

Automatische Generierung der ETL-Prozesse: OWB vs. ODI

MT AG

business by integration

Die MT AG ist ein IT-Dienstleister, der sich auf Prozess- und Softwareintegration in heterogenen IT-Landschaften seiner Kunden spezialisiert hat.

Als anerkannter Partner der führenden Technologiehersteller verbinden wir die Agilität eines mittelständischen Unternehmens mit der Lösungskompetenz internationaler Beratungshäuser.

Wir bieten daher ein durchgängiges und unabhängiges Portfolio von der Beratung über die Konzeption sowie Umsetzung bis hin zur Betreuung der Systeme.

Neben der regionalen Nähe zu unseren Kunden leisten wir den entscheidenden Mehrwert durch engagierten und kompetenten Einsatz für ihren Geschäftserfolg.

Die MT AG hat ihren Stammsitz in Ratingen und betreut ihre Kunden über diverse Niederlassungen in Deutschland sowie Luxemburg.

MT AG

Daten und Fakten

Inhabergeführte AG

Aktienkapital 1.500.000 €

Gründung

1994

Hauptsitz

Ratingen

Niederlassungen

Hamburg, Dortmund, Frankfurt,
Luxemburg

Tochtergesellschaften

MT-ifs GmbH, MT-ics GmbH

Vorstand

Friedrich Hess (Vorsitzender)
Siegfried Lassak

Aufsichtsrat

Dr. Jürgen Schürenberg
Matthias M. Richter
Rainer Symanski

Beschäftigte (2012)

220 Festangestellte
80 Freiberufler



Agenda

- Nachteile manueller DWH Entwicklung
- Generische ETL Entwicklung
- ETL Generator
 - OWB Mapping Generator & ODI Interface Generator
 - Framework Architektur
 - Definition von Templates
 - Automatische Generierung
- Vergleich und Zusammenfassung

Manuelle DWH Entwicklung

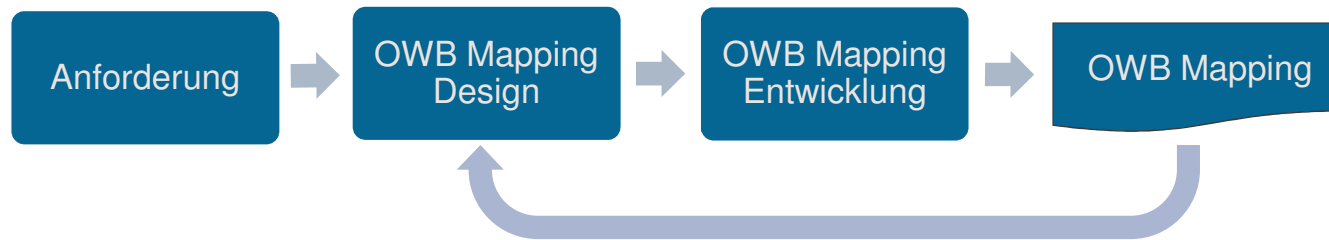
- Gleichartige ETL-Prozesse in vielen Projekten
- Identische Logik in mehreren Prozessen
- Manuelle Erstellung einzelner Objekte
- Umfangreiche Anpassung bei neuen Anforderungen
- Testen jedes einzelnen Prozesses



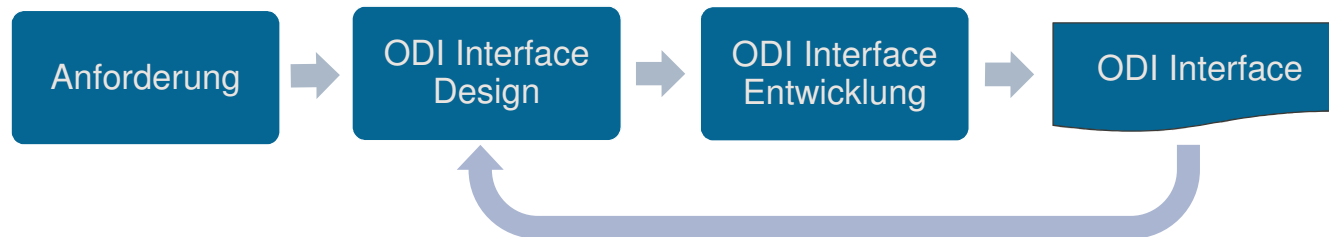
- Hoher Entwicklungsaufwand (auch bei Änderungen)
- Fehler nicht vermeidbar
- Intensives Testen erforderlich

Manuelle Entwicklung: OWB und ODI

- Manuelle Entwicklung mit OWB

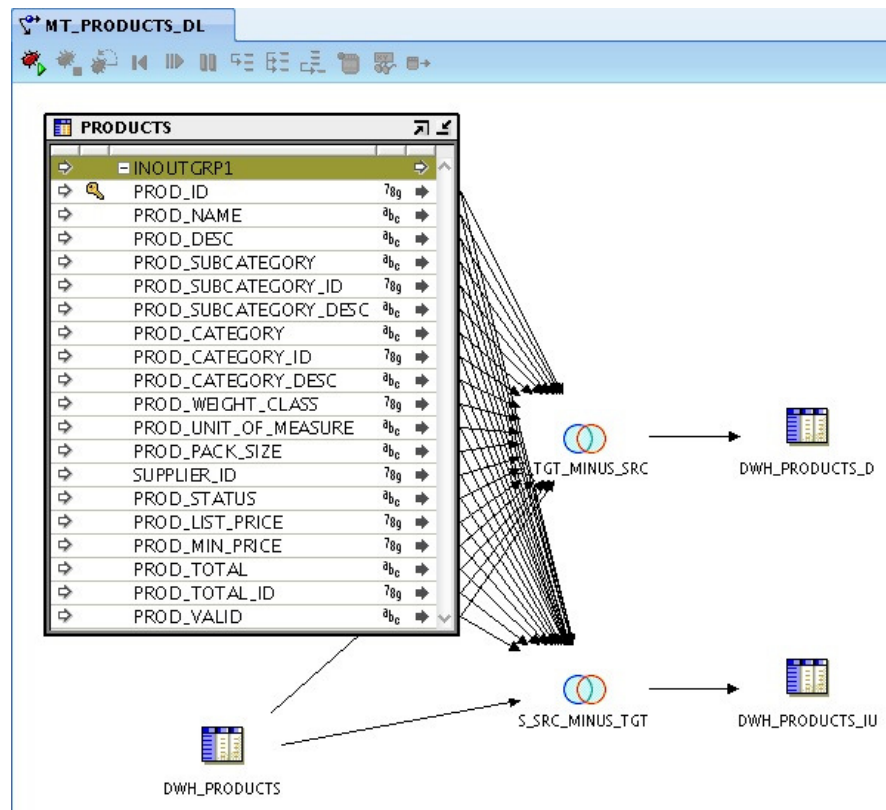


- Manuelle Entwicklung mit ODI



Generische ETL Entwicklung mit dem OWB?

- Problem: Definition ohne konkrete Objekte nicht möglich

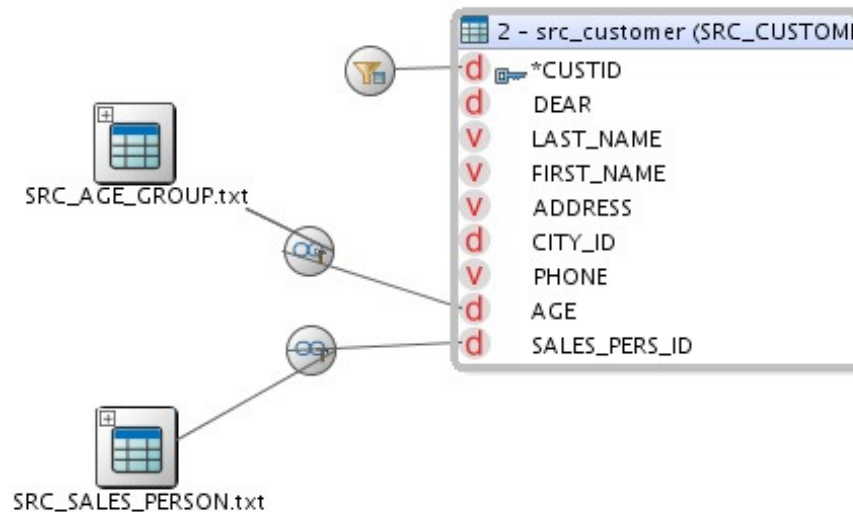


Fachliche
Logik

Umgebungs-
informationen

Generische ETL Entwicklung mit dem ODI?

- Problem: Definition ohne konkrete Objekte nicht möglich



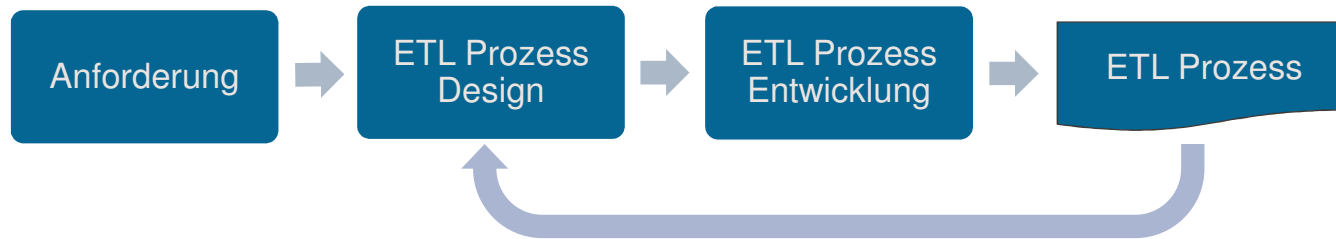
Fachliche Logik

Umgebungs-
informationen

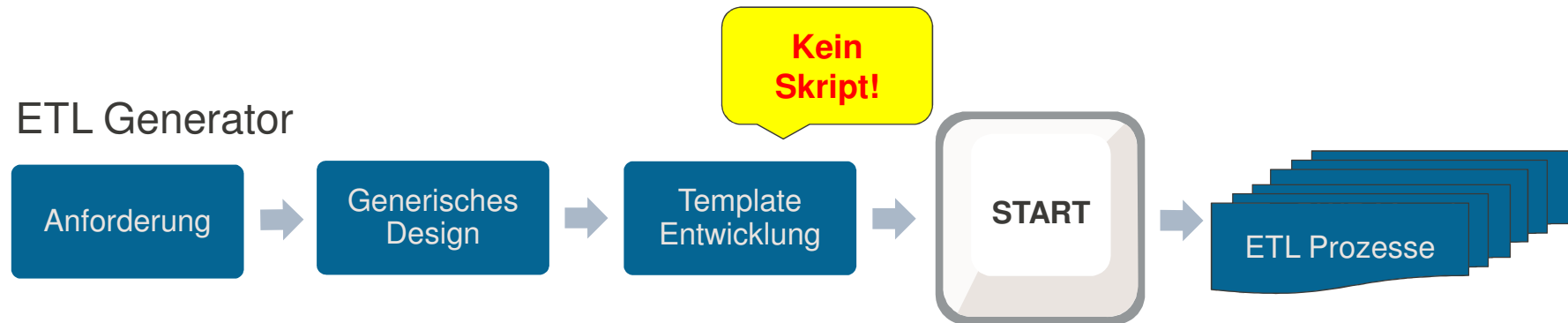
➔ Verwendung von Knowledge Modulen

ETL Generator – Lösungsansatz

- Manuelle Entwicklung



- ETL Generator



ETL Generator – Überblick

- Standard-Implementierung mit Oracle 11g
- Keine zusätzlichen Lizenzkosten
- Anpassung für andere Versionen möglich
- Vordefinierte Templates
- Logik und Steuerung in DB Tabellen

OWB Mapping Generator

- **Technik:** OMB Plus/TCL
- DB Packages zum Generieren des TCL Scripts
- TCL Script zum Generieren der Mappings

ODI Interface Generator

- **Technik:** ODI Java API
- Java Klassen zum Generieren der Interfaces

ETL Generator – Anwendungsbeispiele

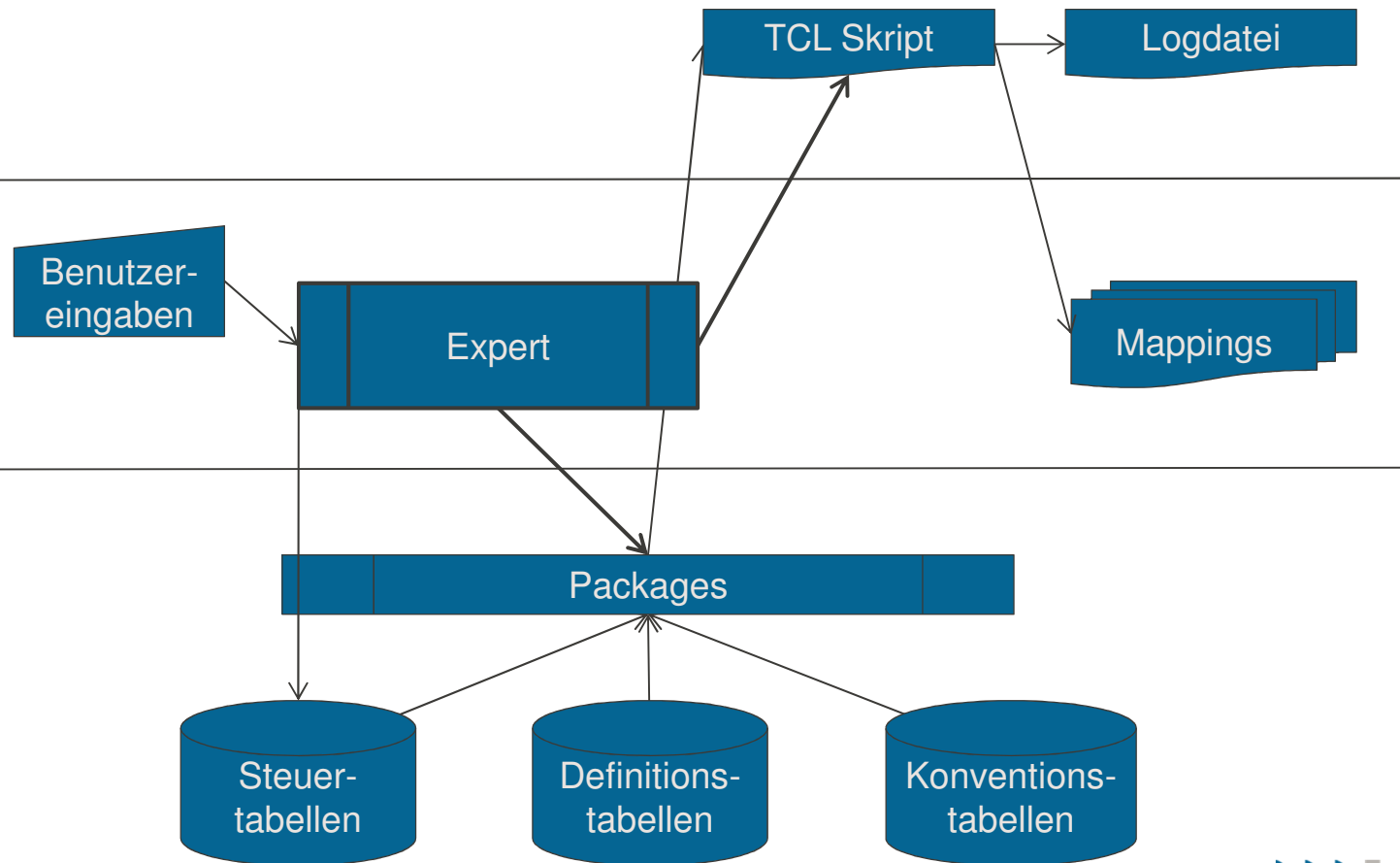
- Schnelles Aufbau der Staging Area, z.B. mit Delta Load
 - Sichere Implementierung einer neuen komplexen Logik, z.B. SCD2
 - Einführung eines einheitlichen Error Handling in allen Prozessen
 - Sofortiges Neuaufbau der Prozesse nach einer allgemeinen Änderung
 - Automatische Propagierung neuer Spalten
-
- Migration von OWB zu ODI durch Template Übersetzung

OWB Mapping Generator – Architektur

- Server

- OWB

- DB

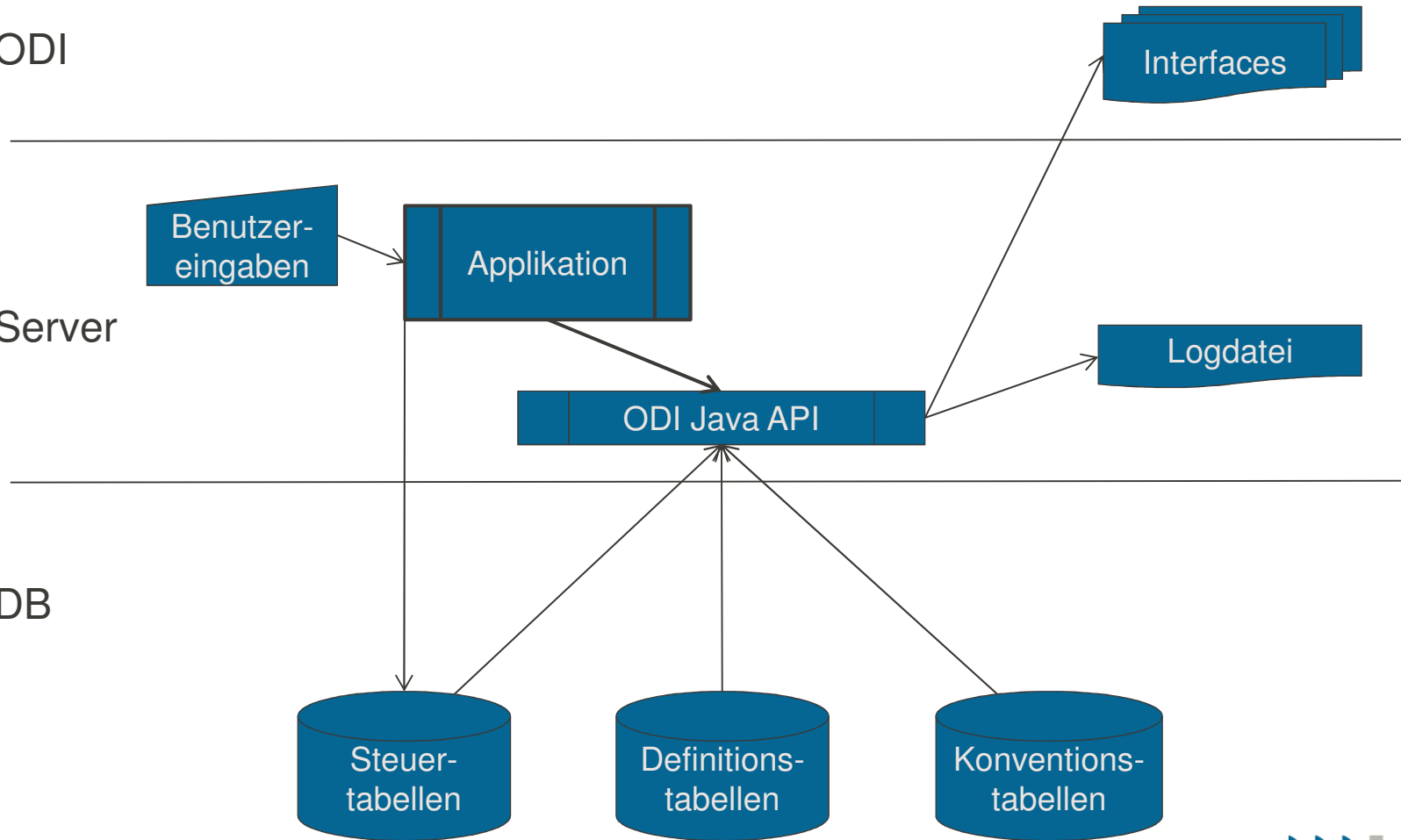


ODI Interface Generator – Architektur

- ODI

- Server

- DB



ETL Generator – Template Entwicklung

- Deklarativ
- Generisch
- Dynamisch



- Allgemeingültig und wiederverwendbar
- Speicherung in DB Tabellen
- Einbindung von Benutzerdefinierten Funktionen
- Definition von Ausnahmen möglich
- Verwendung der Tool-spezifischen Syntax

ETL Generator – Template Entwicklung

Object	Property name	Property value
OWB Table Operator	Operator type	TABLE
OWB Set Operator	SET_OPERATION	MINUS
OWB Expression Operator	EXPRESSION	INGRP1.\$attr_name
ODI Source Datastore	Model	STAGE
ODI DataSet	Operator	UNION
ODI Target Column	Mapping	SYSDATE
ODI Filter	Filter condition	\$func_get_my_filter_cond

Automatische Generierung – Namenskonventionen

- Framework Voraussetzungen
 - Datenbankobjekte folgen einer Namenskonvention
 - „Tabellenstamm“ in allen Schemata vorhanden

SOURCE



STAGE



CORE



- Reguläre Ausdrücke für Namenskonventionen
 - Vordefinierte dynamische Parameter
 - Definition weiterer Parameter möglich
 - Abbildung beliebiger Konventionen !

Automatische Generierung – Namenskonventionen

Namenskonventionen der Datenbankobjekte

MODULE_ID	MODULE	MODULE_ABBR	TABLE_NAMING
SOURCE	SOURCE	SRC	\$owb_module_abbr_\$tab_radical
STAGE	OLTP	STG	\$tab_radical
CORE	SH	(null)	DWH_\$tab_radical

Umgebung beschreiben

Namenskonventionen der OWB/ODI-Objekte

PROJECT	NAMING_RULE
LOAD_ALL	LOAD_\$tab_radical_\$template

Generierungsregeln festlegen

Automatische Generierung – OWB vs. ODI

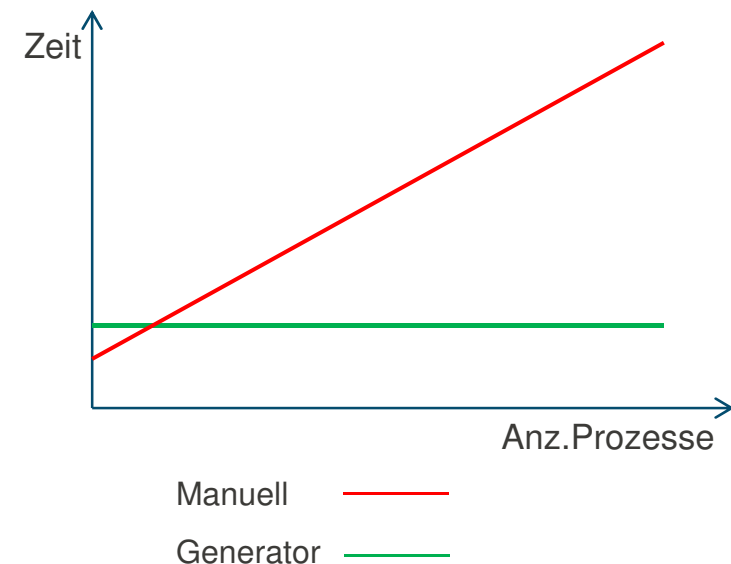
Object	OWB	ODI
Prozess	<pre> LOOP insert_line (OMBCREATE MAPPING <name>); END LOOP; </pre>	<pre> LOOP odInterface = new OdInterface (...,<name>,...); END LOOP; </pre>
Operator	<pre> LOOP insert_line (OMBALTER MAPPING <name> ADD <operator type> OPERATOR <operator name>); END LOOP; </pre>	<pre> LOOP Source table sourceDatastore = ...findByName (<name>,<model>); interfaceHelper.setSourceDataStore (sourceDatastore); Filter interfaceHelper.setSourceFilter (<filter condition>); LOOP </pre>

Automatische Generierung – OWB vs. ODI

OWB Mapping Generator	ODI Interface Generator
Ausführbares Skript wird zur Laufzeit generiert und ausgeführt	Java Funktionen sind vordefiniert und werden zur Laufzeit aufgerufen
Dynamische PL/SQL Implementierung	Statische Java Klassen
Generische OMBPlus Syntax	Feste ODI API Syntax
Logfile wird automatisch generiert	Logging muss implementiert werden

ETL Generator – Zusammenfassung

- Technische Zusammenfassung
 - Generische Templates in Tabellen
 - Kein Skript, Framework generiert ETL-Prozesse automatisch
- Ergebnis und Nutzen
 - Beschleunigung der Entwicklung
 - Vereinheitlichung des Codes
 - Reduzierung des Testaufwands
 - Flexibilität gegenüber Neuanforderungen
 - Minimierung des Wartungsaufwands
 - Wiederverwendbarkeit in neuen Projekten



Besuchen Sie auch unsere weiteren Vorträge auf der DOAG 2012

Dienstag, 12 Uhr, Raum Riga

Dienstag, 13 Uhr, Raum Seoul

Dienstag, 14 Uhr, Raum Stockholm

Dienstag, 15 Uhr, Raum Kopenhagen

Dienstag, 16 Uhr, Raum Stockholm

Mittwoch, 13 Uhr, Raum Riga

Mittwoch, 15 Uhr, Raum Riga

Mittwoch, 16 Uhr, Raum Seoul

Donnerstag, 09 Uhr, Raum Istanbul

Donnerstag, 14 Uhr, Raum Konf. EG

Donnerstag, 15 Uhr, Raum Istanbul

Donnerstag, 16 Uhr, Raum Oslo

Dynamisch Unterschiede in Datensätzen auf Feldebene finden by S.O. Kelbert

Route to ASM by Ernst Leber

Automatische Generierung der ETL-Prozesse: OWB vs. ODI by Irina Gotlibovych

Wiederverwendung von bestehendem PL/SQL Code in ADF Anwendungen by Hendrik Gossens

„Managed Code“ mit OWB – Methoden und Wege by Bernhard Rosenberger

Dateizugriff mit new I/O 2 by Wolfgang Nast

WebServices in Java SE und EE by Wolfgang Nast

Das Mysterium OPatch by Volker Mach

Das größte APEX Projekt der Welt @ Union Investment by Niels de Bruijn

Testen mit Pfefferminzgeschmack by Birgit Kratz

APEX goes UNIT Testing by Oliver Lemm

SOA verspielt – rekursive BPEL Prozesse by Guido Neander



Vielen Dank.

Irina Gotlibovych

Senior System Beraterin

Telefon: +49 (0) 2102 309 61 - 0

Telefax: +49 (0) 2102 309 61 - 10

E-Mail: Irina.gotlibovych@mt-ag.com

www.mt-ag.com