Hauptüberschrift

Michael Klose
CGI (Germany) GmbH & Co. KG
Sulzbach (Taunus)

Schlüsselworte
OWB, ODI, ETL, Toolauswahl

Einleitung

Oracle Warehouse Builder wird von Oracle nicht weiterentwickelt und der Standard Support endet bereits in naher Zukunft. Gerade Kunden, welche keine 12cR1 installieren wollen sind gezwungen sich nach Alternativen umzusehen. Oracle stellt mit dem Data Integrator einen würdigen Nachfolger bereit, welcher allerdings mit zusätzlichen Lizenzkosten verbunden ist.

Der Vortrag zeigt verschiedene Möglichkeiten des Umstiegs auf ein anderes ETL-Tool auf. Diese werden anhand von Praxiserfahrungen aus einer ETL-Toolauswahl dargestellt. Weiterhin werden die Vor- und Nachteile der verschiedenen Tools, sowie grundsätzliche Veränderungen in der Entwicklungstätigkeit und Architektur aufgezeigt.

Die Schwerpunkte liegen hier im Bereich Oracle Data Integrator, Informatica Powercenter, Talend und PL/SQL.

Grundsätzliches
Im ETL Tool Bereich gibt es durch die große Anzahl von Entwicklungswerkzeugen, gerade aus Architektursicht, unterschiedliche Ansätze der Hersteller. Grundlegend unterscheidet man klassisch zwischen ETL (Extract-Transform-Load) und ELT (Extract-Load-Transform). Beim ETL-Prozess werden die Daten aus den Quellsystemen beispielsweise als Flat Files extrahiert, im Anschluss in einer ETL-Engine transformiert, um letztendlich final in die Zieldatenbank (Data Warehouse) geladen zu werden. Im Gegensatz dazu werden bei ELT basierten Werkzeugen die Daten zuerst in die Zieldatenbank geladen, in welcher dann in der Folge die Transformationen durchgeführt werden. Ein weiterer großer Unterschied liegt in der Programmiersprache des generierten Codes der Transformationsprozesse.

ETL Tools im Überblick
Im Vortrag werden folgende Varianten näher betrachtet:

Oracle Data Integrator:
Der Oracle Data Integrator (ODI) wird von Oracle als Nachfolger des Warehouse Builder (OWB) positioniert. In der aktuellen Version wird ein Migrationstool mitgeliefert, welches es ermöglicht je nach Komplexität der Mappings 70-80% automatisiert zu migrieren.
Der ODI gehört zu der Kategorie der ELT Werkzeuge und verwendet die Datenbank als ETL-Engine. Häufig wird von ODI-Neueinsteigern der ODI-Agent als ETL-Engine bezeichnet. Beim ODI-Agent handelt es sich um einen Java Prozess, welcher für die Ausführung der generierten Statements auf den Datenbanken zuständig ist, selbst aber keine Transformationen durchführt.

Eine weitere Kernkomponente sind die Knowledge Module, die für die verschiedenen Datenbanken entsprechende SQL Statements generieren, welche die Transformations- und Ladeprozesse abbilden. Eine eigene ETL-Engine besitzt der ODI hingegen nicht und kann somit „nur“ die Funktionsbibliotheken der verwendeten Datenbanken nutzen. Knowledge Module können vom Benutzer auf die entsprechenden Anforderungen angepasst werden. Durch die Generierung von SQL Statements bleibt der „Code“ lesbar und kann beispielsweise für Tuning Zwecke gut nachvollzogen werden.
Weitere wichtige Bestandteile sind Workflow und Load Plans. Über diese Komponenten wird die Lade- und Ablaufsteuerung realisiert. Im Vergleich zum OWB sind diese wesentlich mächtiger. Dies zeigt sich vor allem im Bereich der Wiederaufsetzbarkeit von Ladeprozessen nach Abbrüchen.
Auch in der Versionierung und Paketierung für das Deployment gibt es nennenswerte Verbesserungen. Im ODI besteht die Möglichkeit jede einzelne Komponente zu versionieren und in Deployment Paketen zusammenzufassen. Der Funktionsumfang von Versionierungswerkzeugen wie Subversion wird zwar nicht erreicht, allerdings ist ein sinnvolles Arbeiten mit Versionierung möglich.
Im Gegensatz zum OWB ist eine Vielzahl von Datenbanken als Quell- und Zielsystem anbindbar. Die Verbindung zur Datenbank wird über den ODI Agent und entsprechenden JDBC Treibern hergestellt. Für nahezu alle namhaften Datenbankhersteller werden umfangreiche Best-Practice Knowledge Module mitgeliefert. Dies hat zur Folge, dass der ODI auch in sehr heterogenen Systemlandschaften im Gegensatz zum OWB sehr gut eingesetzt werden kann.

Informatica Powercenter:
Informatica Powercenter (IPC) gehört zur Kategorie der ETL-Tools und ist sicher eines der mächtigsten und umfangreichsten Werkzeuge im ETL-Bereich, was Analysten regelmäßig bestätigen. Der Einsatz einer eigenen ETL-Engine führt zu einer Unabhängigkeit gegenüber der Funktionsbibliothek der Datenbank. Allerdings benötigt diese Engine einen (im Normalfall) eigenständigen Server zur Ausführung der Transformationsprozesse. Dadurch ist es möglich, Daten aus verschiedenen Quellen zu extrahieren, in die ETL-Engine zu laden, Transformationen, Joins und Aggregationen durchzuführen und die Daten final in die Zieldatenbank zu schreiben. In der klassischen Datawarehouse Layer Architektur (Stage/EDWH/Data Marts) ist es in diesem Fall allerdings auch notwendig, die Daten beim Transport durch die verschiedenen Layer aus dem Warehouse zu extrahieren und nach den Transformationen wieder in dieses zu laden (z.B. EDWH nach Data Mart). Informatica bietet für Powercenter zwar eine sogenannte Push-Down Funktion an, welche versucht möglichst viele Tätigkeiten auf die Datenbank auszulagern, allerdings kann nicht alles in der Datenbank ausgeführt werden. Diese Funktionalität war in der Vergangenheit mit zusätzlichen Kosten verbunden und wurde deswegen vom Kunden häufig nicht eingesetzt. Um beispielsweise bei Delta-Verarbeitung die Performance beim Laden zwischen den DWH Layern zu verbessern, werden häufig die Source Qualifier (Informatica Objekt welches das SELECT-Statement enthält) manuell mit Joins und Filtern angepasst. Dies führt zu einem Verlust der Data-Lineage Funktionaliät in diesem Bereich, ist aber häufig die einzige Optimierungsmöglichkeit.
IPC generiert keinen „lesbaren“ Code, welcher optimiert und visualisiert werden kann und stellt somit eine „Black Box“ dar. Die Oberfläche selbst erinnert stark an den OWB und aus der Erfahrung zeigt sich, dass Mitarbeiter mit OWB Kenntnissen sehr schnell beginnen können Mappings in IPC zu entwickeln.
Zur Ausführung und Prozesssteuerung ist eine Workflow Komponente verfügbar. Wird kein externes Workflow/Scheduling Tool wie CTRL-M eingesetzt, ist die Vorgehensweise, gerade beim verschachteln von Workflows, eher schlecht gelöst.
Durch die kostenpflichtige Option „Team Based Development“ wird die Versionierung, sowie ein Check-In/Out ähnlich Subversion ermöglicht. IPC besitzt viele (lizenzpflichtige) Konnektoren zu

Datenbanken, welche bei den großen Datenbankherstellern sogar ein Change Data Capture für den Datenabzug unterstützen.

Talend Enterprise Data Integration

Talend bietet, abhängig von den Anforderungen, verschiedene Ausbaustufen seiner Data Integration Suite an. Um das Leistungsspektrum des Oracle Warehouse Builders abzubilden, ist hier die Enterprise Data Integration Variante zu betrachten. Talend gehört ebenso wie Informatica zur Kategorie der ETL-Tools. Als ETL-Engine dient hier letztendlich die JAVA Runtime Engine. Mappings werden wie bei den vorherigen Tools grafisch entwickelt und ausführbarer lesbarer Java-Code generiert. Die Anbindung von Quell- und Zielsystemen erfolgt über JDBC-Treiber, welche von fast allen Datenbankherstellern angeboten werden. Dadurch eignet sich Talend für sehr heterogene Quell- und Zielsysteme. Die Daten müssen zur Durchführung von Transformationen aus dem Datawarehouse extrahiert und in der JAVA Runtime Engine verarbeitet werden. Hierfür ist ein zusätzlicher ETL-Server empfehlenswert, der vor allem im Bereich CPU und Memory nicht zu klein ausgestattet sein sollte. Der Push-Down von Datenbank Abfragen ist möglich, allerdings wird dies nicht für alle Datenbanken angeboten. Die mitgelieferte Funktionsbibliothek, welche Out-of-the-Box in einer grafischen Entwicklung verwendet werden kann, ist nicht sehr umfangreich. Dies hat zur Folge, dass die Transformationen in JAVA programmiert werden müssen. Hier steht wiederum der komplette Funktionsumfang von JAVA zur Verfügung, so dass auch weitere Libraries eingebunden werden können. Da es sich um reinen JAVA Code handelt, bettet sich Talend optimal in Versionierungssoftware wie Subversion ein.

Oracle PL/SQL

Sicherlich nimmt Oracle PL/SQL für den ETL-Prozess bei einer Tool-Betrachtung eine Nebenrolle ein, nichtsdestotrotz haben viele Unternehmen mit Oracle Datenbanken diese Variante erfolgreich im Einsatz. Im Vergleich zu ETL-Tools wie Informatica oder Talend ist PL/SQL und SQL wesentlich näher am Ergebnis der Mapping Generierung des OWB. Die großen Vorteile der Tools liegen in den Bereichen grafische Entwicklung, Metadaten Repository, automatische Protokollierung und Data Lineage. Auf den ersten Blick stellen diese nicht zu vernachlässigende Komponenten dar, allerdings können einige auf andere Weise in einem Oracle DB und PL/SQL Umfeld alternativ abgebildet werden.

Die grafische Entwicklungsoberfläche ist sicherlich nicht durch PL/SQL zu ersetzen. Eine Variante um SQL Statements zu generieren stellt beispielsweise der Query Builder dar. Allerdings werden nur wenige Entwickler diese Alternative einsetzen wollen.

Beim Metadaten Repository und der Data Lineage ist die Diskrepanz lange nicht so groß. Letztendlich stellt das Data Dictionary der Datenbank eine Art „Metadaten Repository“ dar. Auf Datenbankobjektebene können hier viele Informationen hinterlegt werden. Um die Transformations-Metadaten zu hinterlegen, bietet es sich an ein eigenes „Mini-Repository“ anzulegen. In der Oracle DWH Community wird ein solches kostenfrei zur Verfügung gestellt. Setzt man dieses ein, kann eine Data Lineage Analyse auf einfachem Weg aufgesetzt werden. Anderenfalls bietet die Datenbank selbst Möglichkeiten die Abhängigkeiten zwischen verschiedenen Objekten auszuwerten.

Ein wichtiger Bestandteil jedes ETL-Prozesses ist die Protokollierung der Laufzeitinformationen. In diesem Bereich existieren fertige Logging Frameworks, welche teilweise kostenfrei eingesetzt werden können. Alternativ dazu können eigene Frameworks in PL/SQL entwickelt und auf einfachem Wege eingebunden werden.

Eine weitere wichtige Komponente stellt die Ablaufsteuerung dar. Gerade was Parallelisierung von Ladeprozessen angeht sind die Möglichkeiten von PL/SQL doch sehr begrenzt. Für diesen Zweck kann der Oracle Enterprise Manager verwendet werden. Dort können sowohl Datenbank als auch

Betriebssystem Jobs definiert werden. Eine Parallelisierung der Prozesse und Abbildung von Abhängigkeiten unter den Prozessen ist möglich.
Viele OWB Entwickler verfügen über PL/SQL Kenntnisse und somit wäre ein einfacher Umstieg was die Entwicklung betrifft möglich. Bei bestehenden OWB Implementierungen könnten die „Insert-As-Select“ Statements über OMB+ als Script generiert werden und somit die Basis für die Migration darstellen. Selbst Row Based Mappings können überführt werden. Voraussetzung dafür ist allerdings, dass die OWB Komponenten (Logging, Initialisierung,...) vorher entfernt werden.

Fazit
Im Vortrag werden detailliert und anhand von Beispielen die verschiedenen Möglichkeiten des Umstiegs auf ein anderes ETL-Tool näher betrachtet. Jedes Tool hat seine Vor- und Nachteile. Aus diesem Grund ist es umso wichtiger die Anforderungen, individuell auf das eigene Umfeld betrachtet, festzulegen und in einem Tool-Auswahlprozess zu bewerten. Erfahrungen aus einem solchen Prozess, welcher über Long- und Shortlist sowie intensive Proof of Technology/Concept Workshops zu einer Tool Entscheidung führte, werden im Vortrag näher dargestellt.

Kontaktadresse:
Michael Klose
CGI (Germany) GmbH & Co. KG
Am Limespark 2
D-65843 Sulzbach (Taunus)

Telefon: +49 (0) 171-977 90 99
E-Mail Michael.Klose@cgi.com
Internet: www.de.cgi.com