



# OWB2ODI-Migration – eine Operation am Herzen des ETL

Achim Stump, Christian Lenzhölzer und Oliver Hüskes, OPITZ CONSULTING Deutschland GmbH

*Man hatte schon so ein Bauchgefühl ... und dann passiert es: Der Hausarzt stellt bei einer Routine-Untersuchung fest, dass das Herz Alterserscheinungen aufweist und die Lebenserwartung nur noch wenige Jahre betragen kann. Der daraufhin konsultierte Spezialist rät zu einer komplexen und riskanten Transplantation. Der Schock und die Verunsicherung wären in dieser Situation nicht auszudenken – und sind mit der Verunsicherung im Vorfeld einer Migrationsvorhabens keineswegs gleichzusetzen. Doch es gibt einige Parallelen bei den Rahmenbedingungen beider Szenarien.*

Szenarien, die auf Anhub bedrohlich erscheinen, werden häufig berechenbarer, wenn man sie genauer betrachtet. Man stelle sich vor, den Hausarzt durch die IT-Abteilung des Unternehmens und das menschliche Herz durch das Herz des ETL-Systems, den Oracle Warehouse Builder (OWB), zu ersetzen. In diesem Fall wäre glücklicherweise kein Menschenleben bedroht und statt einer gefährlichen Herztransplantation stünde lediglich eine Migration zum Oracle Data Integrator (ODI) auf dem Plan. Aber auch diese Aussicht lässt so manchem BI-Verantwortlichen das Blut in den Adern gefrieren. Um den ersten Schock zu überwinden, betrachte man das Problem einmal ganz nüchtern.

Mithilfe analytischer Planung, sorgfältiger Vorbereitung und der Erfahrung eines Spezialisten lässt sich das Risiko in vielen

Fällen auf ein kleinstmögliches Maß reduzieren. Um die ETL-Operation einmal ganz ruhig anzugehen, wird die Migration vom OWB auf den ODI auf drei grobe Phasen reduziert (siehe *Abbildung 1*):

- **Phase 1**  
Analyse von Situation und Möglichkeiten
- **Phase 2**  
Konzeption und Planung von Vorgehensweise und Maßnahmen
- **Phase 3**  
Umsetzung der geplanten Maßnahmen

In den folgenden Abschnitten werden für jede dieser Phasen Fragen und Aspekte genannt, die helfen können, das für die jeweilige Situation am besten geeignete Vorgehen zu finden. Man sollte jedoch nicht die

Ernsthaftigkeit der Problemlage unterschätzen und sich ausreichend Zeit für jede „Behandlungsphase“ nehmen. Wie in der Humanmedizin kann jeder Krankheitsverlauf unterschiedlich sein und jeder Patient sollte individuell behandelt werden.

## **Phase 1: Analyse der Rahmenbedingungen**

Lohnt sich eine Migration zum ODI überhaupt noch? Diese Frage sollte man sich zunächst stellen. Ein Austausch der ETL-Strecken stellt immerhin eine erhebliche Investition und ein Risiko dar. Es ist daher zu prüfen, wie lange die Applikation, die über den ETL-Prozess beliefert wird, noch betrieben werden soll. Der Premier Support für den OWB läuft noch bis zum Jahr 2018. Wenn die Applikation davor (oder kurz danach) außer Betrieb gestellt werden sollte,

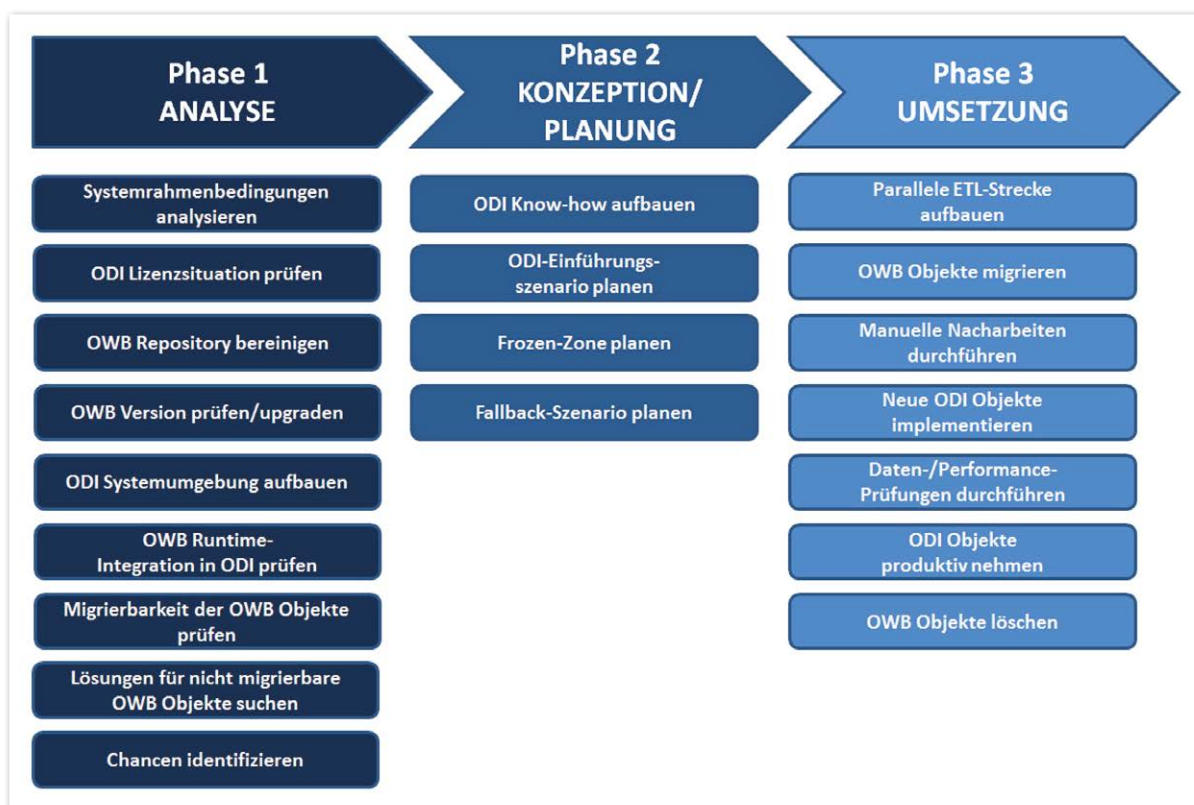


Abbildung 1: Phasen eines ODI-Migrationsprojekts

lohnt sich eine vorherige Migration zum ODI natürlich nicht. Der Verzicht auf die Migration hätte allerdings eine kleine Nebenwirkung zur Folge: Man müsste auf ein Upgrade der zugrunde liegenden Oracle-Datenbank auf 12.2 ff. verzichten und könnte eventuelle Features der neuen Version nicht nutzen.

Zur Lizenzsituation: Die Grundversion des OWB ist in der Oracle-Datenbank enthalten und verursacht damit keine zusätzlichen Lizenzkosten. Bei einer Migration zum ODI müssen neben den vorhandenen Datenbank-Lizenzen separate ODI-Lizenzen erworben werden. Werden hingegen jetzt schon Features des OWB genutzt, die der ODI-EE-Lizenzpflicht unterliegen, ändert sich nichts.

Um den Umfang der OWB-Migration auf das Notwendige zu reduzieren, sollte man versuchen, das OWB-Repository um nicht mehr genutzte Objekte zu bereinigen. Dafür sind verschiedene Abfragen auf das Repository hilfreich, um zum Beispiel mit einer Datenbank-Abfrage die OWB-Objekte zu identifizieren, die seit einem längeren Zeitraum nicht mehr ausgeführt wurden (<http://bs.doag.org/go/businessnews/201502/lis-ting1>). Anschließend wird geprüft, ob die gefundenen Objekte stillgelegt oder gelöscht werden können. Dazu vorher unbedingt eine Sicherung der betroffenen Objekte erstellen.

Ist die OWB-Version überhaupt migrierbar? Dazu gilt es zunächst herauszufinden, welche konkrete Version des OWB installiert ist. Für eine automatische Migration von OWB-Objekten zum ODI liefert Oracle ein Migrations-Tool aus, das ausschließlich auf 64-bit-Windows oder Linux X86 läuft. Das Tool gibt es nur unter OWB 11.2.0.3 mit Patch 18537222 oder 11.2.0.4 mit Patch 18537208. Wer bisher eine andere OWB- oder eine andere Betriebssystem-Version im Einsatz hat, muss vorher noch eine Zwischenmigration durchführen, um auf das erforderliche Sprungbrett in Richtung ODI zu kommen. Ziel der Migration ist ODI 12.1.2 oder 12.1.3.

Zum Aufbauen der ODI-Systemumgebung installiert man die aktuelle Version (derzeit 12.1.3.0.0) des ODI mit den zugehörigen Patches, legt über das Repository Creation Utility (RCU) ein Master- und Arbeits-Repository an und richtet den ODI-Agenten ein. Es ist zu überlegen, welche weiteren ODI-Repositories für die Systemumgebung sinnvoll sind, und man definiert ein Deployment-Verfahren.

Ab ODI 12c ist eine direkte Integration von OWB Mappings/OWF Process Flows in ODI-Packages möglich. Dazu legt man das aktuelle OWB-Runtime-Repository in der ODI-Topologie an. Die OWB Mappings/OWF Process Flows lassen sich anschlie-

ßend über die Komponente „OdiStartOwb-Job“ direkt in den ODI-Packages verwenden. Bei einer schleichenden Migration könnte die Ablaufsteuerung sofort und ausschließlich über ODI erfolgen, obwohl die einzelnen OWB-Objekte erst zu einem späteren Zeitpunkt überführt werden. Ebenso besteht die Möglichkeit, im ersten Schritt lediglich die OWF Process Flows manuell durch ODI-Packages zu ersetzen, die allerdings weiterhin OWB-Mappings aufrufen. Es empfiehlt sich, die „Side by Side“-Möglichkeiten an eigenen Beispielen zu prüfen und zu entscheiden, ob und wie man diese Funktionalität auf dem Migrationsweg nutzen möchte.

Anschließend ist die Migrierbarkeit der OWB-Objekte zu prüfen. Das Migrations-Tool kann viel, doch es ist nicht in der Lage, alle OWB-Objektarten zu migrieren (siehe „<https://docs.oracle.com/middleware/1213/odi/install-migrate-owb-odi/understanding.htm#ODIMG148>“). Das Tool kann in drei verschiedenen Modi ausgeführt werden:

- FAST\_CHECK prüft lediglich, welche OWB-Objekte theoretisch migriert werden können.
- DRY\_RUN führt die Migration durch, aber ohne Commit in das ODI-Repository. Das



## Digitale Transformation mit Durchblick

Profitieren Sie von unserem Know-how in der Digitalisierung intelligenter Geschäftsprozesse mit Oracle Applikationen, Technologien und Cloud Services:

- Enterprise Cloud Services (SaaS) für Oracle EPM, ERP, CX, HCM
- Oracle E-Business Suite
- Hyperion EPM
- Oracle Fusion Middleware (PaaS)

Unser Alleinstellungsmerkmal: Als Oracle Pionier und Platinum Partner bieten wir seit über 20 Jahren erfolgreiche Projektarbeit im gehobenen Mittelstand und in global tätigen Großunternehmen. Erfahren Sie mehr...

...und besuchen Sie uns vom  
**9. - 10. Juni 2015**  
auf unserem Stand!

# PROMATIS



PROMATIS software GmbH  
Tel.: +49 7243 2179-0  
Fax: +49 7243 2179-99  
www.promatis.de · hq@promatis.de  
Ettlingen/Baden · Hamburg · Berlin

Log enthält mehr Informationen.

- RUN führt die Migration mit Commit durch.

Das Tool hat weitere Schwachstellen. Es liefert bestimmte Informationen nicht, so listet es einige Repository-Objekte nicht auf oder berücksichtigt bestimmte Zusatz-Informationen zu Objekten nicht. Aus der Oracle-Dokumentation ist ersichtlich, dass unter anderem Tabellen-Partitionen nicht migriert werden, ohne dass das Migrations-Tool darauf hinweist. Deshalb ist es schwierig abzuschätzen, ob und wie stark man von den Einschränkungen betroffen ist; man muss daher unter Umständen einen erheblichen manuellen Analyse-Mehraufwand in Kauf nehmen.

Um dieses Defizit zu umgehen, empfiehlt es sich, ergänzend zum Migrations-Tool noch manuell ein SQL-Skript zu schreiben, das eine zusätzliche Überprüfung des OWB-Repository ermöglicht. Welche Objekte migrierbar sind und welche nicht, steht in der Oracle-Dokumentation. Es ist ratsam, zusätzlich auf die Best Practices erfahrener Experten zurückzugreifen.

Mithilfe des eigenen Analyse-Skripts lässt sich beispielsweise durch Ermittlung der verwendeten Transformationen die Komplexität von Mappings erheblich besser einschätzen. Diese Informationen liefert das Migrations-Tool nämlich ebenfalls nicht. Die bloße Meldung, dass ein Mapping migriert wurde, hat keine große Aussagekraft. Interessanter ist, wie viel Aufwand im Nachgang zu erwarten ist.

Die Abfrage auf das OWB-Repository liefert in diesem Fall einen direkten Informationsgewinn für die Planung und Budgetierung des Projekts. Wer allerdings nur Standard-Funktionalitäten nutzt und nur eine Handvoll einfacherer Mappings betreibt, kann auf ein Analyse-Skript getrost verzichten, hier reichen die Möglichkeiten des Migrations-Tools.

### Lösungen für nicht migrierbare OWB-Objekte

Es muss sichergestellt sein, dass zu allen Problemen, die von der Migrierbarkeits-Analyse identifiziert wurden, eine Lösung oder ein Workaround existiert. Bei nicht migrierbaren Objekten sollte man gegebenenfalls eine direkte Neu-Implementierung im ODI in Betracht ziehen und die Realisierbarkeit über geeignete Prototypen sicherstellen. Exemplarisch zwei Probleme mit einer kurzen Lösungsbeschreibung:

- **Problem 1**  
Es existieren OWF Process Flows, denn diese können nicht migriert werden.
- **Lösung**  
Hier wären entweder vorläufig ein direkter Aufruf der OWF Process Flows in ODI oder die finale Neu-Implementation als ODI-Package möglich.
- **Problem 2**  
Es werden Variablen oder Konstanten verwendet, die der OWB beim Deployment erzeugt (wie „get\_model\_name“ oder „get\_audit\_id“)
- **Lösung**  
Entsprechende Konstrukte des ODI-Substitution-API sind zu verwenden

Es hat sich bewährt, diese Tätigkeiten in einem gemischten Entwicklerteam von OWB- und ODI-Experten durchzuführen.

Zusätzliche Chancen der Migration: Eine ODI-Migration bietet die Möglichkeit, grundsätzliche Dinge einer aktuellen Lösung zu hinterfragen und nach Verbesserungen zu suchen. So sollte man sich überlegen, welche Dinge in der bisherigen Lösung gestört haben oder lediglich über Workarounds gelöst werden konnten. Diese lassen sich jetzt im Zuge des Migrationsvorhabens optimieren. Ein Beispiel wäre die Automatisierung des Deployment-Prozesses, um den Weg zu einer agilen Vorgehensweise zu ebnen oder Mappings durch bestimmte ODI-Knowledge-Module zu vereinfachen.

Man sollte sich auch überlegen, bei dieser Gelegenheit neue Architektur- beziehungsweise Modellierungsansätze wie Data Vault einzuführen, um Synergieeffekte zu erzielen, wenn die Analysen nahelegen, ETL-Strecken teilweise komplett neu zu erstellen. Diese Chance ist allerdings mit dem Risiko eines zusätzlichen Daten-Migrationsprojekts abzuwägen.

### Phase 2: Konzeption und Planung

Die Architektur und Vorgehensweise zur Erstellung von ETL-Strecken und die Begrifflichkeiten in OWB und ODI sind teilweise sehr unterschiedlich. Auch wenn erfahrene OWB-Entwickler sich relativ schnell in der neuen Umgebung zurechtfinden, werden sie beim Umstieg auf den ODI Einarbeitungszeit und Unterstützung benötigen. Sie sollten sich deshalb intensiv mit dem allgemeinen Konzept von Knowledge-Modulen und den mitgelieferten Standard-Knowledge-Modulen auseinandersetzen und frühzeitig Budget und

genügend Zeit für ein Selbststudium oder Schulungen einplanen.

Es gibt verschiedene Möglichkeiten, ODI-Objekte produktiv zu nehmen, die in das Testsystem migriert wurden. In den meisten Migrationsprojekten hat sich ein iterativ-inkrementelles Vorgehen als zielführend erwiesen. Dazu identifiziert man Gruppen von ETL-Objekten, bei denen eine gleichzeitige Produktivnahme sinnvoll oder vielleicht sogar notwendig ist. So können beispielsweise alle ETL-Objekte eines bestimmten Layers oder alle ETL-Objekte zu einem bestimmten fachlichen Bereich eine Gruppe bilden.

Anschließend plant man Aktivitäten und setzt Meilensteine für diese Gruppen. Ziel jeder Iteration ist ein ausführbares Inkrement beziehungsweise die erfolgreiche Inbetriebnahme einer migrierten ETL-Objektgruppe. Die erste Iteration sollte nur

wenige Objekte enthalten, um die Vorgehensweise kennenzulernen und Erfahrungen bei der Produktivnahme zu sammeln (siehe Abbildung 2).

Es ist sicherzustellen, dass abhängig vom gewählten ODI-Einführungsszenario während der Migrationsvorhaben keine Änderungen an den ETL-Strecken durchgeführt werden. Dazu legt man zusammen mit dem Systemanwender einen „Frozone-Zeitraum“ fest und definiert Ausnahmen (etwa für Bug-Fixing).

In jedem Fall ist ein Fallback-Szenario einzuplanen. Egal, wie gut die Migration geplant ist, etwas Unvorhersehbares kann immer dazwischenkommen. Mithilfe des Fallback-Szenarios sollte man den Produktivbetrieb im Notfall schnellstmöglich wieder aufnehmen können. Eine flexible ETL-Laufzeitsteuerung schafft eine geeignete und schnelle Fallback-Möglichkeit.

### Phase 3: Umsetzung

Zum Migrieren der OWB-Objekte startet man das Migrationstool und prüft die Einträge in der Log-Datei. Das Migrationstool überführt die Objekte des OWB-Repository in das angegebene ODI-Work-Repository. Dabei lassen sich alle oder nur bestimmte Objekte eines OWB-Repository in das ODI-Repository überführen. Eine vollständige initiale Migration ist zu empfehlen.

Nach der automatischen Migration fallen Nacharbeiten an, die in der Analyse-Phase identifiziert und erprobt wurden. Dazu gehört die Prüfung der verwendeten Knowledge-Module und ihrer Optionen – möglicherweise sind diese auszutauschen oder anzupassen.

Im ODI werden die einzelnen Schritte der Mappings erst zur Laufzeit interpretiert und ausgeführt. Die Korrektheit eines Mappings kann man nur bei der Ausführung prüfen. Dabei entsteht im Simulationsmodus ein ausführliches Log über die Kommandos, die in den einzelnen Tasks ausgeführt würden.

Neue ODI-Objekte werden für migrierbare OWB-Objekte sowie für Bereiche nicht implementiert, die vollständig neu aufgebaut werden sollen. Es ist auch zu überlegen, ob hybride Übergangslösungen mit OWB- und neuen ODI-Objekten als Übergangslösung sinnvoll sein können.

Für jedes fertige ETL-Objekt sind ausgiebige Tests erforderlich. Wenn möglich, baut man eine parallele ETL-Strecke mit Tabellen und produktionsnahen Daten in einem separaten Datenbank-Schema auf. In dieser Umgebung lassen sich in einem Regressionstest vor der Produktivnahme Daten und Ausführungszeiten migrierter ODI-Objekte mit den produktiven Ergebnissen vergleichen. Ob ein vollständiges Duplikat der Produktionsumgebung angelegt wird oder nur eines von einzelnen ETL-Strecken, ist abhängig vom gewählten ODI-Einführungsszenario und von der Komplexität und dem Datenvolumen der Produktionsumgebung (siehe Abbildung 3).

Ein entscheidender Punkt wird in Migrationsprojekten oftmals unterschätzt: Die Validierung der ODI-Mappings ist mit erheblichem Aufwand verbunden. Funktionieren diese so wie die vorigen OWB-Mappings? Dies ist unbedingt zu überprüfen, denn die Erfahrung hat gezeigt, dass dies bei einigen migrierten Mappings nicht der Fall ist.

Die zugrunde liegenden Laufzeit-Architekturen von OWB und ODI sind unterschiedlich und verlangen eine besondere Beachtung. OWB-Mappings werden ausschließlich als PL/SQL-Logik in der Oracle-Datenbank abgelegt

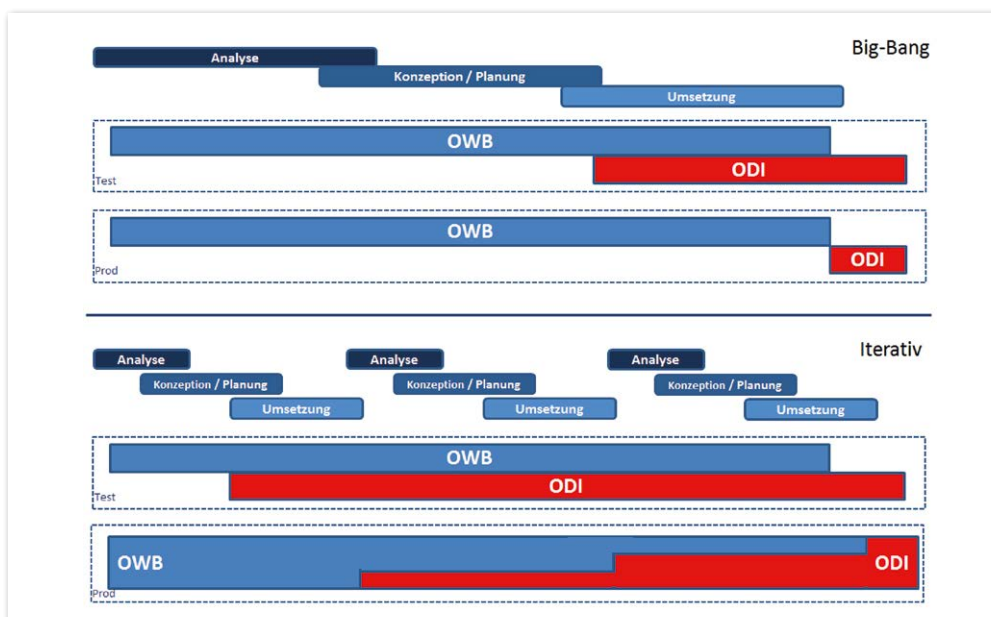


Abbildung 2: ODI-Einführungsszenarien

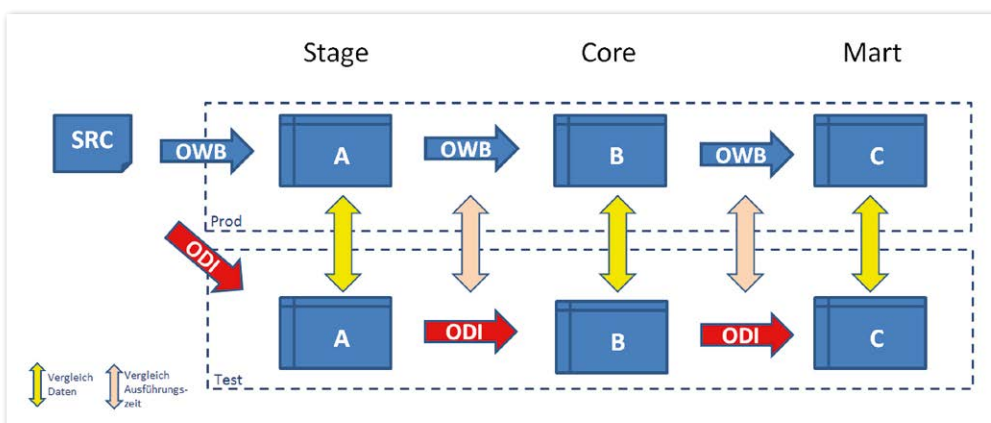


Abbildung 3: Aufbau einer parallelen ETL-Teststrecke

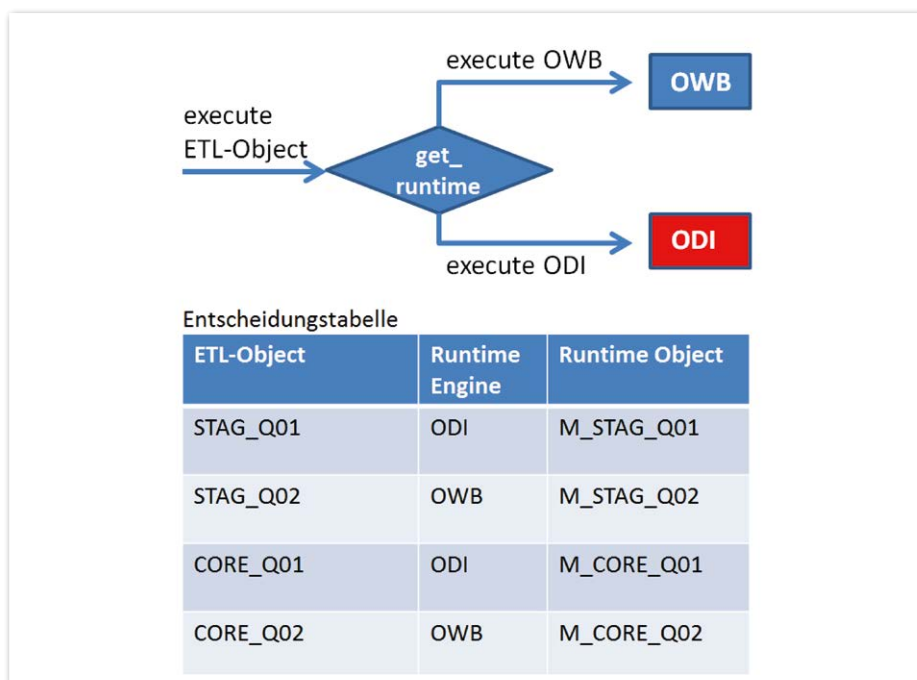


Abbildung 4: Dynamische Laufzeitsteuerung von ETL-Objekten

und ausgeführt, die ODI-Mappings teilweise direkt in der Datenbank, teilweise im Application-Server.

Bei lastintensiven ETL-Strecken ist beim OWB ausschließlich die Oracle-Datenbank für eine geeignete Skalierung verantwortlich. Beim Einsatz von ODI verteilt sich die Skalierung auf die Datenbank und den Application-Server. Nach einer Migration können sich somit das Verhalten und die Performance einer ETL-Strecke signifikant ändern. Hier sollten geeignete Prüfmechanismen, wie die Prüfung der Ausführungs-

dauer, vorgesehen werden. *Listing 2* und *Listing 3* zeigen SQL-Skripte zur Ermittlung der Ausführungszeit eines OWB-Mappings und eines ODI-Mappings (<http://bs.doag.org/go/businessnews/201502/listing2> und <http://bs.doag.org/go/businessnews/201502/listing3>).

Entsprechend des geplanten ODI-Einführungsszenarios werden neue ODI-Objekte in die Produktion übernommen. Bei einer schleichenden Migration bietet eine eigenentwickelte, dynamische ETL-Laufzeitsteuerung die notwendige Flexibilität. Diese entscheidet, für

welche ETL-Strecke welche ETL-Technologie (ODI/OWB) eingesetzt werden soll (*siehe Abbildung 4*). In der Anfangszeit sind verstärkt die Ausführungszeiten und stichprobenartig die erstellten Daten zu überprüfen.

Das eingangs genannte Skript (*Listing 1*) prüft, ob die alten OWB-Objekte noch verwendet werden. Um sicherzugehen, dass diese tatsächlich nicht mehr aufgerufen werden, sollte man sie sukzessive löschen. Zuvor ist jedoch unbedingt für ein Backup mit Recovery-Möglichkeit zu sorgen.

### Die Prognose

Bei Beachtung der genannten Aspekte kann davon ausgegangen werden, dass sich der Gesundheitszustand des Systems signifikant verbessert und die Lebenserwartung um viele Jahre steigt. Zu Risiken und Nebenwirkungen der ODI-Migration lesen Sie bitte weitere Dokumentationen und fragen Sie Ihren ETL-Spezialisten oder ein Systemhaus.

Achim Stump

[achim.stump@opitz-consulting.com](mailto:achim.stump@opitz-consulting.com)

Christian Lenzhölzer

[christian.lenzhoelzer@opitz-consulting.com](mailto:christian.lenzhoelzer@opitz-consulting.com)

Oliver Hüskes

[oliver.hueskes@opitz-consulting.com](mailto:oliver.hueskes@opitz-consulting.com)

## DOAG 2015 BI: Von Evolution und Revolution

Die Veranstaltung am 23. April 2015 in München greift die neuesten Trends zur Business Intelligence auf. Dirk Schmachtenberg führt in seiner Keynote mit eindrucksvollen Beispielen vor, wie sich die Welt immer wieder komplett verändert. In der Vergangenheit haben die Mechanisierung, der Gebrauch von Elektrizität und die Massenproduktion von Gütern ihre Spuren hinterlassen. Auch in der jetzigen Zeit hat sich Bestehendes immer weiter entwickelt und perfektioniert. Beispiele sind die Verdrängung der analogen Fotografie oder die Ablösung des traditionellen Handys.

Parallel dazu wächst die Bedeutung von Unternehmen, die Informationen, Sozialität und Vernetzung zu ihrem Geschäftsmodell machen. Player wie Google, Facebook

oder Ebay zeigen, wie man die Regeln des Markts verändern kann.

Als Hinweis, wie sich Unternehmen zukünftig positionieren und entwickeln können, stellt er zwei Modelle vor, die Evolution und die Revolution. Die Evolution umfasst eine Weiterentwicklung des Bestehenden und hat in der Regel kurzfristige Erfolgs-Chancen. Bei der Revolution hingegen wird Grundlegendes hinterfragt und verändert. Entsprechend länger lässt der Erfolg auf sich warten.

Peter Welker stellte als DOAG-Aktiver das Programm der Veranstaltung zusammen. Auch er hat sowohl evolutionäre als auch revolutionäre Vorträge berücksichtigt. Eindeutig in die Zukunft weisen Themen wie „Big Data“, „Self Service/BI“ und „In-

Memory“. Aktuell hingegen arbeiten viele Teilnehmer mit „Data Management“, „Performance-Gewinn“ und „Data-Warehouse-Architekturen“. Ein dritte Schiene ist als neuer Versuch erstmals im Programm. Es geht um die Vermittlung von Grundlagen-Themen, um Teilnehmer auf die Veranstaltung zu holen, die sich nicht unmittelbar mit der Business Intelligence beschäftigen.

Auch Michael Klose, der Leiter der neu gegründeten BI Community, hat große Ziele. Da die Arbeit ist jetzt auf viele Schultern verteilt ist, lassen sich weitaus mehr Themen angehen, mehr Infos aufbereiten und mehr Veranstaltungen organisieren. Eines seiner Ziele ist, die DOAG-Business-Intelligence-Konferenz künftig auf zwei Tage auszuweiten.