

Auf Schatzsuche in Oracle BPM und SOA Suite 11g

Dr. Jens Hündling, ORACLE Deutschland B.V. & Co. KG, und Danilo Schmiedel, OPITZ CONSULTING GmbH

Die aktuelle SOA/BPM Suite bietet eine Reihe nützlicher Features, die selten genutzt und wenig beworben werden. Die Autoren wissen aus verschiedenen Kundenprojekten, dass sich viele Anforderungen, die zunächst mit einem erhöhten Implementierungs- oder Anpassungsaufwand verbunden zu sein scheinen, anschließend doch mit Bordmitteln realisieren lassen. In diesem Artikel geben wir uns auf die Suche nach solchen „versteckten Schätzen“.

Diese Schätze zu nutzen, bedeutet im Projektalltag einen deutlichen Mehrwert. Neben leicht zu findenden Juwelen wie den Metadata Services (MDS) oder dem Fault Management Framework widmen wir uns auch „tiefer vergrabenen“ Features wie den sogenannten „Component Properties“, den Möglichkeiten zur dynamischen Adressierung von Endpunkten und den Steuerungsoptionen zur Speicherung der Prozessinstanzen in der Datenbank.

Mit der BPM Suite stellt Oracle eine Infrastruktur-Lösung bereit, um die wesentlichen Anforderungen der technischen Implementierung einer SOA auf der Grundlage von Webservice-Technologien zu unterstützen. Somit lassen sich Services erstellen, verwalten und zu modularen Anwendungen sowie technischen Geschäftsprozessen zusammenstellen. Mit sogenannten „Hot-Plug-Komponenten“ können Unternehmen ihre Architekturen problemlos ausbauen und weiterentwickeln, statt vorhandene Investitionsgüter zwingend gegen neue austauschen zu müssen.

Wiederverwendung von Daten und Services mittels MDS

Ein wichtiger Aspekt in einer serviceorientierten Architektur ist die Veröffentlichung der Metadaten von potenziell wiederverwendbaren SOA-Artefakten. Mit den Metadata Services (MDS) bietet die Oracle SOA/BPM Suite eine solche Komponente an, um die zentrale Verwaltung und Bereitstellung von Schnittstellen (WSDL), Daten (XSD), Geschäftsregeln (Rules), FaultPolicies (XML) und Geschäftsereignissen (EDL) zu ermöglichen (siehe Abbildung 1). Somit können Daten und Dokumente zur Design- und

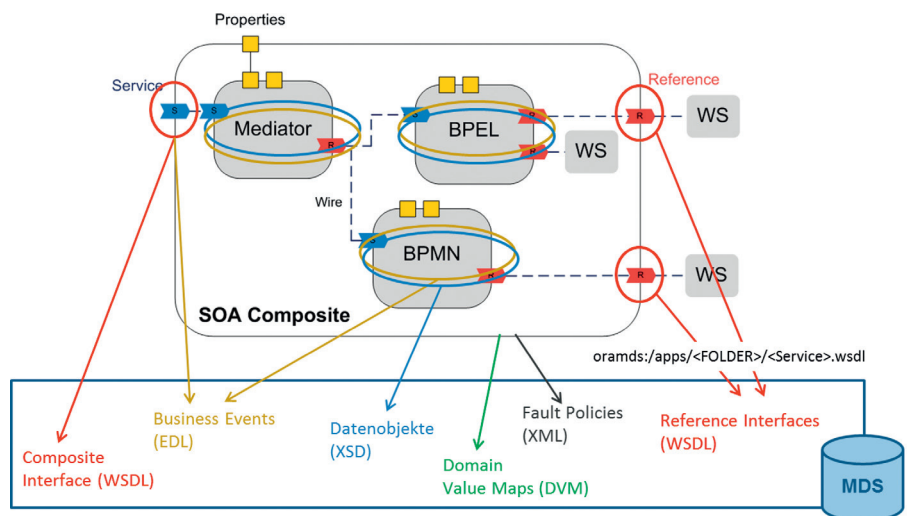


Abbildung 1: Effektive Software-Entwicklung mit MDS

Laufzeit applikationsübergreifend angeboten werden und die Duplizierung von Code wird reduziert.

Ohne eine solche Komponente müssten derartige Artefakte entweder für verschiedene Projekte dupliziert abgelegt oder direkt auf den Applikationsserver referenziert werden. Diese beiden Ansätze bringen jedoch auch Nachteile mit sich. Während die lokale Mehrfachspeicherung von Schnittstellen (WSDL) und Datenbeschreibungen (XSD) in unterschiedlichen Projekten zu Inkonsistenzen führen kann, ist die direkte Referenzierung auf einen Applikationsserver an dessen Erreichbarkeit gebunden. Ist dieser etwa aus Wartungsgründen vorübergehend nicht verfügbar, so würde die Entwicklung in dem Moment ebenfalls gestört werden.

Die Berücksichtigung des MDS führte in verschiedenen SOA-Projekten zu folgenden Erkenntnissen und Best Practices:

1. Die Versionierung von Services und Daten ist für den Betrieb einer serviceorientierten Architektur unumgänglich und sollte im Rahmen des Service-Designs berücksichtigt werden (Service Lifecycle Management). Dies gilt insbesondere für Schnittstellen und Daten-Objekte, die in verschiedenen Projekten wiederverwendet werden. Andernfalls bedeuten Änderungen an diesen Komponenten einen erhöhten Entwicklungsaufwand und wenig Potenzial für die Wiederverwendung.
2. MDS unterstützt die Software-Entwicklung nach dem „Contract-first-Prinzip“. Nach dem Design der Schnittstelle durch den Service-Architekten wird diese in MDS geladen und steht dem Service-Entwickler zur Implementierung bereit.
3. Als Best Practice hat sich zur Entwicklungszeit die Verwendung eines lokalen

MDS-Repository bewährt. Dieses wird bereits mit der Standard-Installation des Oracle JDevelopers bereitgestellt. Auf performante Art und Weise kann somit die zukünftige Speicherung von verteilten Artefakten in einem zentralen Repository von Beginn an berücksichtigt werden. Besonders vorteilhaft ist dabei auch, dass jeder Entwickler eine eigene Lokation für seine Metadaten besitzt und nicht jeder Entwicklungsschritt sofort „deployed“ werden muss.

4. Zur Laufzeit stellt die Oracle SOA Suite ein zentrales MDS-Repository auf dem Server bereit, in dem die verteilten Schnittstellen, Daten, Event-Definitionen und Fault-Policies im Rahmen des Applikations-Deployments bereitgestellt werden.
5. Unabhängig davon, ob der lokale oder der Server-MDS verwendet werden soll, erfolgt die Referenzierung der Inhalte aus einem SCA Composite über eine einheitliche URL: `oramds:/apps/<ORDNER>/<Service>.wsdl`. Somit ist beim Wechsel zwischen lokalem und zentralem MDS keine gesonderte Code-Anpassung erforderlich.
6. Die Bereitstellung zentraler Schnittstellen bringt den Vorteil mit sich, dass beispielsweise beim Neustart des Applikationsservers alle SOA-Applikationen unabhängig voneinander gestartet werden können. Die Applikationen sind also nicht an eine spezielle Reihenfolge gebunden, da alle Schnittstellen-Beschreibungen von Beginn an verfügbar sind.
7. Die Administration des MDS Repository zum Export, Import oder Löschen der Metadaten erfolgt über den Enterprise Manager, ANT oder das WebLogic Scripting Tool (WLST).

Behandlung von Fehlern mit dem Fault Management Framework

Die BPM und SOA Suite 11g bietet einen einheitlichen Ansatz für die Behandlung von Fehlern auch über die Grenzen der einzelnen Komponenten eines SCA-Composite hinweg. Wichtig ist hierbei, dass unterschieden wird zwischen fachlichen Fehlern, die Bestandteil der Businesslogik sind, und technischen Fehlern (etwa aufgrund von fehlerhaften Datentransformationen oder nicht verfügbaren Service-Endpunkten).

```
<component name="TestProcess" version="1.1">
  <implementation.bpel src="BPEL_PROZESS_NAME.bpel"/>
  <property name="bpel.preference.testPreference">TestValue</property>
</component>
```

Listing 1

Neben der Möglichkeit, das Fehlerhandling direkt in einem „Business Process Execution Language (BPEL)“- oder „Business Process Model and Notation (BPMN)“-Prozess zu erledigen, stellt die Oracle-Plattform auch das sogenannte „Fault-Management-Framework“ bereit. Tritt etwa ein Fehler beim Aufruf einer Prozess-Aktivität auf, fängt das Framework diesen ab und führt eine konfigurierte Aktion durch, die zuvor in einem assoziierten Fault-Policy-File definiert wurde. Diese Definition kann zentral abgelegt werden, womit eine Basis für die Wiederverwendung von Logik für ein effizientes Fehler-Alerting, -Logging und -Monitoring gegeben ist. Besonders hervorzuheben ist in diesem Rahmen auch die Möglichkeit, während der Laufzeit in Prozess-Instanzen einzugreifen.

Auf diese Weise können fehlerhafte Aufrufe über die Administrationsoberfläche erneut angestoßen werden, ohne einen übergeordneten Gesamtprozess neu starten zu müssen. Indem das Fault-Management-Framework aus den konfigurierten Aktionen beim Eintreffen eines Fehlers einheitliche Benachrichtigungs- und Auswertungsmechanismen anstößt, unterstützt es die zügige Identifikation von verschiedenen Fehlerklassen. Insofern bildet es einen wichtigen Baustein zur Verbesserung der Software-Qualität.

Component Properties: zur Laufzeit veränderbare Konfigurationsparameter

In jedem Software-Projekt entsteht über kurz oder lang der Bedarf, das Laufzeitverhalten einer Applikation mithilfe von Properties zu steuern. Ein Beispiel dafür stellt die Ausführungsart dar, die besagt, ob eine Anwendung im Test-, Simulations- oder Produktions-Betrieb gestartet werden soll. Auch die Festlegung eines Dateipfades, von dem etwa ein FTP-Adapter seine Files einlesen soll, ist ein Anwendungsbeispiel für die Notwendigkeit von Properties, da derartige Informationen idealerweise nicht in einer

fachlich gestalteten Schnittstelle verortet sind.

In der BPM und SOA Suite bietet sich hierfür die Verwendung sogenannter „Component Properties“ an. Die Definition der Properties erfolgt in der Hauptdatei einer Composite-Applikation – der „composite.xml“ (siehe Listing 1).

Mit der Funktion „ora:getPreference („testPreference“)“ können derartige Properties nun beispielsweise innerhalb eines BPEL-Prozesses ausgelesen und verwendet werden. Zusätzlich zur umgebungsspezifischen Anpassung der Werte für unterschiedliche Deployments (mittels Config-Plan) können diese auch zur Laufzeit verändert werden. Dies geschieht im sogenannten „MBean-Browser“ des Enterprise Managers, der unter den folgenden Menü-Optionen erreicht werden kann: „Menü Administration -> System-MBean Browser -> Anwendungsdefinierte MBeans -> oracle.soa.config -> COMPOSITE_NAME -> SCAComposite.SCAComponent -> PROCESS_NAME“. Der Vorteil dieser Properties liegt insbesondere darin, dass diese Funktionalität als integrierter Produktbestandteil vorliegt und somit keine separaten Konfigurationsdateien mit gesonderter Ablage, speziellen Berechtigungen und individuellen Einlese-Routinen benötigt werden.

BPMN, BPEL – und Java?

Im Kontext der Service Component Architecture (SCA) besitzen BPMN und BPEL auch Berührungspunkte mit der weit verbreiteten Programmiersprache Java. So kann vorhandener Java-Code beispielsweise über die Spring-Context-Komponente in eine Composite-Applikation eingebunden werden. Ferner besteht mittels sogenannter „Embedded-Java-Aktivitäten“ beziehungsweise via Mediator die Möglichkeit, den Titel einer Prozess-Instanz in der Instanzen-Ansicht des Enterprise Managers zu modifizieren – eine für die Fehleranalyse und das Monitoring enorm hilfreiche Funktionalität, da die Auflistung mit fachlichen

Informationen angereichert werden kann. In der Praxis sorgt dies für eine effizientere Auswahl relevanter Durchläufe.

Zur Integration eines LDAP-Verzeichnisses bieten BPM Suite und SOA Suite eine Reihe sogenannter „Identity Provider“ an. Für seltene und nicht standardisierte Lösungen ist es auch möglich, einen eigenen Java-basierten Identity Provider zu integrieren und somit die Plattform um wichtige Security-Aspekte zu erweitern. Schließlich können aus dem oben erläuterten Fault-Management-Framework ebenfalls Aufrufe zu Java-Klassen initiiert werden.

Steuerung der Dehydration

Im Lebenszyklus eines BPEL-Prozesses werden dessen Prozess-Instanzen mit Status-Informationen im Repository der Oracle SOA Suite gespeichert. Diesen Speichervorgang nennt man im BPEL-Umfeld auch „Dehydration“ und die dazugehörige Datenbank ist der „Dehydration Store“. Sobald eine laufende Instanz (an bestimmten Aktivitäten im Prozess) dehydriert wurde, entfernt der BPEL-Server diese aus dem Memory. Wenn daraufhin etwa ein Callback eines asynchronen Service-Aufrufs eintrifft oder ein abgelaufener Timer „feuert“, holt

sich der BPEL-Server die passende Instanz aus dem Dehydration Store zurück in den Memory und fährt mit der Ausführung fort. Einerseits gewährleistet dieser Vorgang die Ausfallsicherheit, da laufende Instanzen nach einem Neustart des BPEL-Servers oder nach einem fehlerhaften Aufruf automatisch am letzten Dehydrierungspunkt fortgesetzt werden können. Andererseits kann die Dehydrierung in einem Umfeld mit High-Performance-Anforderungen und einer enormen Anzahl an Prozess-Instanzen kritisch sein.

Wann und wie die Dehydration ausgeführt wird, hängt vom gewählten Prozess-Typ ab. Die Oracle BPEL Engine unterscheidet hierbei zwischen „transient“ und „durable“. Transiente Prozesse verursachen keine Dehydration während der Ausführung der Instanz. In Summe werden transiente Prozesse nur einmal am Ende des Prozesses dehydriert. Beispiele hierfür sind kurzlaufende, synchrone Request-Response-Szenarios. Prozesse vom Typ „durable“ werden mehr als einmal – jeweils bei bestimmten Aktivitäten (wie Receive, OnMessage oder Wait) – dehydriert.

Prozesse, für die im Betrieb sehr viele Instanzen zu erwarten sind, sollten daher als transiente Prozesstypen umgesetzt werden. Im Allgemeinen sind hierfür die folgenden Anforderungen zu erfüllen:

- Keine Receive-Aktivitäten innerhalb des Prozessflusses
- Keine Wait-Aktivitäten
- Keine OnMessage-Aktivitäten
- Eine synchrone Schnittstellenbeschreibung

Sind diese Voraussetzungen erfüllt, müssen auf Composite-Ebene weitere Attribute gesetzt sein, um eine vollständige Ausführung des Prozesses im Memory zu gewährleisten (siehe Tabelle 1).

Geschwindigkeitsvorteil

Das Setzen der Werte führt bei synchronen Prozessen zur Steigerung der Performance und zur Reduktion der Laufzeitdaten, die in das Repository geschrieben werden. Gemäß den Erfahrungen aus unterschiedlichen Projekten konnte damit ein messbarer Geschwindigkeitsvorteil realisiert werden.

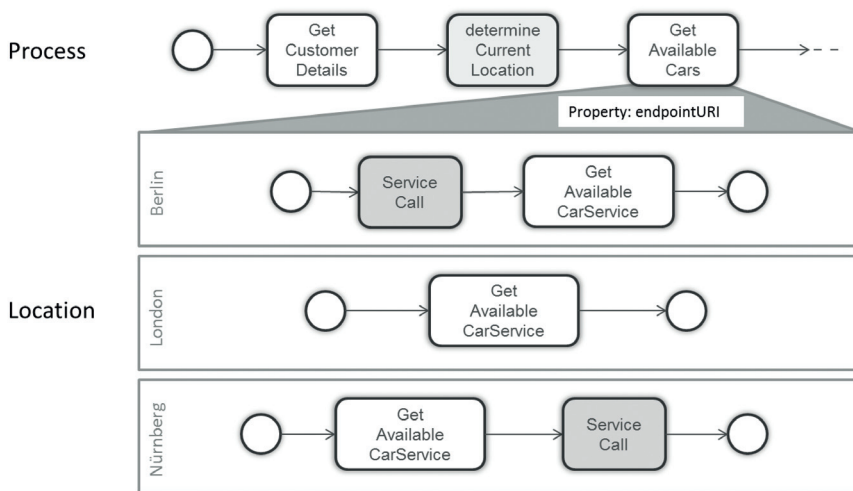


Abbildung 2: Lose Kopplung mit dynamischem Endpunkt-Binding

Konfigurationsparameter	Beschreibung
inMemoryOptimization=true	Der Standardwert dieses Parameters ist „false“. Die Eigenschaft kann nur gesetzt werden, wenn keine der oben genannten Aktivitäten im Prozess enthalten ist. Steht der Wert auf „true“, wird für Instanzen dieses Prozesses eine Speicheroptimierung bei Wertzuweisungen vorgenommen.
completionPersistPolicy=faulted	Diese Property definiert, welche Prozess-Instanzen gespeichert sind. Mit dem Setzen des Wertes auf „faulted“ werden nur fehlgeschlagene Instanzen gespeichert.
oneWayDeliveryPolicy=sync	Der Parameter setzt die Persistenz-Eigenschaften des Prozesses im Delivery Layer. Mit der Zuweisung des Werts „sync“ wird erreicht, dass die Instanz-erzeugende Nachricht nicht temporär im Delivery Layer gespeichert wird. Die Engine benutzt den Speicher für das Erzeugen der Nachricht.

Tabelle 1: Weitere Attribute auf Composite-Ebene

In einer SOA findet sich in Praxis-Projekten häufig ein spezielles Problem: die dynamische Adressierung von Services. Es betrifft typischerweise Services, die zwar die gleiche Schnittstelle besitzen, aber auf unterschiedlichen Maschinen gehostet sind. Zur Laufzeit muss nun ermittelt werden, welcher konkrete Endpunkt auf welchem Server respektive in welcher Lokation angesprochen werden kann (siehe Abbildung 2). Durch die Verbindung mit Oracle Business Rules, Domain Value Maps (DVM) oder Component Properties ist es möglich, die Adresse von Service-Endpunkten zur Laufzeit anzupassen. Das dynamische Routing erfolgt mithilfe der Property „endpointURI“.

Konzeptionell lässt sich Ähnliches natürlich auch mit einem Service Bus oder der Mediator-Komponente realisieren. Die hier vorgestellte Methode eignet sich allerdings besser in speziellen Szenarien beziehungsweise, wenn kein Service Bus in der vorhandenen Architektur existiert. Nur so kann zur Laufzeit der tatsächliche physikalische Endpunkt eines Service beeinflusst werden. Der Vorteil gegenüber einem Mediator oder Service Bus ist hierbei, dass kein Routing und keine Transformation für die Integration eines zusätzlichen Endpunktes konfiguriert beziehungsweise geändert werden muss und somit eine dynamische Skalierung möglich ist.

Fazit

Zu Beginn dieses Artikels bezeichneten wir die beschriebenen Features als „Schätze“. Dies liegt vor allem darin begründet, dass deren Existenz nicht sofort ersichtlich ist. Hat man sie jedoch einmal gefunden, so zeigt die Erfahrung aus vielen Projekten, dass ihre Verwendung einen erheblichen Mehrwert im Bereich der Software-Qualität und des Lifecycle-Managements liefert.

Dr. Jens Hündling

jens.huendling@oracle.com

Danilo Schmiedel

danilo.schmiedel@opitz-consulting.com



Dr. Frank Schönthaler, Leiter DOAG Business Solutions Community und der E-Business Suite Community

E-Business Suite Community Day und Workshop der DOAG BSC

Die Business Solutions Community veranstaltete vom 17. bis 18. Oktober 2011 in München den E-Business Suite (EBS) Community Day und Workshop. Der Community Day war durch abwechslungsreiche und hoch interessante Fachbeiträge gekennzeichnet. Aber auch das Networking mit Anwendern und Experten stand im Vordergrund. Zudem bot der Community Workshop am 18. Oktober Best-Practice-Lösungen rund um das Thema „Costing“.

Trotz positiver Entwicklungen der globalen Märkte nimmt das Risiko-Potenzial weiter zu – unsichere Währungsentwicklungen,

gen, technologische Fragen und politische Wirren rufen nach Veränderungsfähigkeit von Unternehmen. Die Unternehmensleitung reagiert auf diese Herausforderungen mit neuen Geschäftsmodellen und Strategien, die eine erhebliche Elastizität und Skalierbarkeit der Geschäftsprozesse notwendig machen. Am ersten Tag des Community-Treffens wurden Fragen zur Umsetzung, Organisation und konzernweiten Vereinheitlichung von Geschäftsprozessen behandelt. Daneben wurde aufgezeigt, welchen Beitrag die hierbei eingesetzten Oracle-Produkte leisten können. Angesprochene Themen waren die Realisierung eines Konzernwechsels mit der E-Business Suite und das Ertrags- und Kosten-Management in komplexen und unsicheren Geschäftssituationen.

Darauf aufbauend sprachen die Mediatorin Margit Müller und Dr. Frank Schönthaler von PROMATIS am Beispiel der E-Business Suite über den Zusammenhang von Governance und der Nutzung einer Unternehmenssoftware. Die zugrunde liegende Thematik befasst sich über die Implementierung von neuen Softwareprodukten hinaus mit der Frage, wie die Mitarbeiter mit der neuen Situation umgehen und wie man ihnen die neue Software näher bringt. Dabei wurden Verfahren vorgeschlagen, die eine nachhaltige Umsetzung der Governance-Strategie und damit auch den Nutzen der Unternehmenssoftware-Investition sicherstellen. Abgerundet wurde die Veranstaltung durch praxisnahe Erfahrungs- und Anwenderberichte, die diese interessant und abwechslungsreich gestalteten.

Am zweiten Tag des Treffens wurde den Community-Mitgliedern und EBS-Interessierten ein Praxis-Workshop angeboten. Auf der Grundlage ihrer reichhaltigen Stammdaten ermöglicht die E-Business Suite kostenmäßige Betrachtungen über die gesamte Wertschöpfungskette hinweg. Geschäftsprozesse und Setup-Parameter müssen optimal miteinander abgestimmt werden. In diesem anwenderorientierten Workshop wurde diese meist unterschätzte Komplexität aufgezeigt und Lösungsansätze dargeboten. Dabei zeigten Experten der E-Business Suite ihre Best-Practice-Lösungen live am System – Erfahrungswissen aus erster Hand.

Das angenehme Ambiente des Veranstaltungsorts und die produktive Stimmung luden die Teilnehmer zum Networking mit intensivem Informations- und Erfahrungsaustausch ein. Insgesamt waren alle Teilnehmer sehr zufrieden mit der Veranstaltung, die eine durchaus größere Beteiligung aus der Community verdient gehabt hätte. Wenn Sie sich näher für die E-Business Suite Community der DOAG interessieren, finden Sie hier weitere Informationen: <http://bs.doag.org/de/bs/communities/e-business-suite.html>

Dr. Frank Schönthaler

frank.schoenthaler@doag.org