Flexible Schnittstellen für Data Warehousing auf XML-Basis

Lutz Bauer
MT AG
Ratingen

Schlüsselworte

DWH Schnittstellen, XML, Oracle XML DB, DWH Releasemanagement, Oracle Warehouse Builder

Einleitung

Änderungen des Datenmodells in den Quellsystemen erfordern Anpassungen an den Import-Schnittstellen eines Data Warehouse. Ist die gleiche operative Applikation in unterschiedlichen Versionsständen mehrfach innerhalb der Organisation ausgerollt, wird das Management der DWH Schnittstellen rasch komplex. Besondere Tücken entstehen, wenn Umstellungen in den operativen Quellsystemen nicht eng abgestimmt mit dem DWH durchgeführt werden. Ein Releasewechsel im Quellsystem bewirkt oft eine Formatänderung der Datenlieferung. Wird ein Releasewechsel inhaltlich nicht eng abgestimmt bzw. nicht rechtzeitig angekündigt, so sind fehlerhafte Datenimports auf Seite des DWH die Folge.

Der vorliegende Beitrag beschreibt ein Verfahren, in dem die Import-Schnittstelle des DWH automatisch die Version des Quellsystems erkennt und mit dem Oracle Warehouse Builder entsprechend verarbeitet. Die flexible Handhabung des Eingangsdatenformats hat eine höhere Robustheit der DWH Verarbeitung zur Folge. Die technische Basis der Implementierung ist ein kundenspezifisches XML Format.

Ausgangssituation

Eine Organisation besitzt 15 Landesgesellschaften, denen jeweils ein unabhängiger, dezentraler IT Bereich angegliedert ist. Im Rahmen einer Data Warehouse Initiative ist ein zentrales DWH zu implementieren, welches extrahierte Dateninhalte aus den operativen Quell-Applikationen der Landesgesellschaften konsolidiert – mit dem Ziel eine einheitliche zentrale Datensicht zu erhalten. Die Landesgesellschaften setzen weitgehend die gleichen operativen Applikationen ein - es werden jedoch unterschiedliche Versionsstände (Releases) sowie teilweise durch Anpassungen (Customizing) modifizierte Versionsstände eingesetzt.

Eine Gesamt-Koordination der Applikationsstände („Releasemanagement“) innerhalb der Organisation erfolgt nur teilweise. Die operativen Applikation werden zentral entwickelt bzw. vom Markt eingekauft. Die Verantwortung für Integration und Betrieb der Applikationen liegt jeweils dezentral in der einzelnen Landesgesellschaft. Aus Sicht der zentralen IT besteht die größte Herausforderung in der zeitlichen Einhaltung von Releaseplänen. Aus verschiedenen Gründen kann es dazu kommen das eine Landesgesellschaft den geplanten Releasewechsel einer operativen Anwendung selbständig verschiebt. Oft sind technische Abhängigkeiten oder lokale Resourcenknappheit die Ursache.

Aus der Sicht eines zentralen DWH ist diese Situation unkomfortabel:

- Die feste Zuordnung einer Import-Schnittstellenversion zur ausgerollten operativen Applikationsversion einer Landesgesellschaft führt zu hohem Pflegeaufwand im DWH – und ist fehleranfällig

- Die Datenextraktion erfolgt in Verantwortung der dezentralen IT. Die verbindliche Vereinbarung von einheitlichen Schnittstellenformaten mit den dezentralen IT-Einheiten ist eine notwendige Voraussetzung . Durch Customizing der Applikation an lokalen Standorten kann es jedoch aus Sicht des DWH zu ungültigen Datenextrakten bzgl. des Dateformats kommen (Bsp.: geänderte Feldlängen)

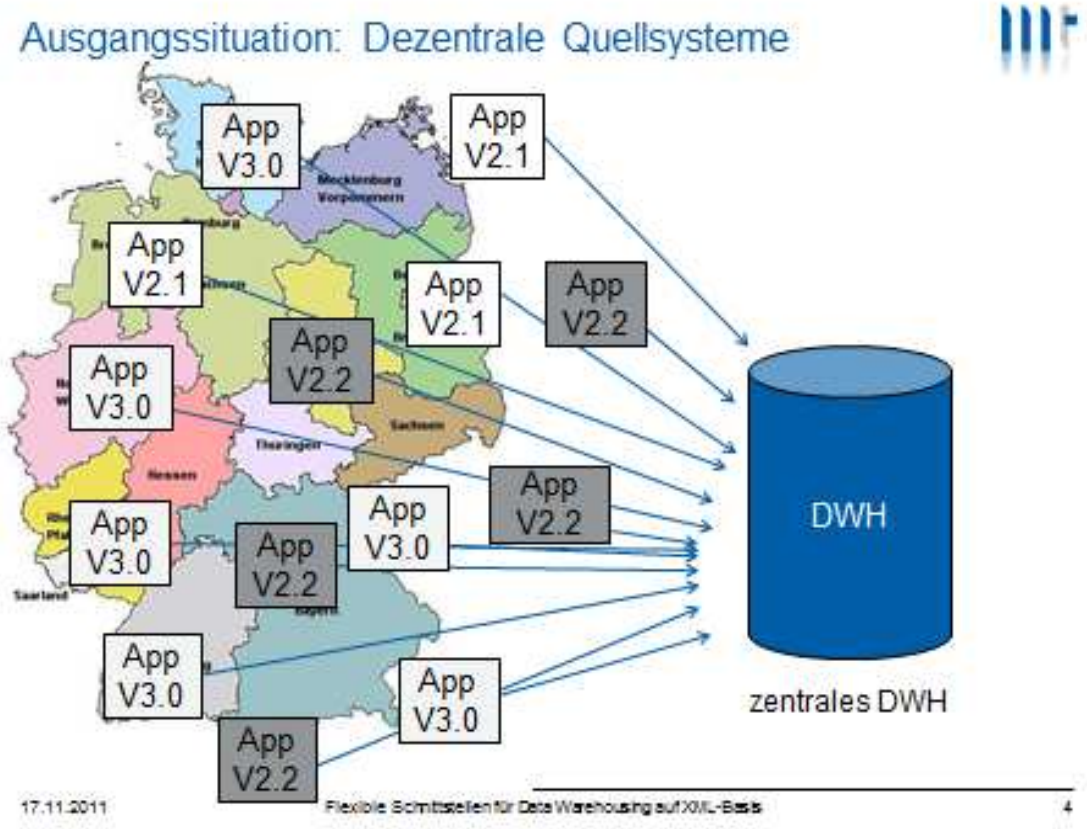


Abbildung 1 Dezentrale Quellsysteme mit unterschiedlichen Versionsständen der Quellapplikationen

Anforderungen an flexible DWH Schnittstellen

Während der Planungsphase des zentralen DWH wurde der Entschluss gefasst, daß eine feste Zuordnung der Releasestände in den Landesgesellschaften für die DWH Import-Schnittstelle zu vermeiden ist. Stattdessen fiel die Entscheidung für eine flexible Schnittstelle mit den folgenden Anforderungen:

- Die DWH Import-Schnittstelle unterstützt mindestens 3 unterschiedliche Versionsstände einer operativen Anwendung

- Die Version einer Dateianlieferung wird vom DWH automatisch erkannt. Die zur jeweiligen Version passenden ETL-Jobs für die Beladung der Staging Area werden automatisiert ausgewählt.

- Für jeden unterstützten Versionsstand der jeweiligen Quellapplikation ist genau ein Datenaustauschformat zentral festzulegen. Individuelle Anforderungen einzelner Landesgesellschaften sind zu prüfen und ggf. in den gemeinsamen Standard aufzunehmen

- Das Datenformat der Datenextraktion aus dem Quellsystem ist auf Seite der dezentralen IT leicht zu validieren. Hierzu wird ein Mechanismus zur Verfügung gestellt. Vor dem Versand der Daten an das DWH ist vom Quellsystem der Datenextrakt auf Gültigkeit zu prüfen.

- Doppelte Prüfung: die Datenanlieferungen werden auch auf der Seite des DWH vor dem Ladevorgang validiert – und ggf. zurückgewiesen

Die Anforderungen wurden so gewählt, um eine möglichst lose Kopplung zwischen den dezentralen Quellsystemen und dem zentralen DWH zu erhalten.

Lösung auf XML-Basis

Die Wahl für die Umsetzung der oben beschriebenen „flexiblen Schnittstelle“ fiel schnell auf XML. Im Gegensatz zur traditionellen Implementierung von Schnittstellen per Flat File Formaten (z.B. csv) bietet XML den Vorteil, daß durch die Definition eines XML-Schemas das Format der Extraktionsdateien auf einfache Weise auch außerhalb der Applikation oder dem DWH geprüft werden kann. Anhand eines XML Parsers kann durch die dezentrale IT vor Versand der Extraktionsdateien die Gültigkeit des Datenformats validiert werden.

XML Schemafiles erlauben u.a. die Definition von: Pflichtfeldern, Datentyp, Feldlängen, Domänen uvm. Hierdurch sind die Anforderungen für Prüfungen zur Beladung einer DWH Staging Area ohne weiteres abzubilden. Die semantische Prüfung der Daten, z.B. auf referentielle Integrität oder weitere Datenqualitätskriterien kann in XML nicht abgebildet werden – dies erfolgt dann bei späteren ETL-Verarbeitungsschritten innerhalb des DWH.

Die erzeugten XML Dateien beziehen sich im Fileheader auf das jeweilige XML Schema. Für das beschriebene Projekt wurden für jede Version der Quellapplikation eine eigene „Version“ von XML-Schemafiles definiert. Aus der Referenz auf die jeweilige Schemafile-Version lässt sich somit für jedes XML-File die Version der Quellapplikation ablesen.

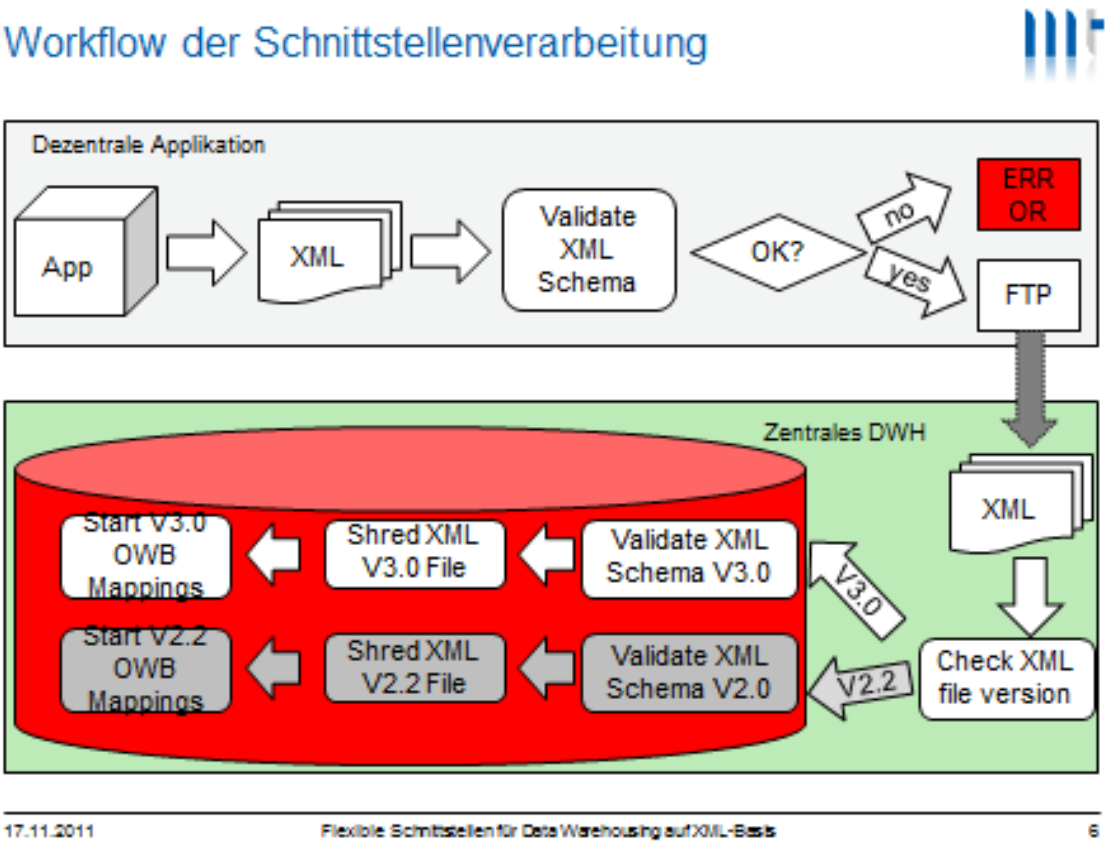
Im folgenden XML Fragment lässt die Referenz auf die Schema-Version V2.2 den Schluß zu, daß das dezentrale Quellsystem in der Version 2.2 betrieben wird.

```
<KUNDE xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:SchemaLocation="http://dwh.dummy.de/Kunde_SchemaV2_2.xsd">

                <KUNDE_ID>17</KUNDE_ID>
                <KUNDE_FIRMA>Xyz_AG</KUNDE_FIRMA>
                …

</KUNDE>
```

Die Erkennung der Schema-Version wird beim Ladevorgang in das DWH über einen einfachen Text-search (hier per grep) durchgeführt. Anschließend wird die XML Datei mit der referenzierten XML-

Schemaversion validiert. Wurde diese Prüfung erfolgreich beendet, so erfolgt das sogenannte „Shredding“ des XML-Files d.h. die Übertragung der (hierarchischen) XML Darstellung in eine relationale Darstellung – in diesem Fall die Tabellen der DWH Staging Area. Für diese Aufgabe wird die Funktionalität der Oracle XML DB genutzt. Die Verarbeitung wird durch den Oracle Warehouse Builder (OWB) gesteuert – Details werden im folgenden Abschnitt genannt.

Der gesamte Ablauf von der Datenextraktion bis zur Beladung der Staging Area sieht wie folgt aus:



Bei der Verwendung von XML-Dateien sind zwei wesentliche Eigenschaften zu beachten: deutlich höheres Datenvolumen wie z.B. im Vergleich zu csv sowie die erhebliche Laufzeit für den XML-Parsing/Validierungs-Vorgang. Insbesondere für große Dateien kann der Parsing-Vorgang die vertretbare Grenze überschreiten. Im angesprochenen Projekt sind beide Faktoren zu vernachlässigen, da das täglich extrahierte Datenvolumen pro Quellsystem vergleichsweise gering ist (wenige MB).

Details zur Umsetzung

Das zentrale Data Warehouse wurde auf Basis Oracle 10gR2 sowie Oracle Warehouse Builder 10gR2 aufgebaut. Die Generierung der XML Files auf Seite der Quellsysteme erfolgt mit verschiedenen Mitteln. Bei einigen Applikationen werden Java-basierte Applikationen zur Erzeugung der XML-Files eingesetzt. Andere Anwendungen basieren ebenfalls auf einer Oracle 10gR2 Datenbank – hier wird XML anhand von SQL/XML erzeugt. Beispiele hierzu finden sich z.B. unter [CZ1]. Die XML Validierung der extrahierten Dateien erfolgt vor dem Upload mit einer zentral entwickelten Java Anwendung.

Die XML Schemafiles werden mit dem Produkt Altova XML Spy erzeugt und gepflegt. Auf der Seite des Data Warehouse wurden die Schemafiles anhand der Oracle XML DB registriert (dbms_xmlschema.registerschema) . Auf diese Weise kann die Validierung der XML Files innerhalb der Oracle XML DB erfolgen (<XMLTYPE>.xml.isSchemaValid). Die Performance der XML Validierung ist hier aufgrund von kleinen Dateigrößen absolut unkritisch.

Die Speicherung der XML Files in eine relationale Darstellung (das sogenannte „Shredding“) wurde über SQL/XML Queries innerhalb von PL/SQL Packages implementiert. Nach einer erfolgreichen Validierung einer XML-Datei werden die betreffenden „Shredding“ PL/SQL Prozeduren durch OWB Process Flows aufgerufen. Beispiele hierzu sowie zur XML Validierung und Handhabung von XML Schemata finden sich ebenfalls unter [CZ1].

Die Auslagerung der Funktionalität zur XML-Verarbeitung in PL/SQL ist im Rahmen einer Oracle Warehouse Builder Implementierung nicht unbedingt anzustreben – da die Implementierung der ETL Logik so auf unterschiedlichen Ebenen liegt. Es ist möglich die SQL/XML Funktionen für das „Shredding“ von XML Daten durch die Verwendung von OWB Mapping Operatoren zu kapseln (Expressions, VArray Iterations, …) siehe [DA1]. Diese „advanced Operators“ setzen jedoch die Lizensierung OWB-EE voraus – im bestehenden Projekt wurde aus Kostengründen der Weg über PL/SQL gewählt.

Lessons Learned

- Die Oracle XML DB Implementierung ist im Stand 10gR2 robust – und für die hier verarbeiteten Datenmengen performant.
- Die Implementierung der XML Extraktion sowie des Shreddings über SQL/XML lässt sich leichter auf dem Weg der manuelle PL/SQL Implementierung als per OWB Mappings durchführen. Eine Unterstützung z.B. durch Generierung von SQL/XML Code basierend auf einem XML-Schema (XSD-File) durch den OWB wäre sehr wünschenswert.
- XML Unterstützung des OWB ist in der Basic-ETL Variante nicht sehr ausgeprägt. Es existiert ein Expert (siehe [DA2]) – dieser hat sich bei komplexen XML Schemata jedoch leider nicht als robust erwiesen. Weiterhin erzeugt der XML-Expert OWB Mappings unter der Verwendung von Pluggable Mappings, dies setzt die Lizensierung der entsprechenden OWB Lizenzoption („Enterprise ETL Option“) voraus. Eine weitere Alternative bietet der ODI XML JDBC Driver innerhalb OWB 11gR2 – hierfür ist jedoch ebenfalls die Lizensierung der Oracle Data Integrator Enterprise Edition notwendig.

Verweise

- [CZ1] Carsten Czarski, Oracle, "Schnittstellen mit Oracle XML DB", http://apex.oracle.com/folien Schlüsselwort: schnittstellen
- [DA1] David Allan, Oracle, "Leveraging XDB (with Oracle Warehouse Builder)“ http://blogs.oracle.com/warehousebuilder/entry/leveraging_xdb
- [DA2] David Allan, Oracle, Expert zur Erzeugung von OWB Mappings aus XML Schema-Files (Voraussetzung: OWB Enterprise ETL Option): http://www.oracle.com/technology/products/warehouse/htdocs/Experts/owb_xml_etl_utils.zip

- [OR1] Materialien zur Oracle XML DB: http://www.oracle.com/technetwork/database/features/xmldb/index.html

Kontaktadresse:

Lutz Bauer
MT AG
Balcke-Dürr-Allee 9
D-40882 Ratingen

Telefon: +49 (0) 2102 309 61-0
Fax: +49 (0) 2102 309 61-101
E-Mail Lutz.Bauer@mt-ag.com
Internet: www.mt-ag.com