

Oracle BAM – Die unentdeckten Möglichkeiten

Dr. Markus Vincon
Opitz Consulting Essen GmbH
Essen

Schlüsselworte:

Oracle SOA Suite 11g, Oracle Business Activity Monitoring (BAM), BPEL, Performance Monitoring

Einleitung

Wir erleben es immer noch sehr häufig, dass den Kunden und Nutzern der Oracle SOA Suite nicht bewusst ist, welche Möglichkeiten ihnen mit Oracle Business Activity Monitoring (BAM) schon zur Verfügung stehen. Während wir bei einigen Kunden Oracle BAM bisher nur als zentrale Sammel- und Verteilstelle für Fehler und sonstige Meldungen auf eher technischer Ebene einsetzen, haben wir immerhin auch schon einige Kunden, die sich anhand von Dashboards Übersicht über ihre Prozesse und die Daten in ihren Prozessen verschaffen. Dies kann sogar so weit gehen, dass Fachbereiche bei übergreifenden Prozesse jederzeit prüfen können, ob erwartete Ereignisse oder Aktionen eingetreten sind. Insgesamt ist die Verbreitung von Oracle BAM angesichts des Nutzen für das Prozess Controllings aber noch zu gering und daher möchte ich diesen Vortrag nutzen, um in Anlehnung an meinen Vortrag des letzten Jahres aber auch darüber hinausgehend den Nutzen und den Einsatz von Oracle BAM zu motivieren. Die Empfehlung ist aber trotzdem, sich bei Interesse an den Grundlagen und den grundlegenden technischen Möglichkeiten, sich den Vortrag von der DOAG-Konferenz 2010 „Volle Übersicht über Metadaten und Prozessdaten mit Oracle BAM“ durchzulesen.

Einsatzmöglichkeiten von Oracle BAM

Zu den typischen Einsatzszenarien, die für Kunden umgesetzt werden, gehören eine zentrale Fehlerüberwachung und Fehlerverteilung aus den SOA-Prozessen sowie eine Übersicht über die Datenflüsse allgemein. Am Beispiel einer möglichen systemweiten (bezogen auf SOA) Fehlerüberwachung soll etwas weiter hinten noch einmal die grundlegenden Strukturen von Oracle BAM aufgezeigt werden. Datenflusskontrollen selbst sind wieder sehr spezifisch bezogen auf die übermittelten Daten, aber mit Hilfe entsprechender Berichte erlauben sie auch dem Fachbereich einen schnellen Überblick über alles, was im System passiert oder vielleicht auch gerade nicht passiert.

Tatsächlich ist es auch schon vorgekommen, dass bei gekoppelten Kommunikationen (d.h. Informationen gehen an Tag X von System A zu System B und werden dann an einem der nächsten Tage und nach Verarbeitung in System B wieder von System B zu System A bestätigt), dass der Fachbereich mit einem Bericht über den Zusammenhang zwischen erster Übermittlung und Bestätigung in Verbindung mit den Zeitstempeln der Übermittlungen, gezielt die richtigen Fragen stellen konnte und sich herausstellte, dass die Übermittlung über das Netzwerk im technischen Sinne für die SOA-Suite als erfolgreich markiert war, aber die Daten in Wirklichkeit nicht in System B angekommen sind (und daher auch nicht bestätigt werden konnten).

Ein weiteres und unserer Meinung nach sehr interessantes Einsatzgebiet von Oracle BAM ist die sogenannte Performanceüberwachung von BPEL-Prozessen (oft auch als Monitor Express

bezeichnet). Sie erlaubt es mit wenigen Einstellungen die Laufzeiten, Anzahl und Kennzahlen von BPEL-Prozessen bis herunter auf die Ebene der einzelnen Aktivitäten zu messen. Neben den wenigen Einstellungen im Prozess müssen natürlich noch Datenobjekte und Berichte in Oracle BAM angelegt werden. Diese gibt es aber schon in einer sehr brauchbaren Version als Beispiel von Oracle selbst (kann problemlos per icommand in Oracle BAM eingespielt werden). Dieses Monitor Express wird später noch ausführlicher erläutert.

Vorteile von Oracle BAM

Neben dem Nachteil, dass Oracle BAM als Überwachungswerkzeug einigermaßen fest an die Oracle SOA Suite gebunden ist, sehe ich genau das aber auch gleich wieder als großen Vorteil. Die Integration von Oracle BAM mit der SOA-Suite und insbesondere den BPEL-Prozessen funktioniert gut, nahtlos und schnell. Neben der Möglichkeit, über Aktivitätssensoren und Sensoraktionen auf recht einfache Art und Weise (ich empfehle zu dem Punkt den schon in der Einführung erwähnten Vortrag aus 2010) Daten nach Oracle BAM zu transportieren, werden wir später noch sehen, wie schnell man erste Ergebnisse mit der Performanceüberwachung bzw. Monitor Express erzielen kann.

Zusätzlich gibt es noch viele andere Möglichkeiten, Daten nach Oracle BAM zu transportieren (siehe folgende Abbildung).

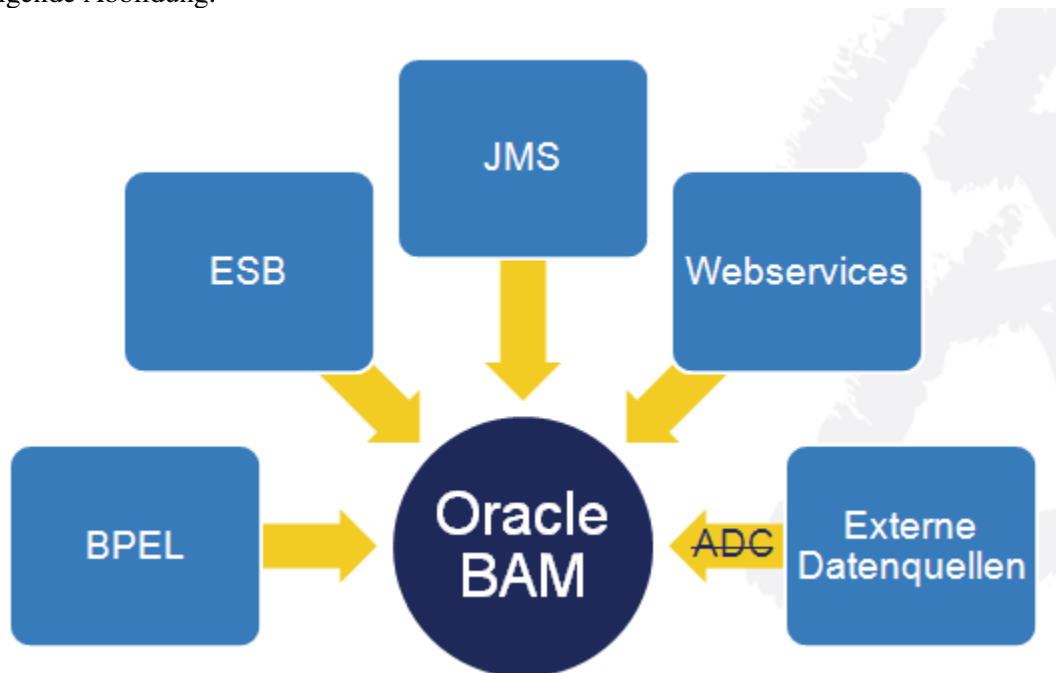


Abbildung 1: Möglichkeiten des Datenflusses nach Oracle BAM

Daten können neben den Anbindungen von BPEL aus auch über einen BAM-Adapter am ESB, JMS-Queues, Webserviceschnittstellen und aus externen Datenquellen (zum Beispiel Tabellen aus einem Microsoft SQL Server) eingebunden werden. Allerdings funktioniert bei den externen Datenquellen nicht der Aktive Daten Cache (ADC), der genutzt werden kann, um Berichte in Echtzeit mit neu eintreffenden Daten zu aktualisieren.

Insgesamt beurteile ich den Kosten/Nutzen-Faktor von Oracle BAM an dieser Stelle sehr positiv, während es sich vom Umfang der Möglichkeiten irgendwo zwischen einer Eigenentwicklung und einer vollständigen BI-Lösung einordnet.

Funktionsweise und Struktur von Oracle BAM

Die Grundstruktur von Oracle BAM weist einige Übereinstimmungen mit den Grundstrukturen auf, die man typischerweise im Bereich Business Intelligence und Data Warehouses findet. Es gibt eine Datenbasis in Form von modellierten und miteinander verknüpfbaren Datenobjekten, die aus dem Mediator, aus Prozessen oder auf anderen Wegen manipuliert werden können (Insert, Update, Delete und Upsert). Aufbauend auf diesen Datenobjekten, die damit die Schnittstelle zwischen Oracle BAM und dem Rest der Welt darstellen, können dann Berichte und Alarmer definiert werden. Im Folgenden sollen die Grundstrukturen von Oracle BAM an einfachen Beispielen aufbauend auf einer Fehlerüberwachung gezeigt werden.

Datenobjekte

Ein Datenobjekt sieht zuerst aus wie eine Tabelle in der Datenbank (und wird intern auch auf eine solche abgebildet). Es besteht aus Spalten, die verschiedene Datentypen und Eigenschaften haben dürfen. In *Abbildung 1* ist einmal das Layout eines einfachen Datenobjektes dargestellt.

Feldname	Feld-ID	Feldtyp	Max. Länge	Nachkommastellen	Auf Null setzbar	Öffentlich	Lookup	Berechnet
ID	_ID	auto-incrementing integer	-	-	No	Yes	-	-
Fehlercode	_Fehlercode	integer	-	-	No	Yes	-	-
Fehlermeldung	_Fehlermeldung	string	100	-	No	Yes	-	-
TransaktionsID	_TransaktionsID	integer	-	-	No	Yes	-	-
Datum	_Datum	datetime	-	-	No	Yes	-	-
Prozess	_Prozessschritt	string	100	-	Yes	Yes	-	-
Prozessschritt	_Verantwortlich	string	100	-	Yes	Yes	-	-
DatumString	_DatumString	string	100	-	Yes	Yes	-	concat(concat(concat(concat(DayOfMonth(Datum), ','), Month(Datum)), ','), Year(Datum)))
TimeString	_TimeString	string	100	-	Yes	Yes	-	concat(concat(concat(concat(Hour(Datum), ':'), Minute(Datum)), ':'), Second(Datum)))
Verantwortlich	_Verantwortlich1	string	100	-	Yes	Yes	-	-

Abbildung 2: Beispiels-Layout eines Datenobjektes

Das obige Beispiel entspricht einem Datenobjekt aus einer der vorgenannten möglichen zentralen Sammel- und Verteilstelle für Fehlermeldungen. Auf Basis dieses Datenobjektes (und auch jedes anderen vorhandenen Datenobjektes) können dann Berichte und Alarmer angelegt werden. So könnten in diesem Falle die Fehlermeldungen per Mail verteilt werden und spezielle Dashboards für Fehler-Übersichten eingerichtet werden, wo der Fachbereich oder andere Anwender weder einen Zugang noch das entsprechende technische Wissen für das Benutzen des Enterprise Managers benötigen, um sich einen schnellen Überblick über das System zu verschaffen.

Berichte

Auf einem Datenobjekt können nun verschiedenste Berichte angelegt werden. Bei den Berichten gibt es ganz unterschiedliche Möglichkeiten, wie man verschiedenste Diagramm und Listentypen auf einem Bericht miteinander verbinden kann, um damit aussagekräftige Berichte zu erzeugen.

Ein sehr einfaches Beispiel könnte eine Übersicht sein, wie viele Fehler pro Tag aufgelaufen sind.

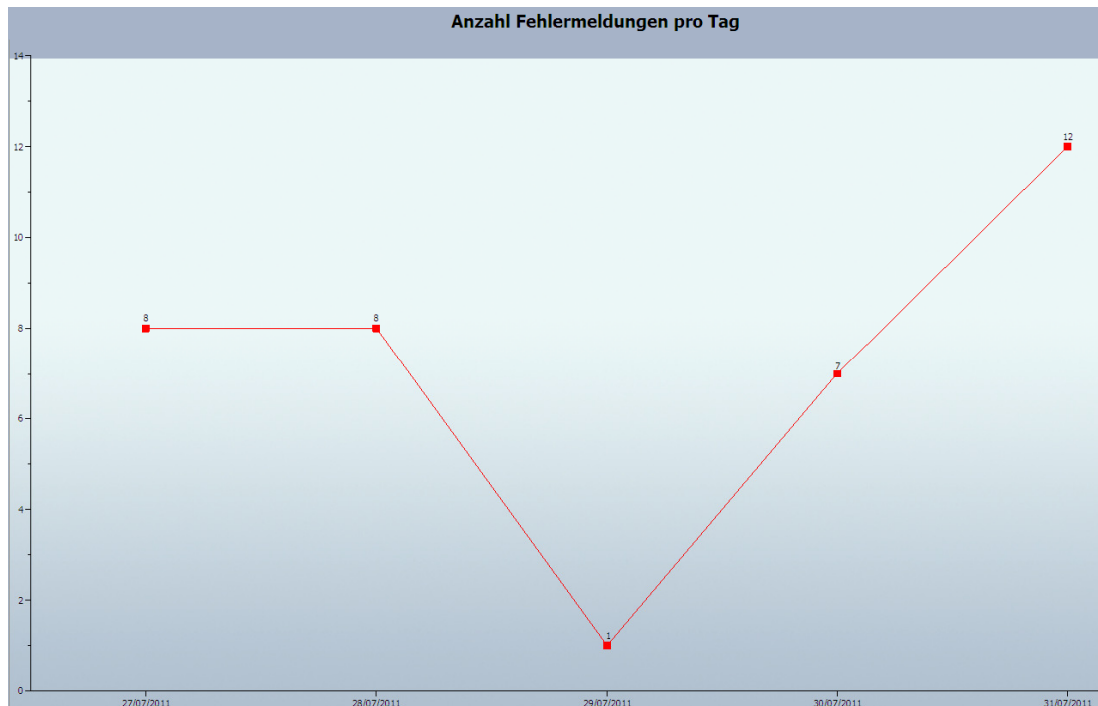


Abbildung 3: Anzahl Fehler pro Tag

Diese erste Übersicht für sich genommen mag noch nicht sehr aussagefähig sein (abgesehen davon, dass der 29. Juli ein sehr ruhiger Tag gewesen sein muss), betrachtet man aber noch einmal die Zuständigkeiten bzw. die Bereiche, in denen die Fehler auftreten (siehe *Abbildung 4*), kann man schnell auf den Gedanken kommen, mal in Richtung des Netzwerkes etwas zu verbessern.

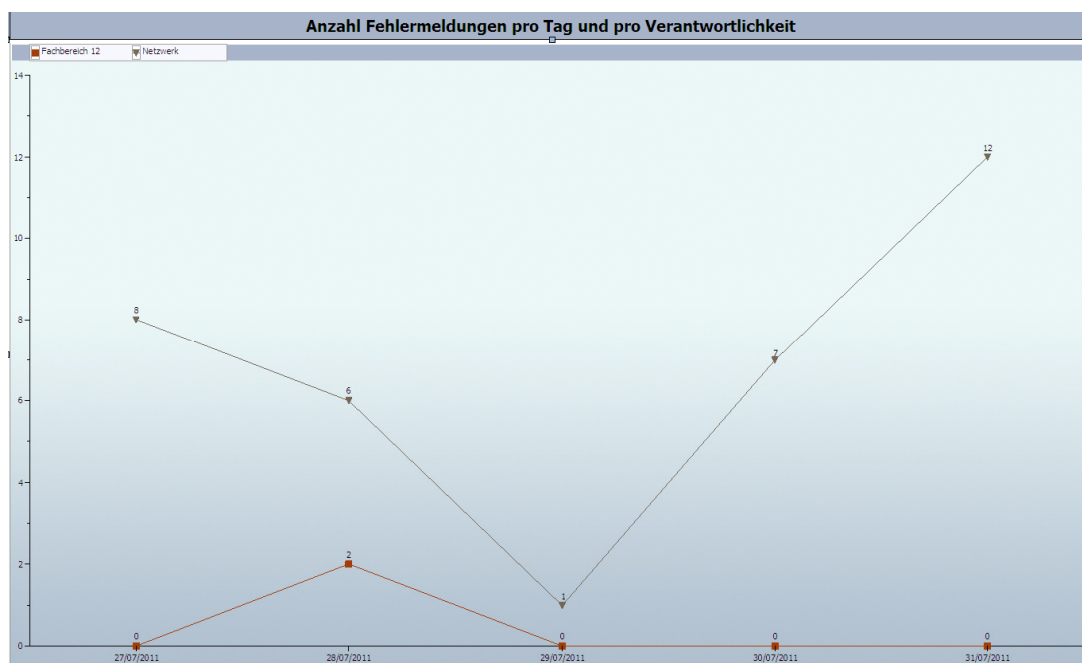


Abbildung 4: Anzahl Fehler pro Tag und Verantwortlichkeit

Alarme

Über Alarme können bei Eintreten von definierbaren Ereignissen bestimmte Aktionen ausgeführt werden. So könnte in vorliegendem Falle ein Alarm definiert werden, der dafür zuständig ist, jede Fehlermeldung, wo die Verantwortlichkeit bei dem Netzwerk liegt, auch per Mail an einen entsprechenden Empfänger zu versenden. Andere Aktionen könnten der Aufruf eines Webservice oder das Starten eines ODI-Szenarios sein. In *Abbildung 5* wird ein konkretes Beispiel eines Alarmes gezeigt.

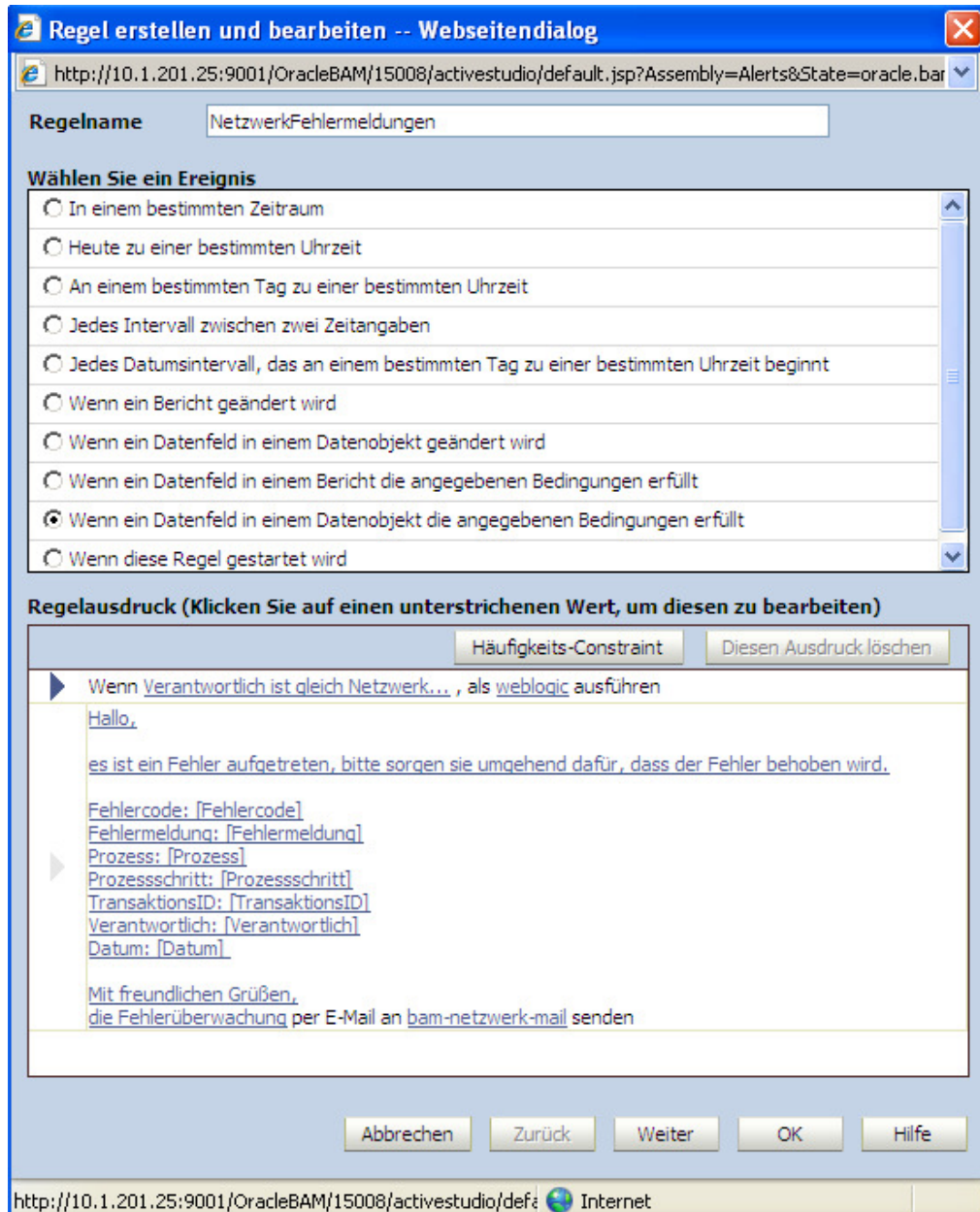


Abbildung 5: Beispiel eines Alarmes

Grundsätzlich gibt es für die Ereignisse zwei unterschiedliche Kategorien, nämlich zeitbasiert und datenbasiert. Die dann möglichen Ereignistypen sieht man schön in *Abbildung 5*. Eine gewisse Herausforderung ist die Kombination von Ereignissen aus beiden Kategorien und mindestens einmal wirklich herausfordernd wird es, wenn man eine Aktion auslösen möchte, weil gerade etwas NICHT passiert ist (der aktive Daten Cache reagiert ja nur auf stattfindende Änderungen oder eintreffende Meldungen), weil zum Beispiel auch das Eintreffen von keiner einzigen Bestellungen von einem Shop-System an ein ERP-System an einem normalen Wochentag ein mögliches Problem aufzeigen könnte.

Monitor Express

Wie schon vorher beschrieben, versteckt sich hinter dem Stichwort Monitor Express die Möglichkeit von Performance-Überwachungen von BPEL-Prozessen. Diese kann mit wenigen Einstellungen für einen BPEL-Prozess eingerichtet werden und die benötigten Datenobjekte sowie erste Berichte liegen als Beispiel von Oracle vor und können auf einfache Weise eingespielt werden.

Betrachten wir im Folgenden mal ein Beispiel, wo ein Prozess gestartet wird, der dann hintereinander die folgenden Schritte ausführt:

1. Validierung der Eingabedaten bei Prozessstart
2. Senden eines Aufrufes an ein System A
3. Validierung der Antwort von System A
4. Senden eines Aufrufes an ein System B
5. Validierung der Antwort von System B
6. Schreiben eines Gesamtergebnisses in eine Datenbank

Dann könnte bei entsprechender Einrichtung ein Teil von einem der mitgelieferten Berichten wie folgt aussehen:

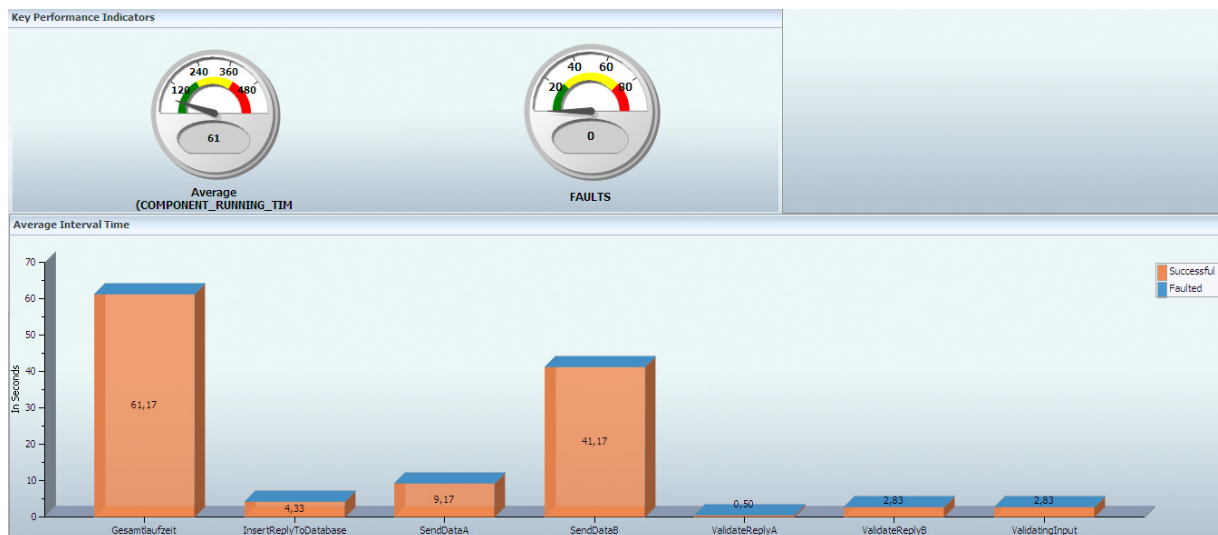


Abbildung 6: Ausschnitt von einem Monitor Express Bericht

Aus diesem Diagramm ist zum Beispiel schnell ersichtlich, dass über zwei Drittel der Gesamtlaufzeit des Prozesses für das Übermitteln und Verarbeiten der Daten an/in System B aufgewendet wird. Hier

könnte man näher untersuchen, ob dies am Netzwerk und an weiteren Optimierungspotentialen in System B liegen kann.

Als nächstes sollen kurz die wenigen Schritte gezeigt werden, die benötigt werden, um Monitor Express einzusetzen. Dazu muss man im JDeveloper den BPEL-Prozess öffnen und zuerst in die Monitor-Perspektive wechseln. Das Wechseln geschieht rechts oben, wobei es teilweise (je nach Version) entweder verschiedene Buttons für die normale BPEL und Monitor-Perspektive gibt, oder es gibt nur einen Button, der wie eine Dropdown-Liste funktioniert. In der Monitor-Perspektive gibt es dann links oben ein kleines Symbol über das sich ein Fenster öffnet, über welches man in ein Fenster kommt, wo man neben dem allgemeinen Aktivieren von Monitor Express (hier heißt es allerdings nur Activity Monitoring) auch noch die Granularität der Performance-Überwachung festlegen kann. Zumeist empfiehlt sich die Überwachung von „Scopes und Human Tasks only“ (entsprechende Nutzung von Scopes vorausgesetzt, was wir aber allgemein empfehlen), da die Datenmenge, die sich bei Überwachung von allen Aktivitäten ergibt, mitunter nur schwer überschaubar ist.

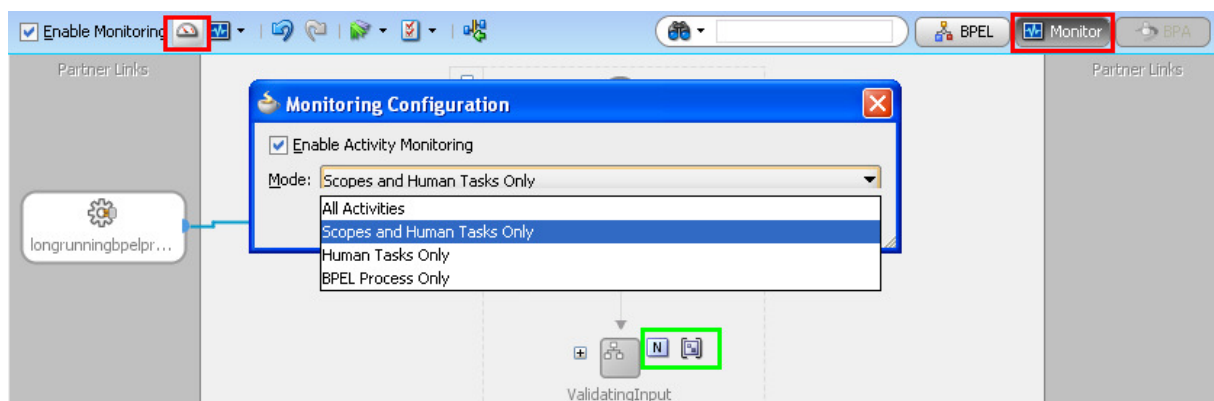


Abbildung 7: Einrichten von Monitor Express in BPEL

Neben diesen allgemeinen Einstellungen können dann noch eigene Zähler oder Intervallmessungen eingefügt werden, diese erscheinen dann als zusätzliche Symbole (siehe grün gekennzeichneten Bereich) in der Prozessdarstellung. Der Punkt „Gesamtlaufzeit“ aus dem vorhergehenden gezeigten Bericht war übrigens ein selbst definiertes Intervall, welches von der Aktivierung bei „ValidatingInput“ bis zum Ende von „InsertReplyIntoDatabase“ gelegt wurde.

Und zum Schluss können auch noch Business Indicators hinzugefügt werden, das bedeutet nichts anderes als das Speichern von einfachen oder zusätzlich berechneten Werten zu bestimmten Zeitpunkten des BPEL-Prozess nach Oracle BAM. Leider sieht es zum momentanen Zeitpunkt noch nicht so aus, als wenn diese Business Indicators auf einfache Art und Weise mit den Laufzeitmessungen verbunden werden können, um zum Beispiel auch Laufzeiten in Verbindung mit den Start- bzw. Eingabewerten des Prozesses festhalten zu können.

Ebenfalls unter den durch Oracle schon zur Verfügung gestellten Berichten findet sich folgende Teilansicht, anzumerken ist, dass in der „Process Usage“-Übersicht nicht die Laufzeiten, sondern nur die Anzahl der Aufrufe einfließt.

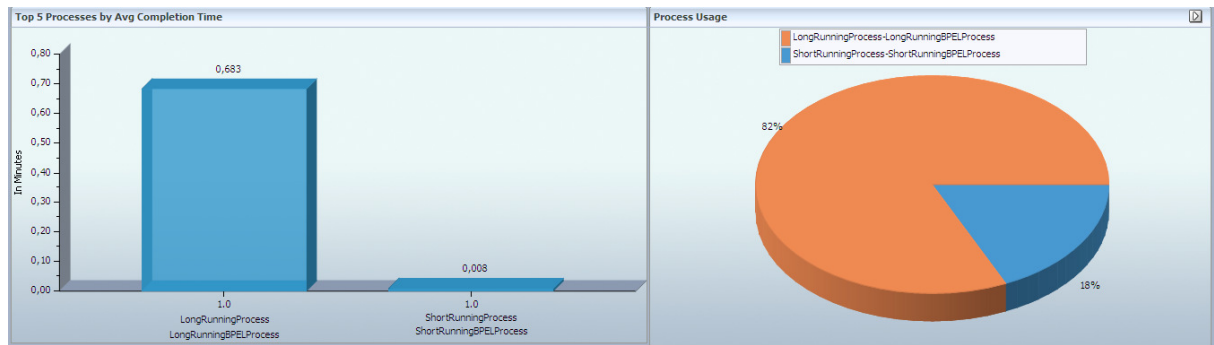


Abbildung 8: Ausschnitt aus einem Monitor Express Berichtes

Die Gegenüberstellung der anteiligen Gesamtlaufzeiten kann aber schnell als eigener Bericht auf den vorhandenen Datenobjekten erstellt werden und könnte wie unten dargestellt aussehen.

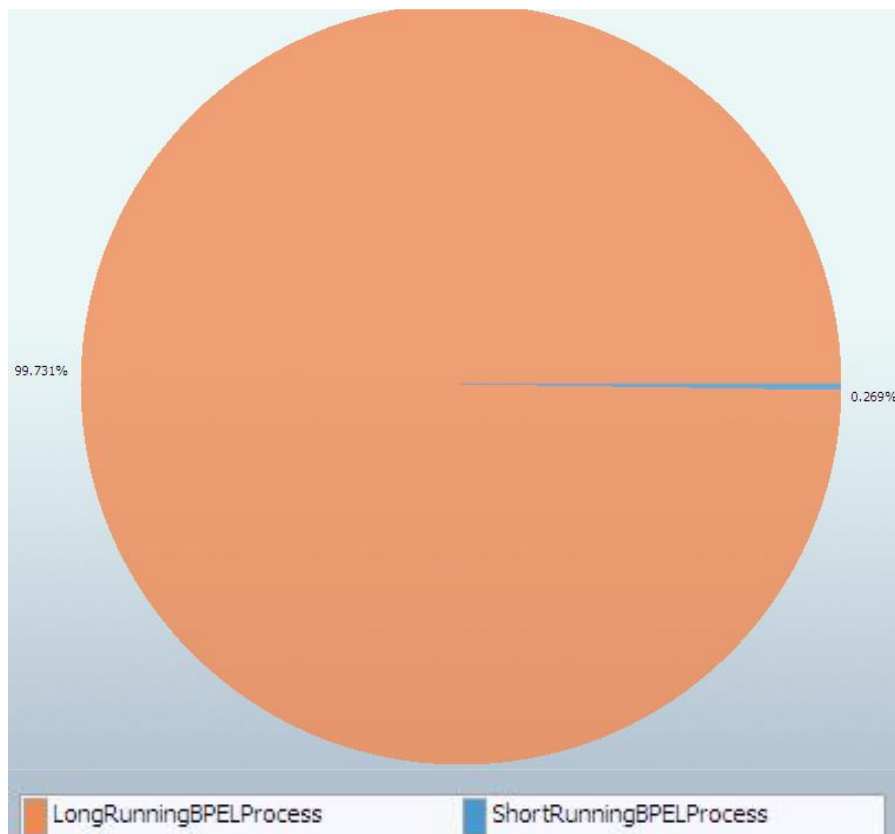


Abbildung 9: Anteilige Gesamtlaufzeiten nach Prozessen

Abschluss

Ich hoffe, dass ich mit dieser Ausarbeitung und dem noch folgenden Vortrag den Einsatz von Oracle BAM motivieren kann. Der Nutzen für das Prozess- und Performance Controlling ist sicher unbestritten. Insgesamt bietet Oracle BAM in Zusammenarbeit mit der Oracle SOA Suite einen großen Mehrwert bei überschaubarem zusätzlichem Aufwand. Oracle BAM lässt sich von den Daten her gut in die Oracle SOA Suite und andere Systeme integrieren und bietet vielfältige Möglichkeiten mit Berichten und Alarmen für verschiedene technische und fachliche Überwachungsmöglichkeiten. Es bringt von Hause aus den Zugang über eine Weboberfläche und eine über Benutzerzugänge gesteuerte Sicherheit und Sichtbarkeit der gespeicherten Informationen mit. Etwas mehr technische Details finden sich in dem DOAG-Vortrag „Volle Übersicht über Metadaten und Prozessdaten mit Oracle BAM“ von 2010, der als Ergänzung zu diesem Vortrag empfohlen wird.

Zusätzlich hoffe ich, dass ich insgesamt deutlich machen konnte, dass Oracle BAM auch eine BI-Anwendung ist, wobei der BI-Aspekt in dem Falle wesentlich von den Daten abhängt, die nach Oracle BAM geschickt werden bzw. die durch die Oracle SOA Suite laufen. Liegen die richtigen Daten vor, kann man schnell Echtzeit-Berichte erstellen, die zum Beispiel Bestellungen nach Regionen oder Niederlassungen oder Umsatz nach Warengruppen anzeigen.

Kontaktadresse:

Dr. Markus Vincon
Opitz Consulting Essen GmbH
Altendorfer Straße 3
D-45127 Essen

Telefon: +49 (0) 201- 892994 1724
Fax: +49 (0) 201- 892994 4700
E-Mail: markus.vincon@opitz-consulting.de
Internet: www.opitz-consulting.de