Automatische Generierung der ETL-Prozesse: OWB vs. ODI

Irina Gotlibovych
MT AG
Ratingen

Schlüsselworte

OWB, ODI, Prozessgenerierung, Automatisierung, generische Templates, deklarative Definition

Einleitung

Wie oft in unserem Leben wünschen wir uns, bestimmte wiederkehrende Routineaufgaben nicht mehr von Hand oder am liebsten gar nicht mehr selbst erledigen zu müssen? Warum soll es bei der Entwicklung von ETL-Prozessen in einem Data Warehouse anders sein? Anhand eines von uns entwickelten Frameworks werden im Vortrag Möglichkeiten der automatischen Generierung der ETL-Prozesse im Oracle Warehouse Builder einerseits und Oracle Data Integrator andererseits gegenübergestellt und verglichen.

ETL Entwicklung in manueller Kleinarbeit

Bei der Entwicklung der ETL-Prozesse in einem Data Warehouse sieht man sich wiederholt vor die Aufgabe gestellt, Prozessschritte aufbauen zu müssen, die einer gleichartigen Logik folgen. So werden in jedem Projekt viele Datenobjekte auf die gleiche Weise aus Quellsystemen in den Arbeitsbereich geladen. Bei dem Transformationsschritt werden Daten in das einheitliche Format der Zieldatenbank überführt; gängige Verfahren dabei sind z.B. Datentypkonvertierung und Datenbereinigung. Anschließend werden Daten nach dem gleichen Prinzip – wie etwa Delta Load oder SCD – in das Data Warehouse eingebracht.

Arbeitet man mit dem Oracle Warehouse Builder, bedeutet dies in der Praxis oft, dass logisch identische Mappings in manueller Kleinarbeit angelegt werden. In jedem dieser Mappings sind von Hand Operatoren anzulegen und zu verbinden. Für jedes Attribut eines Expression Operators muss manuell der Ausdruck eingetragen werden. Eigenschaften von Operatoren und Attributen sind immer wieder neu zu setzen. Bei einer späteren Änderungsanforderung muss jedes einzelne Mapping wieder angepasst und getestet werden. Dieses Vorgehen erfordert einen hohen Entwicklungsaufwand bzw. Aufwand bei späteren Änderungen, ist fehleranfällig und testintensiv.

Ähnlich verhält es sich bei der Entwicklung mit dem Oracle Data Integrator. Interfaces müssen für jede Zieltabelle einzeln angelegt werden. Trotz der gleichen Logik sind in jedem Interface Zuordnungen von Attributen unabhängig voneinander zu definieren und bei späteren Anforderungen auch einzeln zu ändern. Der Oracle Data Integrator beinhaltet durch die mitgelieferten Knowledge-Module bereits die Möglichkeit, generische Funktionalitäten zu nutzen bzw. aufzubauen. Ähnlich wie Makros oder Templates beinhalten Knowledge-Module wiederverwendbare Logiken und nehmen dadurch dem Entwickler einen Teil der manuellen Arbeit ab. Sie können in mehreren Interfaces eingebunden werden, ersetzen aber nicht die manuelle Anlage jedes einzelnen von ihnen.

Wie hätten wir es gerne?

Der Nachteil sowohl bei der Entwicklung im OWB als auch im ODI besteht darin, dass es keine Möglichkeit gibt, Mappings bzw. Interfaces ohne Bindung an konkrete Objekte (Tabellen, Spalten etc.) anzulegen. Die fachliche Logik eines Prozesses ist immer fest mit den Umgebungsinformationen verbunden. Um die gleiche Logik nicht mehrfach neu erzeugen zu müssen, wäre ein Weg erforderlich, Prozesse generisch, also ohne Bezug zu den eigentlichen Objekten definieren zu können. Die Erzeugung der Mappings bzw. Interfaces mit der gleichen Logik kann dann für eine Vielzahl unterschiedlicher Objekte automatisch erfolgen, wobei Objektnamen als Parameter dem Generierungsprozess mitgegeben werden.

Dieser generische Ansatz wurde bei der Entwicklung des ETL Generators zu Grunde gelegt.

ETL Generator

Das Framework besteht aus zwei Komponenten – OWB Mapping Generator und ODI Interface Generator - und ermöglicht es Ihnen, Prozesse auf Basis von mitgelieferten oder selbstentwickelten Templates automatisch zu generieren. Diese Templates bilden die fachliche Logik Ihrer ETL-Prozesse ab und werden in der Datenbank deklarativ anhand von Metadaten definiert. Die Definition der Templates erfolgt generisch, d.h. sie legen die fachliche Logik der Prozesse fest, sind jedoch unabhängig von konkreten Objekten.

Im Gegensatz zur manuellen Entwicklung setzt man beim Gebrauch des Frameworks nicht mehr jedes Mapping im OWB Design Center bzw. jedes Interface im ODI Designer einzeln um, sondern definiert ein allgemeingültiges Template für eine „Klasse“ von Prozessen (siehe Abbildung 1 für OWB Mapping Generator). Anschließend generiert man die dazugehörigen Prozesse unter Einbeziehung der Projektvorgaben automatisch.

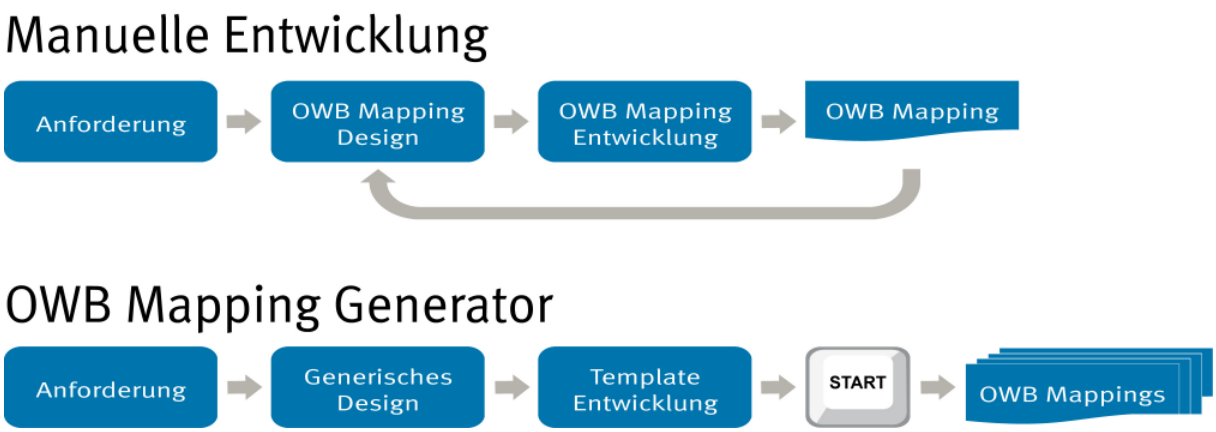

Abb. 1: Prozesskette bei der Erstellung von Mappings mit dem OWB Mapping Generator im Vergleich zur manuellen Entwicklung

An dieser Stelle ist besonders anzumerken, dass es sich bei den Templates nicht um ein programmiertes Skript zur Generierung der Prozesse handelt, sondern genauso wie in OWB und ODI um eine deklarative Definition auf Basis von Metadaten. Die Prozesskette beim Verwenden vom ODI Interface Generator unterscheidet sich im Wesentlichen nicht von der des OWB Mapping Generators.

Bei der Template Entwicklung wird hier aber ein Teil der generischen Logik durch die Verwendung von Knowledge-Modulen bereits abgedeckt.

Einer für alle

Kernstück der Architektur des ETL Generators bilden die bereits erwähnten generischen Templates. Der ETL Generator stellt in der Datenbank einen Satz von Definitionstabellen bereit, in denen die Templates mithilfe der Metadaten beschrieben werden. In den Definitionstabellen findet man keine Tabellen- oder Spaltennamen, die konkreten Objekte werden erst während der Generierung automatisch an die Templates gebunden. Um den Einstieg in das Framework zu erleichtern und die Anlage der Templates zu vereinfachen, besitzen OWB Mapping Generator und ODI Interface Generator jeweils eigene Definitionstabellen. Die Begrifflichkeiten im OWB Mapping Generator sind die gleichen wie im Oracle Warehouse Builder und im ODI Interface Generator wie im Oracle Data Integrator – man findet sich demnach schnell zurecht.

So sind im Datenmodell des OWB Mapping Generators Informationen über Operatoren, Attribute, Properties und Connections enthalten. Das Datenmodell des ODI Interface Generators beinhaltet unter anderem Datasets, Operatoren (z.B. Tabellen, Joiner oder Filter), Properties (z.B. Mappings oder Knowledge-Module) und Optionen. Bei Namen für Tabellenoperatoren, die sich von Prozess zu Prozess unterscheiden und von der zugehörigen Tabelle abhängen, werden Platzhalter verwendet. Die Property-Tabellen können je nach Anforderung oder Komplexität der umzusetzenden Logik sowohl statische als auch dynamische Werte enthalten (vgl. Abb. 2 bis 4 für OWB Mapping Generator). Eine Eigenschaft kann mithilfe vordefinierter dynamischer Parameter festgelegt werden: damit beschreibt man z.B. alle Attribute eines Operators zusammen, und nicht jedes Attribut einzeln. Erfordert die fachliche Logik eine umfassendere Berechnung der Werte, z.B. abhängig vom Primärschlüssel der Tabelle oder von Datentypen der Spalten, hat man im ETL Generator die Möglichkeit eine benutzerdefinierte Funktion anzulegen, die dann in der Property-Tabelle verwendet werden kann.

PROPERTY_NAME:	LOADING_TYPE
PROPERTY_VALUE:	INSERT/UPDATE

Abb. 2: Operatoreigenschaft: statischer Wert

ATTRIBUTE_NAME:	$attr_name
PROPERTY_NAME:	EXPRESSION
PROPERTY_VALUE:	INGRP1.$attr_name

Abb. 3: Attributeigenschaft: dynamischer Wert

PROPERTY_NAME:	SPLIT_CONDITION
PROPERTY_VALUE:	$func_get_scd2_close_set_cond

Abb. 4: Gruppeneigenschaft: benutzerdefinierte Funktion

Ohne Namenskonventionen läuft nichts

Da die Definition der Templates generisch erfolgt, braucht man nun einen Weg, diese mit den erforderlichen Umgebungsobjekten (Schemata, Tabellen usw.) zu verbinden. Um die Generierung von einzelnen Prozessen entsprechend seiner Anforderungen zu ermöglichen, kann man im ETL Generator Namenskonventionen und Umgebungsinformationen ablegen. Dabei spielt der Begriff „Tabellenstamm“ (table radical) eine zentrale Rolle. Damit ist der gemeinsame Teil der Tabellennamen über alle im ETL Prozess verwendeten Schemata hinweg gemeint (siehe Abb. 5).

Schema:	SOURCE	STAGE	CORE
Tabelle:	SRC_PRODUCT	STG_PRODUCT	PRODUCT

Abb. 5: Tabellenstamm „PRODUCT“ in den Schemata SOURCE, STAGE und CORE

Der Tabellenstamm wird bei der Generierung von ETL Prozessen verwendet, um zusammengehörende Objekte in einem Prozess zu verbinden. Die Funktionsweise des Frameworks basiert auf der Annahme, dass alle verwendeten Datenbankobjekte einer allgemeinen Namenskonvention folgen. In den bereitgestellten Konventionstabellen beschreibt man mithilfe der regulären Ausdrücke Namenskonventionen der Datenbankobjekte innerhalb der OWB Module bzw. ODI Modele und legt die Namenskonvention für die zu erzeugenden Mappings bzw. Interfaces fest. Durch die einfache Erweiterbarkeit und Individualisierung des Frameworks können im ETL Generator beliebige Namenskonventionen abgebildet werden.

Generierung ist nicht gleich Generierung

Bei der Generierung der einzelnen Prozesse wird das gewünschte Template aus den Definitionstabellen ausgelesen und die darin gespeicherte Logik mit den Umgebungsinformationen aus den Konventionstabellen angereichert. Ab hier arbeiten OWB Mapping Generator und ODI Interface Generator unterschiedlich. Im Oracle Warehouse Builder steht für die Generierung von Objekten die OMB Plus Sprache zur Verfügung. Der OWB Mapping Generator erzeugt ein TCL (OMB Plus) Skript, mit dem die Mappings anschließend automatisch generiert werden. Im Oracle Data Integrator wird die ODI Java API zur Generierung von Interfaces verwendet. So greift die Java-Schnittstelle auf die Tabellen des ODI Interface Generators zu und generiert die Interfaces direkt. Während der Generierung der Prozesse protokolliert der ETL Generator jeden Schritt in einer Logdatei auf dem Server. Man hat so stets den vollen Überblick über die generierten Objekte im Data Warehouse.

Fazit

Der ETL Generator ist ein speziell entwickeltes Framework, mit dem man die Entwicklung in OWB- und ODI-Projekten „industrialisieren“ und damit den Fertigstellungsprozess eines Data Warehouse enorm beschleunigen kann. Da die Definition der Templates exakt den Strukturen von OWB und ODI folgt, ist eine Einarbeitung in das Framework sehr schnell möglich. Die Vorteile liegen in deutlicher Reduzierung der Entwicklungszeit sowie Vereinheitlichung und damit Qualitätsverbesserung des Codes. Mithilfe des ETL Generators kann ein Data Warehouse in kurzer Zeit neu aufgebaut sowie auf geänderte Anforderungen sehr schnell reagiert werden. Des Weiteren kann durch Übersetzung der Templates eine Migration vom Warehouse Builder zum Data Integrator einfach unterstützt werden.

Kontaktadresse:

Irina Gotlibovych
MT AG
Balcke-Dürr-Allee, 9
D-40882 Ratingen

Telefon:	+49 (0) 2102 309 61-0
Fax:	+49 (0) 2102 309 61-10
E-Mail	irina.gotlibovych@mt-ag.com
Internet:	www.mt-ag.com