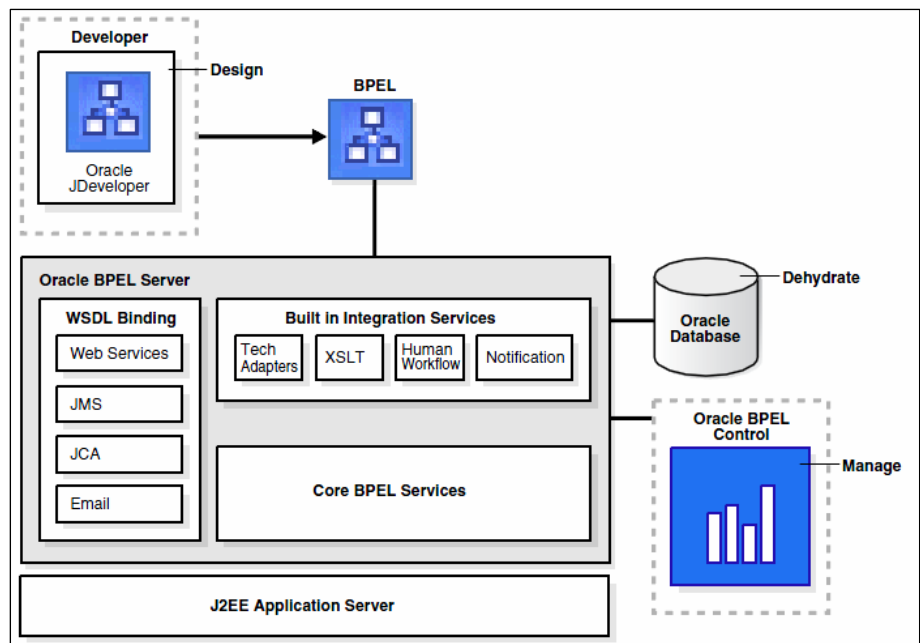
these applications. This section explores these challenges by explaining the typical lifecycle of integration applications running on Oracle SOA Suite.

Oracle SOA Suite provides a set of runtime frameworks; standardized BPEL workflow technologies; Enterprise Service Bus; and Java 2 Platform, Enterprise Edition (J2EE) components that enable the platform to effectively integrate multiple disparate applications into a seamless whole. Oracle SOA Suite has proven effective as a basis for massive enterprisewide integration projects.

The vast majority of Oracle SOA Suite applications are created by use of Oracle JDeveloper, Oracle’s integrated development environment for the Oracle SOA Suite platform. Within Oracle JDeveloper, business processes are defined by use of high-level artifacts (BPEL) that represent a collection of discrete business tasks. This layer of abstraction hides the low-level code execution complexity, enabling application architects and engineers to focus on the application’s business goals instead of on intersystem interactions and low-level component behavior.

A primary function of Oracle SOA Suite is business process orchestration via standardized BPEL, which serves to coordinate disparate events and subprocesses that make up a composite application or business function. But before Oracle SOA Suite can coordinate or “orchestrate,” it must first connect the various applications. As noted in Figure 1, Oracle SOA Suite supports the required connectivity for data and applications via several methods—including use of J2EE Connector Architecture (JCA) adapters, Web services, e-mail, and Java Message Service (JMS).

**A primary function of Oracle SOA Suite is business process orchestration via BPEL, which serves to coordinate disparate events and subprocesses that make up a composite application or business function.**



**Figure 1: Oracle SOA Suite supports the required connectivity for data and applications with multiple methods.**

Business process orchestration and object transformation in Oracle SOA Suite are driven by events. The events may be timer-based; data-driven; or initiated explicitly

by external applications, Web services requests, or internal business services. Once an initiating event occurs, various Oracle SOA Suite subcomponents acquire the needed dependent data, transform the data into normal formats, and execute the business process workflow, using a combination of local method and remote service invocations.<sup>1</sup> In effect, work flows through Oracle SOA Suite via many-to-many transactional orchestrations.

One of the most challenging aspects of SOA application management is the need to track the multiple intermediate states involved as events are serviced. Most computer architects are aware that the growth of connectivity interaction (static complexity) is roughly equal to the square of the connected nodes.<sup>2</sup> Less well known is that the growth of intermediate state information (dynamic complexity) has been shown to be equal to the cube of the number of connected nodes.<sup>3</sup> Oracle SOA Suite manages intermediate state information automatically in persistent containers, freeing the members of the development team from responsibility for one of the most error-prone and time-consuming coding tasks they must complete.

**Oracle SOA Suite was designed to support your applications in runtime, not to manage every aspect of performance and availability over their entire lifecycle.**

However, Oracle SOA Suite was designed to support your applications in runtime, not to manage every aspect of performance and availability over their entire lifecycle. Technologies designed to manage prior generations of applications cannot cope with the new complexity. For that, you need a management solution designed from the ground up to manage and monitor the new class of enterprise applications in production.

## **PERFORMANCE MANAGEMENT OF ORACLE SOA SUITE**

The applications being deployed on Oracle SOA Suite are designed to deliver essential enterprise services and business functions. These mission-critical applications are too important to simply “try your best” to keep them running well. Poor performance during peak load periods can cost upwards of millions of dollars in unrecoverable lost business. Neither simple “on or off” status nor alerts of system crashes will be enough to help in complex environments. The management requirements can be met only by solutions as advanced as the Oracle SOA Suite runtime architecture itself.

During development, the new class of Oracle SOA Suite applications is completely visible to the design team. All of the upper-level business artifacts, processes, workflows, and the application hierarchy can be clearly seen in Oracle JDeveloper. Unlike with prior generations of business process modeling tools, the graphic representations in Oracle JDeveloper are directly compiled into complete

---

<sup>1</sup> See the Oracle SOA Suite descriptions at [oracle.com/technologies/soa/soa-suite.html](http://oracle.com/technologies/soa/soa-suite.html).

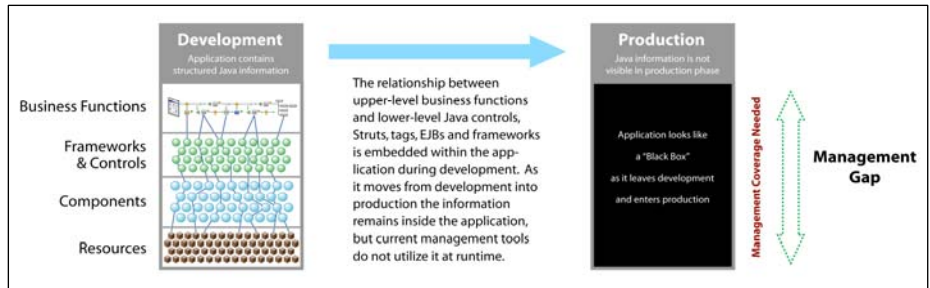
<sup>2</sup> “Metcalf’s Law: Connections Grow at the Rate of  $N^2-N$ .” See [http://www.cemisid.ing.ula.ve/index\\_e.html](http://www.cemisid.ing.ula.ve/index_e.html).

<sup>3</sup> “N-Cubed Phenomenon in Connectivity.” Michael Frank, PhD. University of Florida College of Computer Science.

applications, which carry within them all the information that was seen at design time.

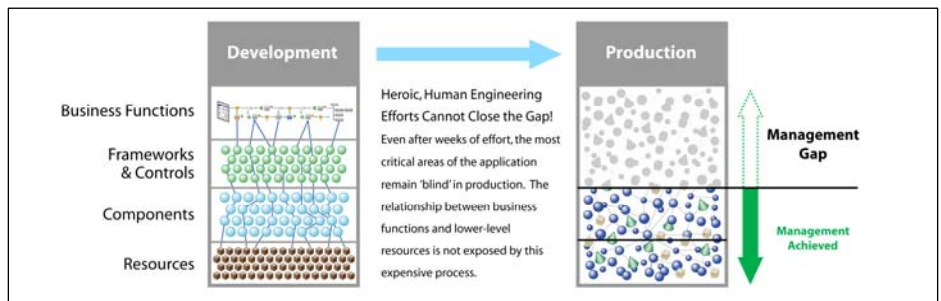
**Operations teams end up taking ownership of “black boxes” when they accept an Oracle SOA Suite Integration application for production, because they cannot use traditional operations tools to help monitor and manage application performance.**

Unfortunately, as applications migrate from development through testing, staging, and production, this visibility is lost. Operations teams end up taking ownership of “black boxes” when they accept an Oracle SOA Suite application for production, because they cannot use traditional operations tools to help monitor and manage application performance. To compensate, they use a combination of application-specific support personnel (such as senior developers or architects) and code-based performance tools to attempt to fill the remaining management gap. The results leave much to be desired (see Figure 2).



**Figure 2: When operations accept an Oracle SOA Suite application, it is a “black box” with vital but inaccessible information for performance management.**

Conventional application monitoring tools were designed to measure the performance of low-level application building blocks and resources. As is often the case with older technologies, their prior strength is now a weakness, in that the architecture and runtime environments have evolved to support applications that are more dynamic and complex. The high costs associated with the labor-intensive stopgap that closes only part of the gap have contributed to the slow adoption of these complex composite applications (see Figure 3).



**Figure 3: The costly stopgap solution still leaves operations “blind” to critical areas of the application.**

As described above, application complexity has grown exponentially in a single generation. Five years ago, seeing a list of all the Java components in a running application helped IT solve issues in single-server, single-application environments. Over time, this approach became cumbersome, given that a single application could yield well over 10,000 components and the management tools included no contextual information about how any individual component related to the whole

application. Effective management in today's complex environments requires the ability to sift through the 10,000+ components and the 1,000,000+ interactions to organize these relationships into a useful hierarchy and then to manage the applications from the top down.

## **MANAGING ORACLE SOA SUITE APPLICATIONS WITH ORACLE COMPOSITE APPLICATION MONITOR AND MODELER**

**Because Oracle Composite Application Monitor and Modeler is built on top of an application modeling engine, it alone has the ability to understand the program-level context for relationships among Oracle SOA Suite application components.**

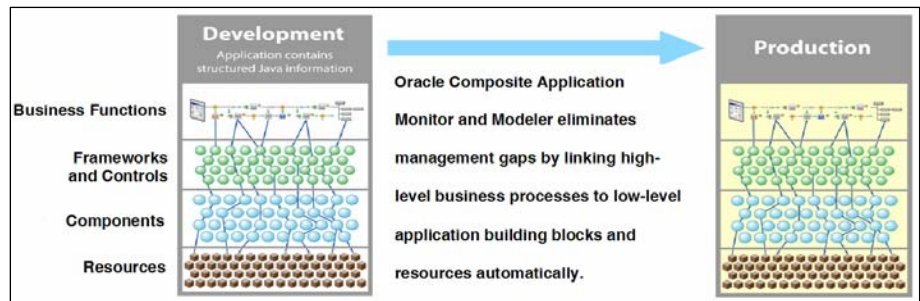
Oracle Composite Application Monitor and Modeler is a next-generation application management solution designed specifically for this new class of distributed applications. Because Oracle Composite Application Monitor and Modeler is built on top of an application modeling engine, it alone has the ability to understand the program-level context for relationships among Oracle SOA Suite application components (application-to-application relationships and dynamic component relationships within each application). Interestingly, the knowledgebase necessary to create the context is derived from more than just monitoring internal application code—it is also derived from harvesting information jammed into the embedded metadata and associated log files. With the abundance of data Oracle Composite Application Monitor and Modeler derives, the “black box” built with so many little black boxes can be seen.

Oracle Composite Application Monitor and Modeler automatically discovers the entire deployment environment—including Oracle SOA Suite servers, clusters, and applications. It then harvests information about these applications by analyzing the Java bytecode and the Oracle SOA Suite metadata and log files—looking for design-time artifacts. For Oracle SOA Suite applications, Oracle Composite Application Monitor and Modeler maps high-level business processes (BPEL) with low-level J2EE application components' relationships, correlating the layers automatically. This intelligent application blueprint, stored in the drill-down application model, is used during all phases of performance management—from continuous service-level monitoring to diagnostic root cause analysis.

For each deployed and running Oracle SOA Suite application, Oracle Composite Application Monitor and Modeler uses the application model to automatically determine optimal locations in which to insert tracers—timers and counter probes for taking performance measurements. The specialized tracer data is combined with other low-level component metrics, Java Management Extensions (JMX) metrics, and system metrics—keeping all data within the context of business processes—to ensure complete overall management.

If the application displays performance anomalies or unusual error conditions, IT personnel can immediately switch to Oracle Composite Application Monitor and Modeler's problem diagnostic views to analyze the problems. Specialized views for both monitoring and diagnosing problems are generated automatically, based on the results of both static (up-front) and dynamic (runtime) analysis. As a result, Oracle Composite Application Monitor and Modeler gives IT organizations a complete performance management solution that provides 100 percent visibility,

monitoring, and performance management for the new complex enterprise applications deployed on Oracle SOA Suite. This solution closes the management gap and eliminates mystery performance problems and downtime surprises.



**Figure 4: Oracle Composite Application Monitor and Modeler enables 100 percent performance management of Oracle SOA Suite applications.**

### Intelligent Metric Mapping Removes Dependence on Experts

Low-level performance measurements (and the ability to retrieve them) have achieved commodity status and are of minimal value in complex distributed environments. To extract additional value from these sets of measurement data, IT organizations must first understand how individual metrics map to various application functions and business processes, or to both. With traditional tools, the only way to get this correlation is to depend on application experts—application developers and architects—who can reverse-engineer the managed applications in hopes of identifying the business context for each low-level component and its associated metrics. As discussed above, though, the exponentially increasing complexity of distributed applications makes this approach untenable. Although it might be worth investing in a manual process of mapping measurements to a single application, repeating the process for multiple applications—and redoing the analysis when changes occur—is not scalable in enterprise environments.

Oracle Composite Application Monitor and Modeler differs from conventional tools, because its model-driven approach automatically maps performance measurements to applications, enabling users to extract value from these measurements almost immediately. As a result, Oracle Composite Application Monitor and Modeler users can identify and solve performance problems faster and easier than they can with conventional tools.

**Oracle Composite Application Monitor and Modeler differs from conventional tools, because its model-driven approach automatically maps performance measurements to applications, enabling users to extract value from these measurements almost immediately.**

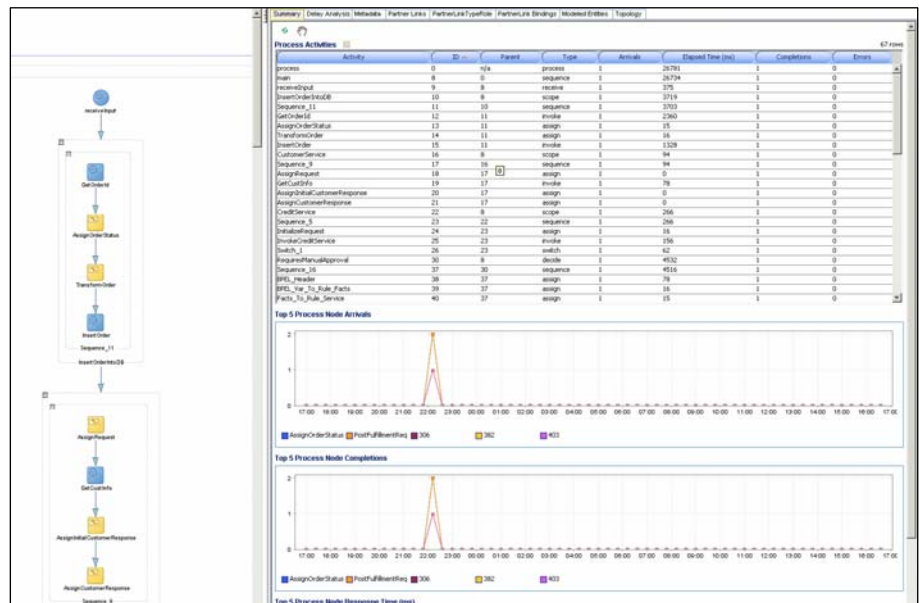
### Rapid Results and Significant Cost Savings

Unlike conventional solutions that require weeks of reverse engineering and manual customization, Oracle Composite Application Monitor and Modeler accomplishes all of the above programmatically, drastically reducing the time to value. By automating a set of complex tasks currently done by teams of engineers, Oracle Composite Application Monitor and Modeler has the potential to save organizations hundreds of thousands of dollars in the first 90 days of operation alone. Development teams can focus on development, not application support, and the enterprise can deploy far more applications in a given time than before.

**Oracle Composite Application Monitor and Modeler also has the unique ability to visually represent the logical flow of Oracle SOA Suite business processes (BPEL).**

## Improve Insight with Workflow Visualization

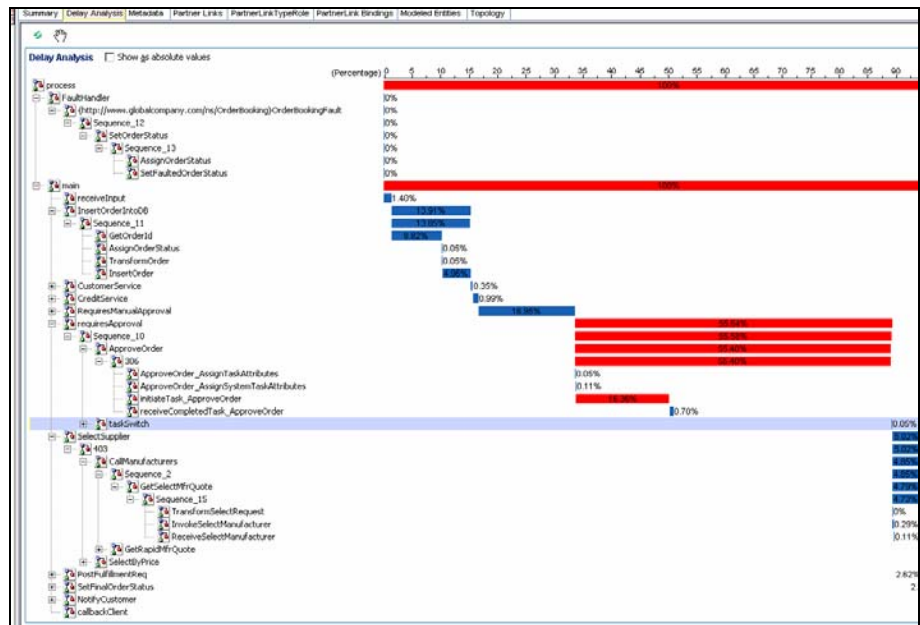
Oracle Composite Application Monitor and Modeler also has the unique ability to visually represent the logical flow of Oracle SOA Suite business processes (BPEL). This capability gives those who lack detailed application knowledge the opportunity to explore, learn, and understand the design of these Oracle SOA Suite applications. Additionally, Oracle Composite Application Monitor and Modeler presents performance measurements alongside the workflow diagram. When users click on a specific process node (or group), the associated performance measurements show up in the right-hand pane.



**Figure 5: Oracle Composite Application Monitor and Modeler shows the workflow for Oracle SOA Suite business processes (BPEL) such as OrderBooking (shown here) and the associated performance measurements.**

## Built-In Delay Analysis Isolates Trouble Spots

When problems occur, the application model again comes to the forefront. Oracle Composite Application Monitor and Modeler uses it to access information about individual Oracle SOA Suite applications and quickly guides IT personnel directly to performance trouble spots. For example, by comparing the role every node in a distributed business process has in existing delays, Oracle Composite Application Monitor and Modeler's built-in delay analysis enables any operations team member—regardless of that person's specific Oracle SOA Suite application knowledge level—to pinpoint the slowest process node in seconds. This is a significant improvement over conventional tools, because users no longer need to export raw data to external tools for this type of analysis.



**Figure 6: Oracle Composite Application Monitor and Modeler's Delay Analysis for the Oracle SOA Suite business process (BPEL) OrderBooking, showing results for the complete process and individual nodes**

Once the problematic process node has been identified, the Oracle Composite Application Monitor and Modeler user can drill down to the application level, utilizing Oracle Composite Application Monitor and Modeler's unique Architecture Views to further diagnose the problem. Diagnosing performance problems by using Oracle Composite Application Monitor and Modeler is far faster and easier than with traditional application performance management (APM) tools, because Oracle Composite Application Monitor and Modeler uses the information stored in its semantic model to organize performance measurements in a logical fashion.

As Gartner's Research vice president Milind Govekar best put it, "using a modeling approach is one of the key ways to manage applications to ensure consistent and predictable performance."

**"Using a modeling approach is one of the key ways to manage applications to ensure consistent and predictable performance."**

**—Milind Govekar,  
Research Vice President  
Gartner**

## CONCLUSION

Today's IT organizations are increasingly adopting SOA technologies that enable them to quickly connect disparate applications and fulfill ever-changing business needs. Although these new SOA applications offer unprecedented flexibility and complexity—with features such as process orchestration—this level of sophistication now makes them difficult to manage. Conventional tools that rely on manual analysis and correlation to organize performance measurements cannot be scaled appropriately in these distributed application environments.

**One of the best ways to deal with this level of fluidity and complexity is to use solutions that harness semantic modeling.**

To effectively manage the performance of this new breed of applications, IT organizations need a new breed of APM solutions. One of the best ways to deal with this level of fluidity and complexity is to use solutions that harness semantic modeling to capture the interdependencies among application building blocks and to create a map that describes the structure of these applications. Oracle Composite Application Monitor and Modeler is one such solution that manages SOA applications running on Oracle SOA Suite.



Effective Management of SOA Applications with Semantic Modeling  
November 2008

Oracle Corporation  
World Headquarters  
500 Oracle Parkway  
Redwood Shores, CA 94065  
U.S.A.

Worldwide Inquiries:  
Phone: +1.650.506.7000  
Fax: +1.650.506.7200  
[oracle.com](http://oracle.com)

Copyright © 2008, Oracle and/or its affiliates. All rights reserved.  
This document is provided for information purposes only and the contents hereof are subject to change without notice.  
This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners. 1108