

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? Nach einer Reihe durchgeführter Projekte haben wir uns diese Frage gestellt und ein mit Oracle- und OWB-Mitteln entwickeltes Framework erstellt, das es ermöglicht, Mappings im Oracle Warehouse Builder automatisch zu generieren.

Automatische Generierung von OWB Mappings: mehr Zeit für das Wesentliche

Irina Gotlibovych, MT AG

Bei der Entwicklung der ETL-Prozesse in einem Data Warehouse sieht man sich wiederholt vor die Aufgabe gestellt, Prozess-Schritte aufbauen zu müssen, die einer gleichartigen Logik folgen. So werden in jedem Projekt viele Daten-Objekte auf die gleiche Weise aus Quellsystemen in den Arbeitsbereich geladen. Der Transformations-Schritt führt Daten in das einheitliche Format der Zieldatenbank über; gängige Verfahren dabei sind beispielsweise Datentyp-Konvertierung und Daten-Bereinigung. Anschließend werden Daten nach dem gleichen Prinzip – wie etwa Delta Load oder SCD – in das Data Warehouse eingebracht. In der Praxis bedeutet dies 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 den Ausdruck eintragen. Eigenschaften von Operatoren und Attributen sind immer wieder neu zu setzen. Kommt in einer Quelltable später eine neue

Spalte hinzu, muss sie in den meisten Fällen identisch zu den anderen Spalten geladen und verarbeitet werden. Um das zu erreichen, ist aber im entsprechenden Mapping jeder betroffene Operator manuell zu ändern. Besonders aufwändig wird es, wenn sich die grundlegende Logik ändert; im Falle von späteren Änderungsanforderungen muss zumeist jedes Mapping wieder angepasst werden. Obwohl sie der gleichen Logik folgen, ist trotzdem jedes dieser Mappings einzeln zu testen: Da sie unabhängig voneinander entwickelt wurden, können in jedem auch unterschiedliche Fehler auftreten. Bei der manuellen Entwicklung spielt der menschliche Faktor eine enorme Rolle. Sich wiederholende Entwicklungsarbeiten wie das fünfzigfache Anfertigen eines Delta-Load-Mappings sind monoton und führen dadurch zu Fehlern.

Generische ETL-Entwicklung mit dem OWB

Es stellt sich die Frage: „Warum entsteht so ein Mehraufwand bei der manuellen Entwicklung und wie kann man diesen

vermeiden?“ Wäre es nicht schöner, die Logik nur einmal zu entwickeln und diese dann in weiteren Mappings beziehungsweise Projekten mehrmals zu verwenden? Das Problem bei der Entwicklung im OWB besteht darin, dass es keine Möglichkeit gibt, Mappings ohne Bindung an konkrete Objekte (Tabellen, Spalten etc.) anzulegen. Die fachliche Logik eines Mappings ist immer fest mit den Umgebungs-Informationen verbunden. Um die gleiche Logik nicht mehrfach neu erzeugen zu müssen, wäre ein Weg erforderlich, Mappings generisch, also ohne Bezug zu den eigentlichen Objekten definieren zu können. Die Erzeugung der Mappings kann dann automatisch erfolgen, wobei Objekt-Namen als Parameter dem Generierungsprozess mitgegeben werden. Dieser Ansatz liegt bei der Entwicklung des OWB Mapping Generators zugrunde.

Welche Vorteile bringt so ein generischer Ansatz? Angenommen, man möchte Slowly Changing Dimensions Typ 2 in seinem Data Warehouse implementieren. Bei der manuellen Entwicklung würde man für jede Ziel-

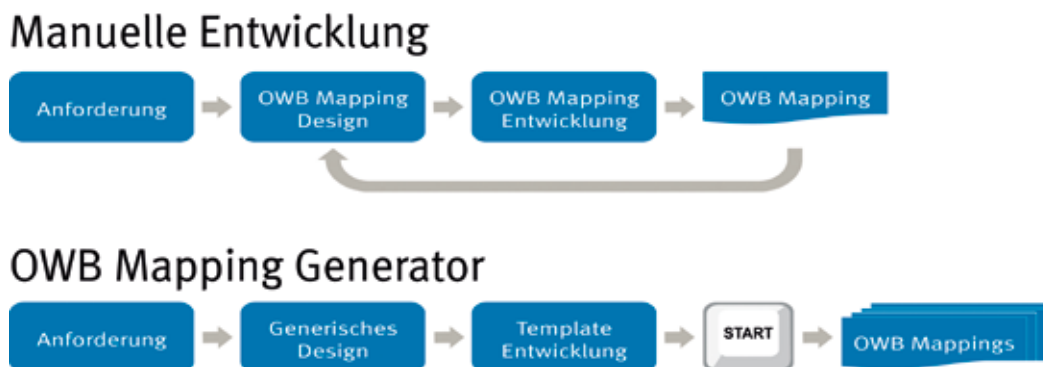


Abbildung 1: Prozesskette bei der Erstellung von Mappings mit dem OWB Mapping Generator im Vergleich zur manuellen Entwicklung

Tabelle die komplexen Join- und Splitterbedingungen in Abhängigkeit von den jeweiligen Primärschlüsseln und Spalten einzeln implementieren. Wählt man den generischen Ansatz, wird die Logik unabhängig von Tabellen und Spalten einmal in Form eines Templates definiert. Der Generierungsprozess arbeitet nun nach allgemeinen Regeln, was identischen Code und gleiche Qualität für alle Mappings garantiert. Und wenn später eine neue Spalte hinzukommt? Da das Template allgemein definiert wurde und die konkreten Primärschlüssel beziehungsweise Spalten erst bei der Verarbeitung aus der Datenbank ausgelesen werden, wird die neue Spalte bei einer erneuten Generierung des Mappings automatisch berücksichtigt. Es ist keine manuelle Anpassung des Mappings im OWB Design Center notwendig. Und wie sieht es mit Fehlerbehebung und Testen aus? Da man die Logik an einer Stelle entwickelt, müssen die Fehler auch nur an einer Stelle

behooben werden – nämlich in dem zugrunde liegenden Template und nicht in jedem Mapping einzeln. Ist man einmal sicher, dass die Logik in dem Template richtig definiert ist, kann man auch sicher sein, dass jedes damit generierte Mapping korrekt laufen wird.

OWB Mapping Generator

Der OWB Mapping Generator ist ein kleines Framework, mit dem sich Mappings im Oracle Warehouse Builder auf Basis von mitgelieferten oder selbstentwickelten Templates automatisch generieren lassen. Die Standard-Implementierung basiert auf Oracle 11g R2 und Oracle Warehouse Builder 11g R2 Basic ETL, eine Anpassung für andere Versionen ist aber problemlos möglich. Bei der Entwicklung wurde Wert auf die Verwendung von Standard-OWB-Features gelegt, um zusätzliche Lizenzkosten zu vermeiden. Im Gegensatz zur manuellen Entwicklung setzt man beim Gebrauch des Frameworks nicht

mehr jedes Mapping im OWB Design Center einzeln um, sondern definiert ein allgemeingültiges Template für eine „Klasse“ von Mappings (siehe Abbildung 1). Anschließend generiert man die dazugehörigen Mappings unter Einbeziehung der Projektvorgaben automatisch. An dieser Stelle ist besonders anzumerken, dass es sich bei den Templates nicht um ein programmiertes TCL-Skript zur Generierung der Mappings handelt, sondern genauso wie im OWB Design Center um eine deklarative Definition auf Basis von Metadaten (siehe Abbildung 2).

Die Generierung der Mappings wird über einen „OWB Expert“ aus der Oracle Warehouse Builder GUI angestoßen. Dieser leitet dialoggestützt durch die einzelnen Schritte. Nachdem man seine Auswahl bezüglich des zu generierenden Templates und der zu verwendenden Objekte getroffen hat, legt der OWB Mapping Generator die Eingaben in den Steuer-Tabellen auf

KeepTool mit neuer Version 10.1

Das handliche Werkzeug für Oracle™-Datenbanken



Zahlreiche neue Funktionen, z.B.

- Erzeugen von AWR-Reports 
bei vorhandener *Lizenz des Oracle Datenbank Diagnostic Packs.*
- Schnelle Textsuche quer über alle Tabellen im Schema.
- Praktische dynamische Tooltip-Hinweise im DataContent.
- Mehrstufige Pivot-Ansicht mit Visualisierung im DataContent.

Laden Sie die kostenlose Testversion unter www.keeptool.com herunter.



keeptool

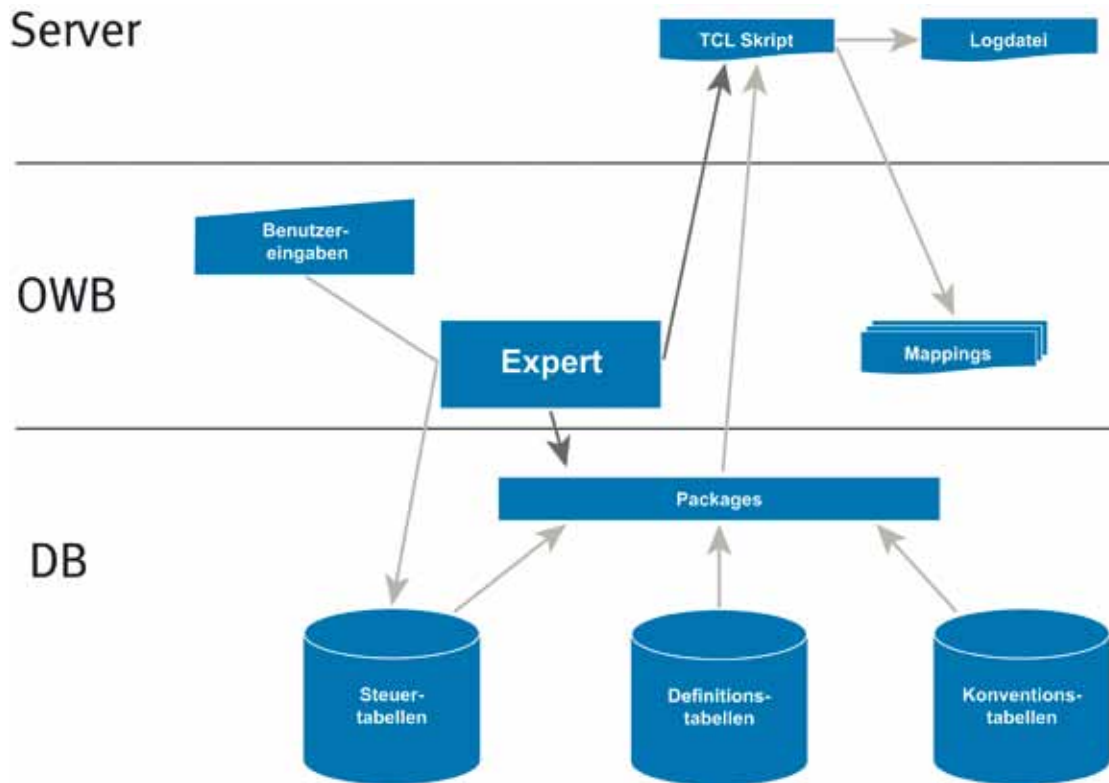


Abbildung 2: Architektur des Frameworks

der Datenbank ab. Danach werden PL/SQL-Prozesse gestartet (siehe Abbildung 2), die aus den Definitions-Tabellen das gewünschte Template auslesen und die darin gespeicherte Logik mit den Umgebungs-Informationen aus den Konventions-Tabellen anreichern. Damit wird nun ein TCL (OMB Plus) Skript generiert, mit dem die Mappings anschließend automatisch erzeugt werden. Das Ergebnis ist sofort im Oracle Warehouse Builder sichtbar und kann weiterverwendet werden. Während der Generierung der Mappings wird jeder Schritt in einer Logdatei auf dem Server protokolliert. Man hat so stets den vollen Überblick über die generierten Objekte.

Ein Template statt vieler Mappings

Kernstück der Architektur des OWB Mapping Generators bilden die bereits mehrfach erwähnten generischen Templates. Der OWB Mapping Generator stellt in der Datenbank einen Satz von Definitions-Tabellen bereit, in denen die Templates mithilfe der Metadaten beschrieben werden. In den Definitions-Tabellen findet man keine Tabellen- oder Attribut-Namen; die konkreten Objekte werden erst

während der Generierung automatisch an die Templates gebunden. Die zentralen Tabellen des Datenmodells enthalten Informationen über Operatoren, Attribute, Properties und Connections. Die Begrifflichkeiten im OWB Mapping Generator sind gleich wie im Oracle Warehouse Builder – man findet sich demnach schnell zurecht. So lassen sich für einen Operator unter anderem Name, Typ und OWB-Modul angeben. Bei den Namen für Tabellen-Operatoren, die sich von Mapping zu Mapping unterscheiden und von der zugehörigen Tabelle abhängen, werden Platzhalter verwendet. Man kann Eigenschaften für Mappings, Operatoren, Gruppen und Attribute definieren. Analog zum Oracle Warehouse Builder sind nur die Eigenschaften zu beschreiben, die nicht automatisch erzeugt werden können. Wenn man beispielsweise im Oracle Warehouse Builder einen Tabellen-Operator mit einem Expression-Operator verbindet, werden die Input-Attribute des Expression-Operators automatisch mit den richtigen Datentypen generiert. Der OWB Mapping Generator arbeitet auf die gleiche Weise. Die Property-Tabelle kann je nach

Anforderung oder Komplexität der umzusetzenden Logik sowohl statische als auch dynamische Werte enthalten (siehe Listing 1 bis 3). Eine Eigenschaft kann mithilfe vordefinierter dynamischer Parameter festgelegt werden: Damit beschreibt man alle Attribute eines Operators zusammen, und nicht jedes Attribut einzeln. Erfordert die fachliche Logik eine umfassendere Berechnung der Werte, etwa abhängig vom Primärschlüssel der Tabelle oder von Datentypen der Attribute, bietet der OWB Mapping Generator die Möglichkeit, eine benutzerdefinierte Funktion anzulegen, die dann in der Property-Tabelle verwendet werden kann.

Da die Generierung der Mappings später mithilfe eines TCL-Skripts (OMB Plus) erfolgt, sind für die Definition von Templates minimale OMB-Plus-Kenntnisse erforderlich. Um komplexe Logiken mit den benutzerdefinierten Funktionen abbilden zu können, sind tiefere TCL-Kenntnisse notwendig.

Ohne Namenskonventionen läuft nichts

Da die Definition der Templates generisch erfolgt, braucht man nun einen Weg, diese mit den erforderlichen OWB-

Objekten (Module, Tabellen etc.) zu verbinden. Um die Generierung von einzelnen Mappings entsprechend seiner Anforderungen zu ermöglichen, kann man im OWB Mapping Generator Namenskonventionen und Umgebungs-Informationen ablegen. Dabei spielt der Begriff „Tabellenstamm“ (table radical) eine zentrale Rolle. Damit ist der gemeinsame Teil der Tabellennamen über alle im Mapping verwendeten Module hinweg gemeint (siehe Listing 4).

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

Fazit

Der Mapping Generator ist ein speziell entwickeltes Framework, mit dem man die Entwicklung in einem OWB-Projekt „industrialisieren“ und damit den Fertigstellungsprozess eines Data Warehouse enorm beschleunigen kann. Da die Definition der Templates exakt der Struktur von OWB Mappings folgt, ist eine Einarbeitung in das Framework sehr schnell möglich.

Das Framework hat nicht den Anspruch, den Oracle Warehouse Builder zu ersetzen, kann aber als Ergänzung bei vielen Aufgabenstellungen sehr hilfreich sein. Wie Abbildung 3 zeigt, lohnt sich der Einsatz des OWB Mapping Generators bei steigender Anzahl der Mappings. Man profitiert dabei in allen Projektphasen:

- Die Entwicklung wird beschleunigt, da statt einer großen Menge von Mappings nur noch ein Template designed werden muss.

PROPERTY_NAME:	LOADING_TYPE
PROPERTY_VALUE:	INSERT/UPDATE

Listing 1: Operatoreigenschaft: statischer Wert

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

Listing 2: Attributeigenschaft: dynamischer Wert

PROPERTY_NAME:	SPLIT_CONDITION
PROPERTY_VALUE:	\$func_get_scd2_close_set_cond

Listing 3: Gruppeneigenschaft: benutzerdefinierte Funktion

OWB Modul:	SOURCE	STAGE	CORE
Tabelle:	SRC_PRODUCT	STG_PRODUCT	PRODUCT

Listing 4: Tabellenstamm „PRODUCT“ in den Modulen „Source“, „Stage“ und „Core“

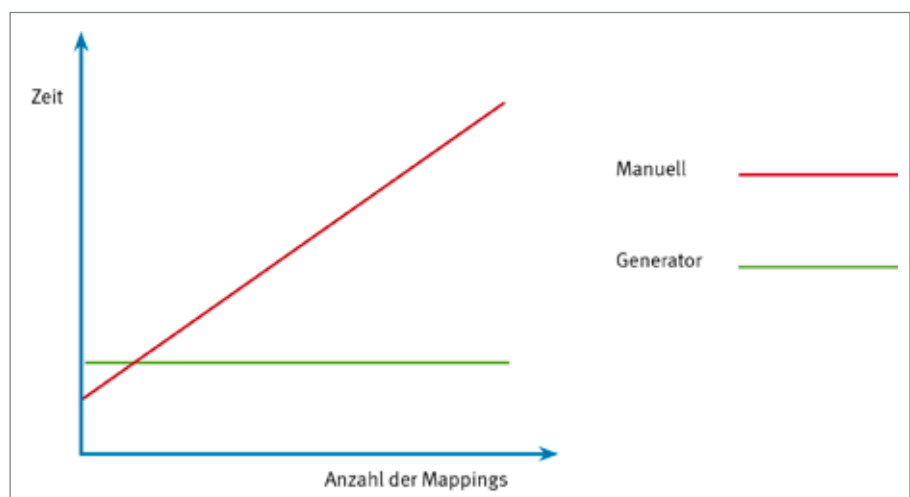


Abbildung 3: Zeitgewinn innerhalb aller Projektphasen beim Einsatz des OWB Mapping Generators

- Vereinheitlichung und damit auch Qualitätsverbesserung des Codes ist ein unstrittiger Gewinn, den man in großen Projekten kaum noch erreichen kann.
- Der Testaufwand wird dank des generischen Ansatzes ebenfalls deutlich reduziert.
- Auf Neuanforderungen kann sehr schnell reagiert und ein Data Warehouse in kurzer Zeit neu aufgebaut werden.
- Die Wartungsaufwände werden bei der Verwendung des OWB Mapping Generators deutlich minimiert, wo-

durch man mehr Zeit für konzeptionelle Aufgaben und neue Projekte gewinnt.

Irina Gotlibovych
irina.gotlibovych@mt-ag.com

