

# Entwickeln mit Oracle BPEL

**Uwe Ritter**  
**People at Work Systems AG**  
**München**

**Schlüsselworte: Business Process Management, Geschäftsprozesse, BPEL, BPM, EAI, SOA**

## **Einleitung**

Die computergestützte Verarbeitung von Geschäftsprozessen ist nicht erst seit gestern ein Thema. Allerdings wurden Geschäftsprozesse in der Vergangenheit zumeist innerhalb monolithischer Anwendungssysteme realisiert. Herausforderungen und Probleme traten immer dann auf, wenn die Funktionalitäten, die zur korrekten und vollständigen Bearbeitung eines Geschäftsprozesses über mehrere Anwendungssysteme verteilt waren. Dies bedingte, dass die dazu notwendigen Anwendungssysteme entlang der einzelnen Prozessschritte miteinander kommunizieren können.

Meist wurden dazu die beteiligten Anwendungssysteme auf einer 1:1 Basis miteinander vernetzt und die entsprechende Verbindungslogik bilateral zwischen den Kommunikationsknoten erstellt. Während dies bei wenigen Systemknoten noch relativ einfach zu bewältigen ist, so steigt die Anzahl der notwendigen Verbindungen quadratisch zur Anzahl der Basissysteme. Viele Systemlandschaften haben sich über die Zeit so entwickelt. Ein so entstandener „Spaghetti“ ist nicht nur teuer im Betrieb, sondern auch schwer wartbar, da ein einzelner Releasewechsel bei einem der angeschlossenen Anwendungssysteme ohne Weiteres zu Nacharbeiten im gesamten Konstrukt führen kann.

Also scheint es eine folgerichtige Strategie zu sein, die Teile der Geschäftsprozesse, die sich um die Orchestrierung der Zusammenarbeit der Anwendungssysteme kümmern aus diesen Systemen herauszunehmen und einer geeigneten Plattform zu übertragen. Diese Überlegung war dann (gemeinsam mit der Notwendigkeit, die Anzahl der physikalischen Schnittstellen zwischen den Systemen zu reduzieren) die gedankliche Basis für heutige EAI (Enterprise Application Integration) und BPM (Business Process Management) Systeme.

Die zunehmende Vernetzung der Menschen durch Internet und Intranet hoben dann die Überlegung auf die nächste Stufe: wie wäre es, wenn der eigentliche Geschäftsprozess komplett auf einer eigenen Plattform abliefe und die bestehenden (oder neue) Anwendungssysteme dem Geschäftsprozess oder Workflow einfach nur mehr oder weniger atomare aber klar umrissene „Dienste“ anböten? Das Ergebnis kennen wir, namentlich Serviceorientierte Architekturen (SOA).

Um sinnvoll und wirtschaftlich eingesetzt werden zu können sollte die Implementierung einer solchen Umgebung idealerweise folgenden Ansprüchen genügen:

- Lesbarkeit der Prozesse für Fachabteilungen nach deren Implementierung
- Agilität der Gesamtumgebung
- Flexibilität und Erweiterbarkeit
- Wartbarkeit der Umgebung
- Portabilität der Prozesse
- Reduktion der Notwendigkeit für manuelle Ausnahmebehandlung
- Releasefähigkeit (von Plattform und Anwendungssystemen)
- Investitionsschutz

Dies bedingt eine Umgebung, mit der geschäftliche Abläufe, die zuvor auf Papier, mit grafischen Werkzeugen oder mit Systemen wie Aris® entworfen wurden, dargestellt, programmiert und ausgeführt werden können. Während zu anfangs große Hersteller jeweils eigene Umgebungen entwickelten, zeichnete sich 2002 eine Standardisierung ab, als Microsoft (XLANG) und IBM WSFL – Web Services Flow Language) ihre Entwicklungen zusammenlegten und die Business Process Execution Language for Web Services (BPEL4WS) vorgestellt wurde und deren Standardisierung vom OASIS Konsortium vorangetrieben wird.

BPEL ist eine XML-basierte Sprache zur Modellierung von Workflows und Geschäftsprozessen auf der Basis von Web Services. Die aktuelle Version 2.0 heißt WS-BPEL. WS steht hier für Web Services.

Als Vorteile von BPEL und die damit erstellten Abläufen werden oft folgende Punkte genannt:

- Modular
- Flexibel
- Wiederverwendbar
- Einfach
- Lesbar
- Einfach zu testen
- Unabhängig von Systemumgebung
- Deklarative Modellierung
- etc.

Während dies in der Theorie und für „Hello World“ Anwendungen zutreffen mag, sollte man BPEL in der Praxis bei größeren Projektvorhaben doch um einiges differenzierter betrachten. Details siehe unten.

## Oracle BPEL in der Praxis

Die BPEL Umgebung von Oracle firmiert unter dem Namen *Oracle BPEL Process Manager*. Die Entwicklungsumgebung des Oracle BPEL Process Manager ist in die Oracle JDeveloper Umgebung eingebettet. Die Hauptkomponenten sind (in 11g):

- Oracle BPEL Process Designer
- Oracle BPEL Process Manager
- Human Workflow
- SOA Composite Editor
- Business Activity Monitoring
- Integration der Oracle Business Rule Engine
- Oracle Adapter Framework

Damit steht eine relativ mächtige und leistungsfähige Entwicklungsumgebung zur Verfügung, die einen grafischen Entwurf der Workflows/Prozesse ermöglicht. Das ist zunächst einmal sehr komfortabel, wer allerdings gewohnt ist, in Projekten mit Java in Umgebungen wie Oracle JDeveloper oder Eclipse zu arbeiten, wird bei näherem Hinsehen relativ schnell auf einige Herausforderungen stoßen. Das liegt weniger in der Implementierung der Oracle BPEL Umgebung, die als eine der derzeit besten BPEL-Umgebungen angesehen werden kann, begründet als vielmehr in der Struktur von BPEL selbst.

BPEL beschreibt Geschäftsprozesse in XML. Dies ist sicherlich ein weiterer der Abstraktion bei der Definition von Programmabläufen. Allerdings ist BPEL im Vergleich zu modernen Programmiersprachen immer noch ein relativ neuer Standard. Dies bedeutet, dass in der Praxis die derzeit vorhandenen Entwicklungswerkzeuge sind trotz der Mächtigkeit von BPEL noch ein Stück vom Funktionsumfang einer modernen Java Entwicklungsumgebung entfernt sind. Die Entwicklung großer Prozesse ist noch etwas schwerfällig und die Fehleranalyse und das systematische Testen von Prozessen gestalten sich eventuell zeitintensiv.

Dies liegt in der Hauptsache an folgenden wesentlichen Einflussfaktoren:

*Strukturelle BPEL Herausforderungen:* BPEL kennt keine Unterprogramme. Damit entfällt zunächst einmal die Möglichkeit, sich Unterprogramm-Bibliotheken für oft genutzte Workflow-Elemente anzulegen. Dies erhöht nicht nur die Entwicklungszeit, sondern senkt auch die theoretisch vorhandene Wiederverwendbarkeit von Elementen. Weiterhin erhöht dies den Aufwand für die Erstellung neuer Prozesse. Man kann sich hier zwar behelfen, indem man die XML-Orientierung von BPEL nutzt und oft genutzte Programmteile vor dem Compilieren über XSL Transformationen in den Code hinein generiert, doch wäre eine generischer Ansatz für künftige Erweiterungen des Standards sicherlich wünschenswert.

*XML Herausforderungen:* BPEL basiert auf XML. Das ist gut, denn XML ist ein verbreiteter und anerkannter Standard. Allerdings neigen die XML Konstrukte innerhalb von BPEL Implementierungen dazu, sehr groß zu werden, was schwierig zu handhaben ist. Weiterhin impliziert BPEL viele formale Anforderungen an das Datenschema. Dies führt typischerweise

zu vielen Mappings. Situationsbezogene Einzeldaten gehen dabei leicht unter, so dass sich die Fehlersuche durchaus als zeitraubend herausstellen kann. Wenn sich beispielsweise in einer XML Variablen mit 400 Zeilen nur ein Element geändert hat, ist das nicht ohne weiteres sofort ersichtlich. Die Oracle BPEL Umgebung gibt hier als Debugginghilfe z.B. dann leider auch nur die Variable an und nicht das exakte Element innerhalb dieser Variablen. Weiterhin sind komplexe Strukturen automatisch dokumentationsintensiv.

*Entwicklungsablauf:* Salopp gesprochen erinnert der Entwicklungsprozess mit BPEL an die „guten alten Zeiten“ z.B. vor TurboPascal. Der Entwicklungsablauf ist streng iterativ:

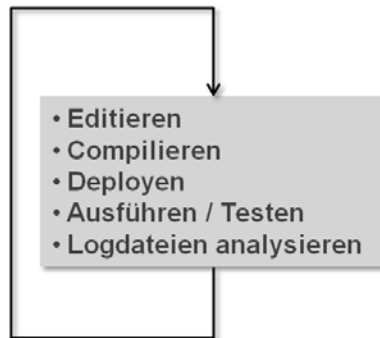


Abb. 1: Entwicklungsablauf

In einem komplexen Projekt (siehe Beispiel unten) kann jede Iteration (ohne Fehlersuche) durchaus 3 bis 10 Minuten dauern, was in Projekten ohne die Unterstützung komplementärer Werkzeuge möglicherweise zu einem erhöhten Ressourcenbedarf hinsichtlich der eingesetzten Entwicklungsmannschaft führen kann.

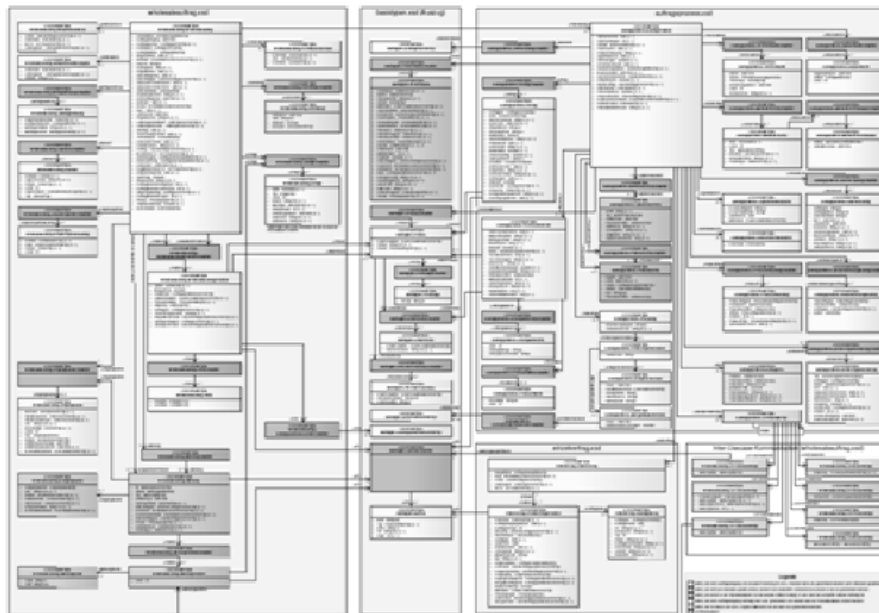


Abb. 2: Beispiel einer Orchestrierungsschicht eines komplexen Projektes

Diese Erfahrungen wurden im Rahmen eines BPEL Projektes gemacht, in dem 40 bis 70 Entwickler an ca. 300 BPEL Prozessen (~150 fachliche Prozesse, ~70 Prozesse zur Datenbankabstraktion, ~30 Schnittstellenprozesse, ~50 sonstige technische Prozesse) mit jeweils 50 bis 150 Prozessschritten, 20 bis 1000 Elementen pro XML Datei und 5 bis 9 aktiven Entwicklungszweigen gearbeitet haben. Die Produktionsdatenbank hat eine Größe von ca. 3 TB und es werden von dem System derzeit mehr als 15.000 Aufträge pro Tag abgewickelt (50.000 sind geplant).

Durch den Einsatz eines selbst entwickelten Add-Ons für den Oracle BPEL Process Manager konnten beispielsweise die Ausführungszeiten für einen Testfall von durchschnittlich 5 Minuten auf unter eine Sekunde gesenkt werden, die Fehlersuche durch Drill-Down-Möglichkeiten in den Ergebnisvariablen beschleunigt und Testfälle erheblich schneller generiert werden.

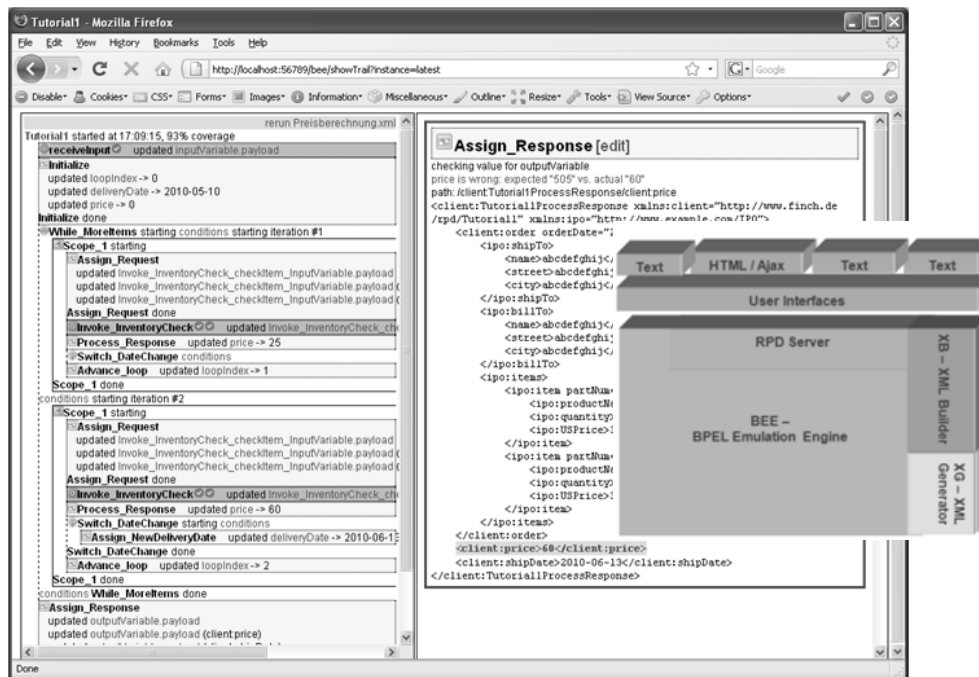


Abb. 3: Add-On für den Oracle BPEL Process Manager

## Fazit

BPEL ist eine mächtige Umgebung und sehr wohl für die Implementierung von Geschäftsprozessen geeignet. Gleichzeitig ist BPEL ein vergleichsweise junger Standard. Wie oben beschrieben, kann BPEL nicht die Antwort auf alle Problemstellungen hinsichtlich der Automatisierung und Orchestrierung von Geschäftsprozessen sein, sondern sollte gezielt und überlegt eingesetzt werden. BPEL eignet sich durch fehlende Komponenten für die Gestaltung von Benutzeroberflächen am besten für die Implementierung von Back-End Prozessen. Angesichts der beschriebenen strukturellen Gegebenheiten von BPEL sollte man

in der Entwurfsphase eines BPEL Projektes sehr genau planen, welche Teile des Geschäftsprozesses/Workflows sinnvoll mit BPEL umgesetzt werden können und welche Teile am besten mit komplementären Techniken (z.B. Java oder BPMN) realisiert werden. Der Einsatz von zum Oracle BPEL Process Manager komplementären Add-Ons macht (soweit vorhanden) ebenfalls Sinn.

**Kontaktadresse:**

**Name**

Uwe Ritter

People at Work Systems AG

D-80939 München

Telefon: +49(0)89-540 42 45 - 0

Fax: +49(0) 89-540 42 45 - 11

E-Mail [uwe.ritter@paw-systems.com](mailto:uwe.ritter@paw-systems.com)

Internet: [www.paw-systems.com](http://www.paw-systems.com)