

Das Ende von OWB – was nun? Migrationspfade für OWB-Projekte

Stanislav Lando
Trivadis GmbH, Düsseldorf
Dani Schnider
Trivadis AG, Zürich/Glattbrugg

Schlüsselworte:

Data Warehouse, Oracle Warehouse Builder, Oracle Data Integrator, ETL-Tools, Migration, DWH Generatoren, PL/SQL

Einleitung

Die Ära des Oracle Warehouse Builder (OWB) neigt sich dem Ende zu. Das bisherige Oracle Standard ETL-Werkzeug war in vielen Data Warehouses das bevorzugte und beliebteste Produkt zur Entwicklung von Integrationsprozessen, weil es kostengünstig war und sehr gut in die Oracle-Datenbankwelt integriert ist. Leider kehrt die zukünftige Ausrichtung der Oracle ETL-Strategie dem Oracle Warehouse Builder den Rücken zu und lässt bei den OWB-Anwendern viele Fragen offen. Als strategisches ETL-Werkzeug wird nun Oracle Data Integrator (ODI) positioniert. Die zukünftige Ausrichtung von Oracle in Bezug auf die Zukunft von OWB und ODI ist in einem „Statement of Direction“¹ zusammengefasst:

- Die Basisfunktionalität („OWB Basic ETL“) bleibt vorerst bestehen und ist in der Datenbanklizenz von Oracle 12c Release 1 enthalten. Sie wird jedoch nicht mehr weiterentwickelt.
- Für erweiterte Funktionalitäten von OWB, die ebenfalls nicht mehr weiterentwickelt werden, muss die ODI Enterprise Edition lizenziert werden. Diese ist nicht in der Datenbanklizenz enthalten.
- Der OWB wird seit Oracle 11g Release 2 nicht mehr weiterentwickelt. Neue Datenbank-Features werden deshalb durch den OWB zukünftig gegebenenfalls nicht mehr unterstützt.
- Oracle Data Integrator ist offizieller Nachfolger vom OWB. Zusätzliche Optionen werden nur noch für den ODI entwickelt und angeboten. Der ODI ist kostenpflichtig und nicht mehr in der Datenbank-Lizenz enthalten.
- Für die Migration von OWB nach ODI steht mit ODI 12c ein Migrationswerkzeug zur Verfügung, mit dem bestehende OWB-Projekte zumindest teilweise in ODI migriert werden können.

Grundsätzlich besteht kein dringender Handlungsbedarf, die bestehenden OWB-Projekte so rasch wie möglich in eine andere ETL-Technologie zu übertragen. Solange mit bisherigen Oracle-Versionen und der vorhandenen Funktionalität weitergearbeitet werden kann, funktioniert alles wie bisher. Wer jedoch neue Datenbank-Features von Oracle 12c oder zukünftigen Versionen einsetzen möchte, sollte sich Gedanken über die Migration der bestehenden OWB-Projekte machen. Welcher Aufwand kommt bei der Migration des OWB auf Projektverantwortliche und ETL-Entwickler zu? Welches Werkzeug kann nun die ETL-Strecken langfristig und sicher abdecken? Hier werden einige mögliche Szenarien vorgestellt, um den OWB-Benutzern aufzuzeigen, was die Zeit nach OWB bringen kann.

¹ <http://www.oracle.com/us/products/middleware/data-integration/odi-statement-of-direction-1922235.pdf>

Migration zu Oracle Data Integrator (ODI)

Oracle Data Integrator ist ein flexibles ELT-Werkzeug, welches einen umfassenden und kontinuierlichen Zugriff auf heterogene Quell- und Zielumgebungen ermöglicht. Dank des Konzepts der Knowledge Module ist der Entwickler in der Lage, die Generierung des Quellcodes weitgehend frei an die vorhandenen Projektbedürfnisse und die jeweilige Datenbanktechnologie anzupassen. Mit ODI Version 12c wurden die wichtigsten Bedienungskonzepte und einige Features aus dem OWB in Oracle Data Integrator übernommen. Für OWB-Entwickler ist somit ein Umstieg auf ODI 12c relativ einfach möglich.

Wer sich für ODI entscheidet, hat einen weiteren Vorteil: Die Migration von OWB kann zu einem großen Teil mit dem OWB Migration Utility automatisch erfolgen und bei Bedarf hinausgezögert werden, da ODI ab Version 12c das Starten von bestehenden OWB-Mappings übernehmen kann. Obwohl Oracle eine Ablösung in mehreren Schritten ebenfalls anbietet, ist diese nicht zu empfehlen, da bei jeder Migration Verbindungsinformationen, Topologien und Modelle neu angelegt werden. Doch wie sieht die Migration von OWB nach ODI mit dem Migration Utility aus? Und welche OWB-Objekte können damit migriert werden?

Die Migration der OWB-Projekte mit dem OWB Migration Utility kann größtenteils automatisch erfolgen, sofern man sich bei der Entwicklung von ETL-Strecken auf die Grundfunktionalität des OWB eingeschränkt hat. Schwieriger wird es, wenn speziellere Operatoren verwendet wurden. Voraussetzung für eine erfolgreiche Migration sind die richtigen Versionen von OWB und ODI. Sind alle Voraussetzungen erfüllt und die richtigen Versionen und Patches installiert, kann ein großer Teil der bestehenden OWB-Projekte migriert werden. Migriert werden können Module (Source und Target), Locations, Datenstrukturen (Tabellen, Views, Materialized Views, External Tables, Flat Files und Sequences) sowie Mappings mit einer großen Palette von Operatoren.

Verschiedene Datenobjekte und Operatoren, die zur erweiterten (kostenpflichtigen) Version des OWB gehören, können nicht automatisch migriert werden. In diesen Fällen ist auf jeden Fall eine manuelle Nachbearbeitung oder unter Umständen sogar eine vollständige Neuimplementierung mit ODI notwendig. Eine vollautomatische Migration von OWB nach ODI kann es nur in seltensten Fällen geben. Zusätzlicher manueller Entwicklungsaufwand wird in den meisten Fällen notwendig sein. Dies trifft vor allem dann zu, wenn im OWB Datenobjekte oder Operatoren verwendet wurden, die vom OWB Migration Utility nicht unterstützt werden, beispielsweise Dimensions und Cubes, Data Rules oder Data Profiles.

Eine weitere konzeptuelle Herausforderung entsteht beim Umstieg von im OWB generierten PL/SQL auf natives SQL im ODI. Somit müssen wesentliche Aspekte des DWH, wie Skalierbarkeit (Partitionen) und CDC-Logik geändert bzw. neu konzipiert werden. Da ODI eine viel größere Auswahl der Funktionalitäten bei den Prozessflüssen (ODI Szenarien) anbietet, können OWB Process Flows weitgehend auf ODI Szenarien übertragen bzw. mit ODI neuentwickelt und bei Bedarf optimiert werden.

Oracle Data Integrator gehört zur Fusion Middleware und muss getrennt von der Oracle Datenbank lizenziert werden. Für Kunden, die bisher mit der OWB Enterprise Edition (bzw. „Enterprise ETL Option“) gearbeitet haben, ändert sich nichts. Sie werden als Neuerung die ODI Enterprise Edition verwenden. Für OWB-Kunden, die sich auf die Basis-ETL-Funktionalität beschränkt und OWB im Rahmen der Datenbanklizenz betrieben haben, fallen durch die Migration auf Oracle Data Integrator zusätzliche Lizenzkosten an.

OWB-generierte PL/SQL Packages weiterverwenden

In vielen Data Warehouses werden die ETL-Prozesse mit PL/SQL entwickelt oder generiert. Auch der generierte Quellcode von OWB besteht aus PL/SQL, indem OWB pro Mapping ein PL/SQL-Package generiert. Ein naheliegender Ansatz besteht deshalb darin, den generierten PL/SQL-Code weiterhin zu verwenden, jedoch ohne Oracle Warehouse Builder.

Die generierten PL/SQL-Packages von OWB verwenden zahlreiche Objekte aus dem Runtime-Repository (OWBSYS). Diese werden für Fehlerbehandlung, Audit-Informationen, Logging und Protokollierung verwendet. Somit ist der PL/SQL-Code mit dem OWB-Repository-Schema stark verflochten. Bleibt das Schema OWBSYS bestehen, bleiben auch die vom OWB generierten PL/SQL-Packages weiterhin lauffähig. Wählt man dieses Szenario, ist es sicherlich nachteilig, die Packages ohne grafische OWB-Oberfläche zu warten und weiter zu entwickeln. Für die Ausführung der bestehenden ETL-Strecken ist diese Variante jedoch durchaus geeignet, solange sie auf den bisherigen Versionen der Oracle-Datenbanken betrieben werden. Wird jedoch das Data Warehouse auf eine neue Oracle-Version migriert, muss auch das OWB-Runtime-Repository migriert werden. Für Oracle 12c Release 1 ist dies mit entsprechenden OWB-Patches noch möglich. Mit Blick auf die Zukunft stellt sich jedoch die Frage, ob vom Oracle Warehouse Builder generierter PL/SQL-Quellcode auf den neuen Datenbankversionen weiterhin lauffähig bleibt.

Ein Ansatz, der hier Abhilfe schaffen kann, besteht darin, die im OWB-Code verwendeten OWBSYS-Objekte durch eine eigene Implementation zu ersetzen. Sind die Namen und Parameter der aufgerufenen Prozeduren und Funktionen gleich, so kann mit dieser Variante das Runtime-Repository von OWB simuliert werden. Damit kann sichergestellt werden, dass die generierten PL/SQL-Packages weiterhin lauffähig sind, auch wenn kein OWB-Repository mehr vorhanden ist. Bei der eigenen Implementation der Funktionen und Prozeduren des Runtime-Schemas muss darauf geachtet werden, dass diese weiterhin effizient ausgeführt werden, um die Laufzeit der ETL-Strecken nicht zu beeinträchtigen. Als Übergangslösung kann dieser Ansatz zweckmäßig sein, aber eine längerfristige Weiterentwicklung auf Basis der von OWB generierten PL/SQL-Packages ist nicht empfehlenswert.

Somit stellt sich die Frage: Wie können die wesentlichen SQL- und PL/SQL-Statements aus den OWB-Mappings extrahiert werden, ohne dass die zusätzlichen Aufrufe des OWB Runtime-Repositories übernommen werden müssen? Die zentrale Logik eines Mappings lässt sich im OWB einfach generieren, indem die „Intermediate Results“ angezeigt und kopiert werden. Für die Generierungsoption „set-based“ werden dabei (große) INSERT- oder MERGE-Statements generiert. Diese lassen sich ohne großen Aufwand in ein eigenes PL/SQL-Package integrieren.

Eine weitere Möglichkeit der automatischen Extraktion und Überführung der ETL Business Logik besteht darin, einen Codeanalyzer² zu verwenden, welcher den generierten PL/SQL-Code analysiert und die relevanten Teile daraus extrahiert. Die Verwendung eines solchen PL/SQL-Parsers ist zwar möglich, jedoch ebenfalls mit größerem Aufwand verbunden.

Die auf diese Art extrahierten Codeteile können wie bei der Extraktion mittels Intermediate Results in eigene, neue PL/SQL-Packages übernommen werden. Der Vorteil eines Codeanalyzers besteht darin, dass das Extrahieren und Kopieren nicht manuell für jedes Mapping ausgeführt werden muss, sondern mittels Scripts automatisiert werden kann. Für sehr große OWB-Projekte mit vielen Mappings kann sich dieser Aufwand durchaus lohnen.

² z.B. Trivadis PL/SQL Analyzer, <http://www.salvis.com/blog/downloads/tvdca-trivadis-plsql-sql-codeanalyzer/>

Einsatz von Data Warehouse Automation Tools

Mit Data Warehouse Automation Tools können Datenstrukturen und Datenflüsse von DWH-Lösungen automatisiert generiert werden. Dies kann für OWB-Projekte mit hohem Standardisierungsgrad ein wesentlicher Vorteil sein, da nicht jedes einzelne Mapping migriert werden muss, sondern die ETL-Prozesse neu generiert werden können. Dieses Verfahren soll hier anhand von biGenius, dem von Trivadis entwickelten Data Warehouse Automation Tool, erläutert werden. Mit dem PL/SQL Generator von biGenius besteht die Möglichkeit, die Tabellen und Views der verschiedenen DWH-Schichten sowie die ETL-Prozesse in Form von PL/SQL-Packages zu generieren.

Die Datenstrukturen und ETL-Prozesse in einem standardisierten Data Warehouse sind immer ähnlich aufgebaut. Je nach verwendeter DWH-Architektur werden die Quelldaten über mehrere Zwischenstufen in die erforderlichen Zielstrukturen (Core Data Warehouse und Data Marts) geladen. Die Funktionalität der dafür benötigten ETL-Prozesse ist oft gleich oder sehr ähnlich aufgebaut. Unabhängig von Inhalt und Struktur der verwendeten Tabellen sind viele Abläufe für jede Quell- und Zieltabelle gleich. Dazu gehören beispielsweise Datenbereinigung, Deltaermittlung, Zuweisung von künstlichen Schlüsseln oder der Aufbau von Hierarchien in denormalisierten Dimensionen.

Ein Data Warehouse, das eine einheitliche Architektur verwendet, kann relativ einfach in biGenius übernommen und dort weiterentwickelt werden. Für die Trivadis DWH-Architektur stehen bereits vordefinierte Templates zur Verfügung. Für kundenspezifische DWH-Architekturen können diese angepasst oder durch eigene Templates ergänzt werden. Für ETL-Prozesse mit viel Speziallogik und Sonderfällen können individuelle Templates verwendet werden. Dies sollte aber nur in Ausnahmefällen gemacht werden, da sonst die Vorteile eines Automation Tools nicht ausgeschöpft werden können.

Im Gegensatz zu Oracle Warehouse Builder (oder anderen Entwicklungswerkzeugen) werden in biGenius nicht einzelne Tabellen und ETL-Prozesse definiert, sondern übergeordnete logische Entitäten sowie zugeordnete Source-Objekte. Für diese generiert dann biGenius die erforderlichen Datenbankobjekte. So werden beispielsweise pro Source-Objekt eine Stage-Tabelle und pro Entität eine Cleanse-Tabelle, eine Head- und Versionstabelle im Core sowie eine Dimensionstabelle im Data Mart generiert. Der mitgelieferte PL/SQL Generator generiert zusätzliche Views für Lookups, Deltaermittlung und historisierte Abfragen auf das Core. Die Views und Tabellen werden in den ebenfalls generierten PL/SQL Packages verwendet, die zum Laden der Daten in die unterschiedlichen DWH-Schichten verwendet werden.

Die Generierung der Tabellen, Views und Packages erfolgt anhand von Metadaten, die in der biGenius-Applikation eingegeben (oder importiert) werden. Auch dazu ein Beispiel: Für jedes Attribut einer Entität wird neben Attributname und Datentyp auch das Historisierungsverhalten (Slowly Changing Dimension Typ 1 oder 2) festgelegt sowie die zugehörigen Source-Attribute zugeordnet. Spezielle Berechnungsregeln können ebenfalls definiert werden. Um standardisierte Transformationen wie Typenumwandlungen, Einfügen von Singleton-Werten oder Surrogate Keys muss sich der Entwickler jedoch nicht kümmern. Diese Aufgaben werden von den mitgelieferten Templates für den jeweiligen Generator übernommen.

Der Vorteil eines Data Warehouse Automation Tools kommt vor allem dann zum Tragen, wenn ein standardisiertes Data Warehouse mit vielen Objekten migriert werden soll. Da nur die logischen Strukturen, nicht aber die ETL-Prozesse migriert werden müssen, ist die Migration weniger aufwendig, als wenn jedes einzelne OWB-Mapping migriert werden müsste.

Migration zu kommerziellen ETL-Tools

In zahlreichen Oracle Data Warehouses werden ETL-Tools von Fremdherstellern eingesetzt, um die Daten zu transformieren und in die Oracle-Datenbank zu laden. Zwar sind diese Tools in der Regel weniger auf Oracle-Datenbanken abgestimmt als Oracle Warehouse Builder oder Oracle Data Integrator, aber sie erlauben ebenfalls eine komfortable und effiziente Entwicklung von Ladeprozessen. Hier soll anhand eines häufig eingesetzten ETL-Tools - Informatica PowerCenter – aufgezeigt werden, wie ein bestehendes OWB-Projekt in ein Integrationswerkzeug eines anderen Herstellers migriert werden kann.

Informatica PowerCenter gehört zu den weitverbreitetsten und umfangreichsten ETL-Tools auf dem Markt. Im Gegensatz zu OWB und ODI werden in diesem Werkzeug die Transformationsprozesse nicht in der Datenbank ausgeführt, sondern auf einem separaten ETL-Server. Die ETL-Prozesse werden von Integration Services ausgeführt, indem die Quelldaten zunächst im eigenen Speicher geladen und dort transformiert werden (Blackbox-Prinzip), um diese in eine oder mehrere Zieltabellen zu laden. PowerCenter ist also ein ETL-, kein ELT-Tool.

Damit PowerCenter auf die Datenstrukturen der Quellsysteme und der vorhandenen DWH-Schichten zugreifen kann, müssen diese Strukturinformationen ins PowerCenter-Repository eingelesen werden. PowerCenter Designer kann problemlos die Metadaten von Tabellen, Views, Materialized Views und Flat Files importieren. Für die meisten OWB-Operatoren (Joiner, Filter, Key Lookup, etc.) stehen in PowerCenter vergleichbare Transformationen zur Verfügung. Da sie sich jedoch zum Teil in der Funktionsweise unterscheiden und auch eine andere Syntax für Expressions und Funktionsaufrufe verwendet wird, ist eine automatische Migration von einem OWB-Mapping in ein PowerCenter-Mapping nicht möglich. Eine manuelle Migration der bestehenden Logik ist zwar in den meisten Fällen möglich, jedoch sehr aufwendig. Sie kommt einer Neuentwicklung der bestehenden ETL-Prozesse gleich.

Um diesen großen manuellen Aufwand zu vermeiden, wird hier eine andere Alternative vorgeschlagen. Sie kann nur für OWB-Mappings verwendet werden, die set-based ausgeführt werden. Die bestehende Transformationslogik kann dabei mit einem Trick in Informatica übernommen werden. Der von Oracle Warehouse Builder generierte SQL-Code wird mittels „Intermediate Results“ extrahiert, wie bereits beschrieben. Es wird nur der „Incoming“-Teil ausgewählt, welcher das SELECT-Statement für die Datenmenge der Zieltabelle enthält. Dieses SELECT wird als „SQL Override“ in den Source Qualifier des PowerCenter-Mappings kopiert. Aus Sicht von Informatica wird nun auf eine „Quelltabelle“ zugegriffen, welche das Resultat des SQL-Statements enthält. Die gelesenen Datensätze können dann zeilenweise in die Zieltabelle geschrieben werden (INSERT oder UPDATE).

Kurzfristig gesehen bietet die SQL-Override Migrationmethode eine schnelle Lösung der Migration an. Langfristig aber, spätestens bei der Weiterentwicklung der Mappings, wird sich diese Vorgehensweise nicht bewähren. Dass die gesamte Transformationslogik im Source Qualifier „versteckt“ ist und die Funktionalität von PowerCenter kaum verwendet wird, erschwert die Weiterentwicklung der Mappings durch ETL-Entwickler mit Informatica-Knowhow.

Die Lizenzkosten von kommerziellen ETL-Tools sind oft ein wesentlicher Kostenfaktor bei Entwicklung und Betrieb eines Data Warehouses. Je nach Hersteller und Lizenzmodell werden dabei unterschiedliche Grundlagen verwendet, was einen Preisvergleich der verschiedenen Werkzeuge erschwert. Bei Informatica PowerCenter beispielsweise werden die Lizenzen je Domain und CPU Core berechnet.

Migration zu Open Source ETL-Tools

Viele OWB-Kunden schrecken vor den zusätzlichen Lizenzkosten zurück, die bei Oracle Data Integrator, aber auch den meisten der ETL-Tools von Fremdherstellern anfallen. Deshalb liegt es nahe, stattdessen Open-Source-Produkte einzusetzen. Es gibt einige ETL-Tools in diesem Bereich. Eines der am weitesten verbreiteten Werkzeuge ist Talend Open Studio, das für die folgenden Erläuterungen als Beispiel verwendet wird.

Talend Open Studio ist ein leistungsfähiges Integrationswerkzeug mit einer offenen, erweiterbaren Architektur, basierend auf Java. Die Entwicklung von ETL-Jobs erfolgt im Talend Job Designer. Die grafische Darstellung der Jobs ist für OWB-Entwickler am Anfang etwas ungewohnt, aber klar strukturiert. Die Funktionalität ist sehr umfangreich, es stehen über 500 sogenannte „Components“ – vergleichbar mit den Operatoren in OWB – zur Verfügung. Die Ausführung erfolgt in der Regel row-based. Zwar bietet Talend auch ELT-Komponenten an, welche eine mengenbasierte Ausführung in der Zieldatenbank ermöglichen. Diese sind jedoch relativ umständlich in der Anwendung. Die Frage, die in diesem Dokument im Zentrum steht, lautet: Wie lassen sich bestehende OWB-Mappings in Talend Open Studio migrieren?

Wie bereits bei Informatica PowerCenter beschrieben, benötigt auch Talend Open Studio die Datenstrukturen der Quell- und Zielsysteme. Diese lassen sich genauso einfach importieren wie bei den kommerziellen Tools. Auch bei der ETL-Entwicklung finden wir eine ähnliche Situation vor wie bei Informatica. Zwar bietet Talend eine große Auswahl an Komponenten mit unterschiedlicher Funktionalität, da sich diese jedoch stark von der Arbeitsweise der Operatoren in OWB unterscheiden, ist eine automatisierte Migration von OWB-Mappings in Talend-Jobs nicht möglich. Stattdessen muss für jedes Mapping ein entsprechender Talend Job implementiert werden.

Um diesen manuellen Entwicklungsaufwand zu reduzieren, gibt es auch hier die Möglichkeit, statt den OWB-Mappings die daraus generierten SQL-Statements in Talend zu übernehmen. Die Talend-Komponente „tOracleRow“ eignet sich – obwohl der Name nicht danach klingt – für die Migration von set-based OWB-Mappings. Mit dieser Komponente kann ein beliebiges SQL-Statement definiert und in der Zieldatenbank ausgeführt werden. Somit kann mittels „Intermediate Results“ das SQL-Statement des OWB-Mappings (INSERT-, UPDATE oder MERGE-Statement) extrahiert und in Talend übernommen werden. Unter Umständen muss das Statement noch angepasst werden, beispielsweise indem Schemanamen ergänzt oder Platzhalter für Variablen (in Talend „Contexts“ genannt) eingefügt werden.

Ähnlich wie beim SQL Override in Informatica besteht auch hier die Problematik, dass die Transformationslogik in einem SQL-Befehl „versteckt“ ist. Die große Vielfalt an Funktionalität von Talend kommt somit nicht zum Zug – es wird im Wesentlichen nur die Komponente zum Ausführen von SQL-Befehlen verwendet. Dafür hat diese Vorgehensweise einen nützlichen Nebeneffekt: Talend-Jobs, welche SQL-Statements mittels der Komponente tOracleRow ausführen, sind bei großen Datenmengen viel schneller als Talend-Jobs, welche die gleiche Funktionalität mit den entsprechenden ETL-Komponenten von Talend ausführen.

Open Source heißt nicht kostenlos. Auch ein ETL-Server eines Open-Source-Produkts benötigt Hardware und Support. Außerdem bieten die meisten Anbieter von Open-Source-Software auch lizenzpflichtige Versionen ihrer Produkte an. So wird bei Talend die Enterprise Data Integration Suite benötigt, wenn mehrere ETL-Entwickler auf einem gemeinsamen Repository arbeiten. Die Lizenzierung dieser kostenpflichtigen Version erfolgt nicht, wie bei anderen Herstellern üblich, nach Anzahl CPUs oder Zielumgebungen, sondern nach Anzahl der Entwickler.

Zusammenfassung

Welcher der hier vorgestellten Migrationspfade ist nun der Beste? Leider gibt es darauf keine eindeutige Antwort. Welche Variante am geeignetsten ist, hängt von verschiedenen Faktoren ab. Dazu gehören beispielsweise Projektgröße, Komplexität der ETL-Mappings, Datenmenge sowie Wartbarkeit. Aber auch die vorhandene Systemlandschaft und das Knowhow des Entwicklungsteams können für die Wahl der besten Lösung ausschlaggebend sein. Außerdem spielt es eine Rolle, ob es sich beim bestehenden Data Warehouse um eine reine Oracle-Lösung handelt oder ob das System in eine heterogene Systemlandschaft eingebettet ist.

Als Entscheidungshilfe für die Migration von OWB-Projekten hat Trivadis ein White Paper³ erarbeitet. Es enthält neben zusätzlichen Hinweisen zu den hier vorgestellten Migrationspfaden auch eine Vergleichsmatrix, in welcher die unterschiedlichen Szenarien anhand verschiedener Kriterien gewichtet und bewertet werden.

Kontaktadresse:

Stanislav Lando
Trivadis GmbH
Central Park Offices
Werdener Strasse 4
40227 Düsseldorf

Dani Schnider
Trivadis AG
Europa-Strasse 5
CH-8152 Glattbrugg

Telefon: +49 211 58 66 64 70
Fax: +49 211 58 66 64 71

+41 44 808 70 20
+41 44 808 70 21

E-Mail stanislav.lando@trivadis.com
Internet: www.trivadis.com

dani.schnider@trivadis.com

³ <http://www.trivadis.com/sites/default/files/downloads/migrationspfade-fuer-owb-projekte.pdf>