Verwaltung von OBI Metadaten: XML-Integration − die Lösung aller Probleme?

Michael Weiler

PROMATIS software GmbH
Ettlingen

Schlüsselworte
Oracle Business Intelligence Enterprise Edition (OBIEE), OBIEE Repository, Subversion Repository, Administrationswerkzeug, MDS XML, Multi User Development (MUD), RPD-Datei, Physische Schicht, Logische Schicht, Präsentationsschicht.

Einleitung

Die Entwicklung umfangreicher analytischer Projekte mit Oracle Business Intelligence Enterprise Edition (OBIEE) setzt eine effiziente Quellenverwaltung voraus. Etabliert haben sich Open Source Code-Verwaltungswerkzeuge wie Subversion oder CVS.
Mit OBIEE 11.1.1.6 wurde die Möglichkeit der Integration von Metadaten in ein Source Code-Verwaltungswerkzeug geschaffen. Dieses oftmals unbekannte Feature wird im Rahmen dieses Beitrags näher vorgestellt. OBI verwaltet die Repository Metadaten in einer großen binären Datei, die Informationen zum physischen Zugriff, dem logischen Datenbankschema und der Präsentationsschicht enthalten. Diese RPD-Datei kann in vielen unterschiedlichen XML-Dateien gespeichert werden. Und diese Dateien wiederum bilden die Basis für die Etablierung der Änderungsverwaltung von Metadaten.

Anhand eines Fallbeispiels mit der neuen Version OBIEE 11.1.1.7 und der Integration mit Subversion wird aufgezeigt, wie eine effiziente Metadatenverwaltung aufgebaut werden kann. Zunächst erfolgt eine Einführung in die Datenstrukturen und deren Zusammenhänge der Metadatendateien im XML-Format. Danach wird veranschaulicht, wie diese Dateien unter einer Source Code-Verwaltung bereitgestellt werden können. Abschließend bewertet der Autor, welche Möglichkeiten diese Integration bietet, aber auch welche Grenzen diese Technik besitzt.

OBIEE Metadatenverwaltung

Den Kern der OBIEE Plattform bildet der Oracle BI-Server. Er stellt leistungsfähige Business Intelligence- und Analysefunktionen zur Verfügung. Mit seiner intelligenten Abfragefunktionalität ermöglicht der Oracle BI-Server den effizienten Zugriff auf unterschiedliche Datenquellen. Die Zugriffsdefinitionen werden in einer physischen Schicht definiert. Dabei werden insbesondere die Verbindungsparameter und die Schemata definiert, auf die zugegriffen werden soll. Dabei können sowohl multidimensionale als auch relational vorliegende Daten genutzt werden.
Aufbauend auf der physischen Schicht wird eine logische Sicht (Geschäftsmodell und Zuordnung) der Datenelemente erzeugt. Dabei kommen die aus dem OLAP bekannten Konstrukte wie Dimensionen, Kennzahlen, Hierarchien, Berechnungen, Aggregationsregeln und Zeitreihenanalysen zum Einsatz. Diese Objekte können in der Präsentationsschicht beliebig nach fachlichen Kriterien zusammengestellt werden. Die Präsentationsschicht stellt die Schnittstelle zum Anwender dar. Alle diese Informationen werden als Metadaten abgespeichert. Zentrales Werkzeug für die Verwaltung der Metadaten ist das Oracle BI-Administrationswerkzeug (siehe Abbildung 1).

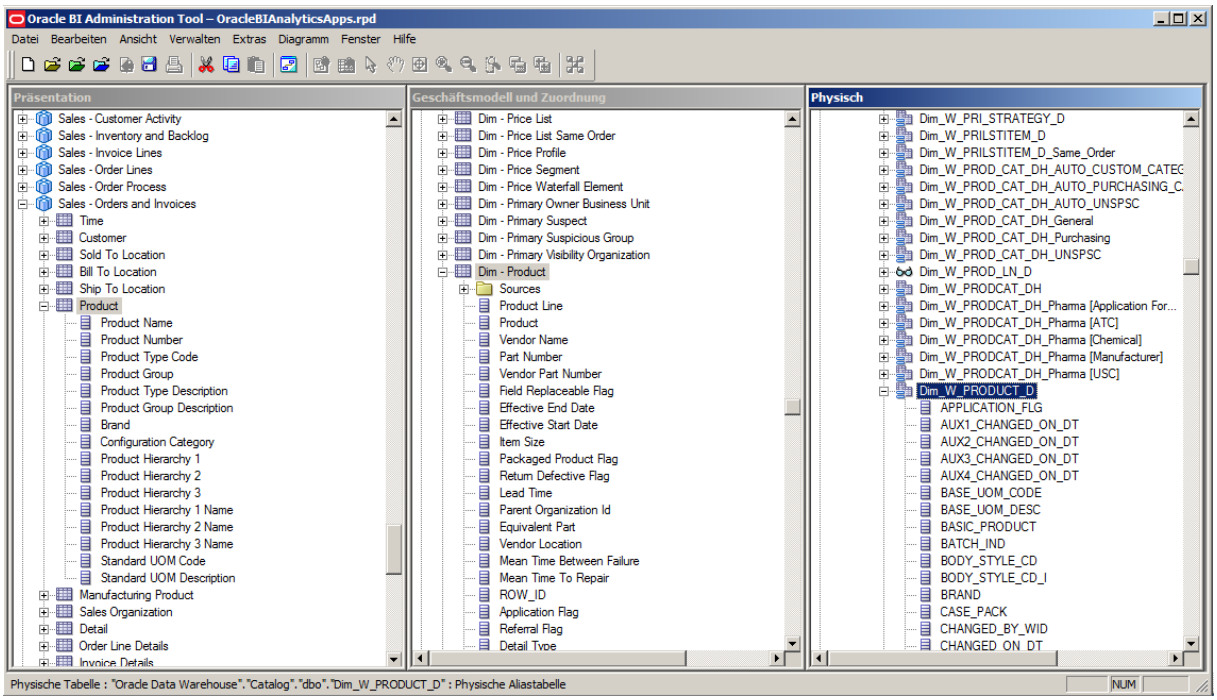

Abb. 1: Oracle BI-Administrationswerkzeug

Die Entwicklung der Metadaten geschieht in der Regel in Zyklen und wird den Endanwendern nach vorherigen Tests zu einem bestimmten Stichtag zur Verfügung gestellt. Die Metadaten stellen das Herzstück eines jeden OBIEE Projekts dar. Daher sollten diese Daten unbedingt über ein Source Code System verwaltet werden. Je nach Größe und Vielzahl der anzubindenden Systeme können die Metadaten eine beachtliche Größe erreichen. Die Metadaten speichert das Oracle BI-Administrationswerkzeug in einer binären Datei ab. Diese Datei wird häufig als RDP-Datei bezeichnet.

Bei großen Systemen mit unterschiedlichen Datenquellen ist es nicht unüblich, dass sich mehrere Administratoren die Arbeit für die Erstellung und Pflege der Metadaten teilen. Oracle stellt mit der Funktion „Multi User Development“ (MUD) eine Auftrennung der RPD-Datei in unterschiedliche Teilbereiche vor. Hierzu stehen Funktionen wie Check-out und Merge im Administrationswerkzeug zur Verfügung. Dieser Beitrag beschäftigt sich nicht mit der Konfiguration und der Nutzung von MUD. In den Oracle Handbüchern findet man detaillierte Anleitungen zum Aufsetzen und dem Betrieb einer solchen Umgebung. MUD gab es schon in der Version 10*g* von OBIEE und ist auch heute noch verfügbar.

In diesem Beitrag wird die Speicherung der Metadaten als XML-Datei vorgestellt, und es wird erläutert, wie die Integration in ein Source Code-Verwaltungswerkzeug (z.B. Subversion) erfolgt.

XML-Struktur der OBIEE Metadaten

Mit dem Administrationswerkzeug kann eine bestehende RPD-Datei einfach als XML-Struktur abgespeichert werden (siehe hierzu Abbildung 2). Das Kürzel MDS steht hierbei für „Meta Data Services“.

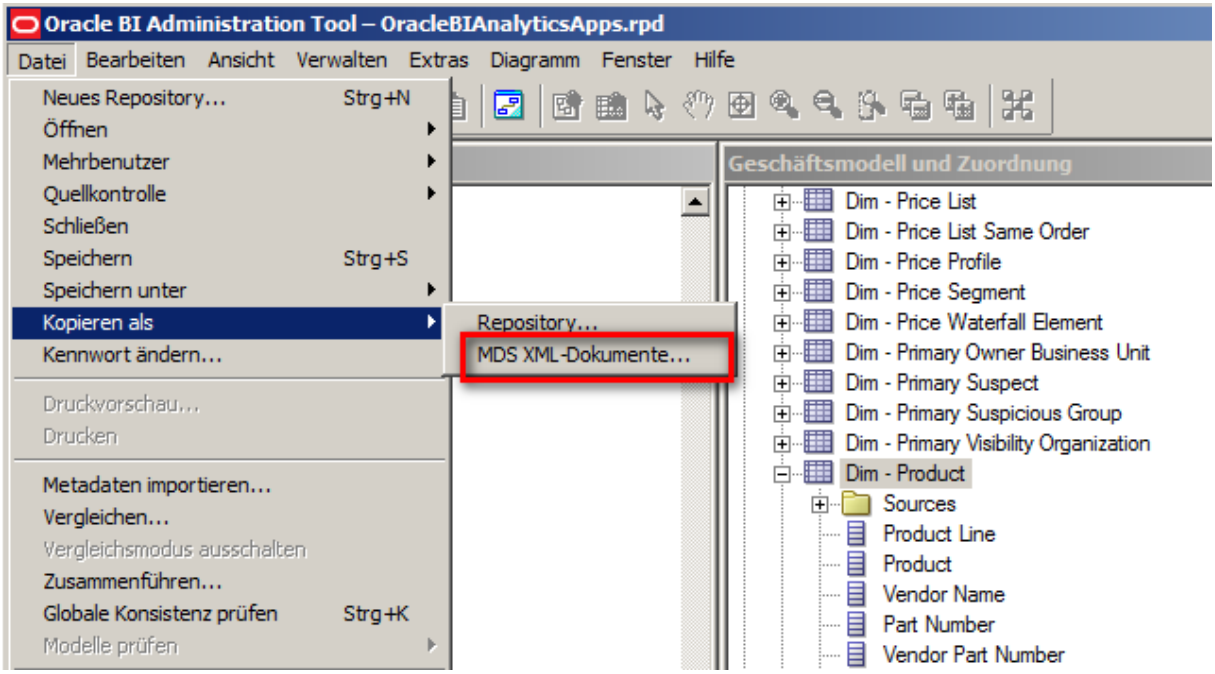

Abb. 2: Oracle BI-Administrationswerkzeug

Es werden XML-Dateien für alle drei zuvor beschriebenen Bereiche (Physische Schicht, Logische Schicht, Präsentationsschicht) erstellt. Dabei werden die XML-Dateien in Verzeichnisse abgelegt. Ein Beispiel für solch eine Verzeichnisstruktur wurde in Abbildung 3 dargestellt. Die Verzeichnisstruktur ist eindimensional, das heißt es existieren keine Unterverzeichnisse. Es ist nicht direkt ersichtlich, welche Verzeichnisse zu welcher Schicht gehören. Die Namen der Dateien sind generiert.

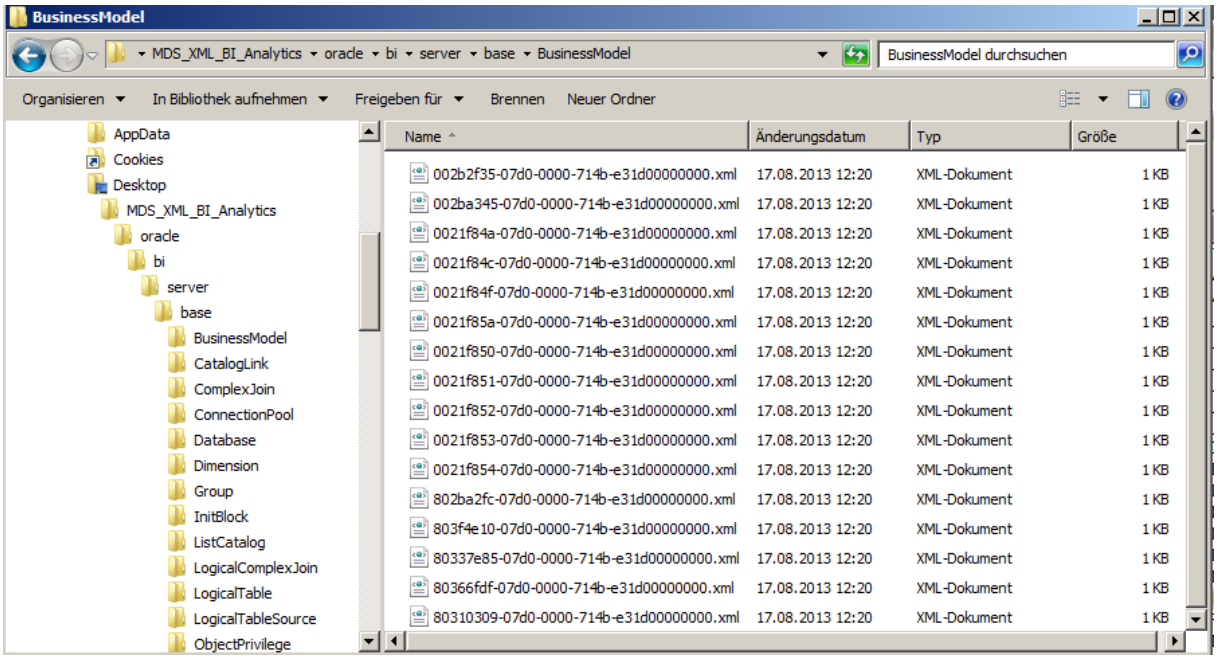

Abb. 3: Verzeichnisstruktur für die MDS XML-Dateien

Beispielhaft soll der Aufbau einer solchen XML-Datei am Beispiel des Objekts Datenbank aus der physischen Schicht erläutert werden. Weiterführende Informationen zu den XML-Dateien findet man im Handbuch „Oracle® Fusion Middleware, XML Schema Reference for Oracle Business Intelligence Enterprise Edition“.

```
<Database
      name=""
      xmlns:xsi=""
      xmlns=""
      type=""
      dbFlag=""
      containerRef=""
      persistConnectionPoolRef="">
      <Description />
      <Features>
            <Feature name="" value=""/>
            <Feature name="" value=""/>
            ...
      </Features>
      <RefConnectionPools>
            <RefConnectionPool connectionPoolRef="" …/>
            <RefConnectionPool connectionPoolRef="" …/>
      </RefConnectionPools>
      <RefDisplayFolders>
            <RefPhysicalDisplayFolder physicalDisplayFolderRef="" …/>
            <RefPhysicalDisplayFolder physicalDisplayFolderRef="" …/>
      </RefDisplayFolders>
      <Properties>
            <PropertyGroup category="">
                  <Property>
                        <Name …/>
                        <Value …/>
                  </Property>
            </PropertyGroup>
      </Properties>
</Database>
```

Jedes Objekt beginnt mit einem Objekt-Tag gefolgt von einer eindeutigen „mdsid“, die dieses Objekt identifiziert (`<Database mdsid="m80c882ea-0bcf-0000-714b-e31d00000000"`). Danach werden für jedes Objekt Name und Verweise auf die Strukturinformation abgelegt:

```
name="Loyalty Input Data Source"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns="http://www.oracle.com/obis/repository"
```

Danach beginnt der objektspezifische Teil – in diesem Fall der Typ der Datenbank wie beispielsweise ODBC20, Oracle etc. Diese objektspezifischen Felder sind in der oben angegebenen Dokumentation enthalten. Wiederholungsgruppen werden in übergreifenden Tags wie beispielsweise die Liste von Funktionen, die eine Datenbank bereitstellt, eingeschlossen (<Features> … </Features>). Verweise zwischen Objekten werden über Referenzen dokumentiert. Dabei wird in der Referenz der vollständige Pfad auf die Datei und die „mdsid“ angegeben. In dem Beispiel zeigt die Subject Area Referenz auf das zugehörige Business Modell und für die Spalte wird auf die physische Tabelle verwiesen (siehe Abbildung 4).

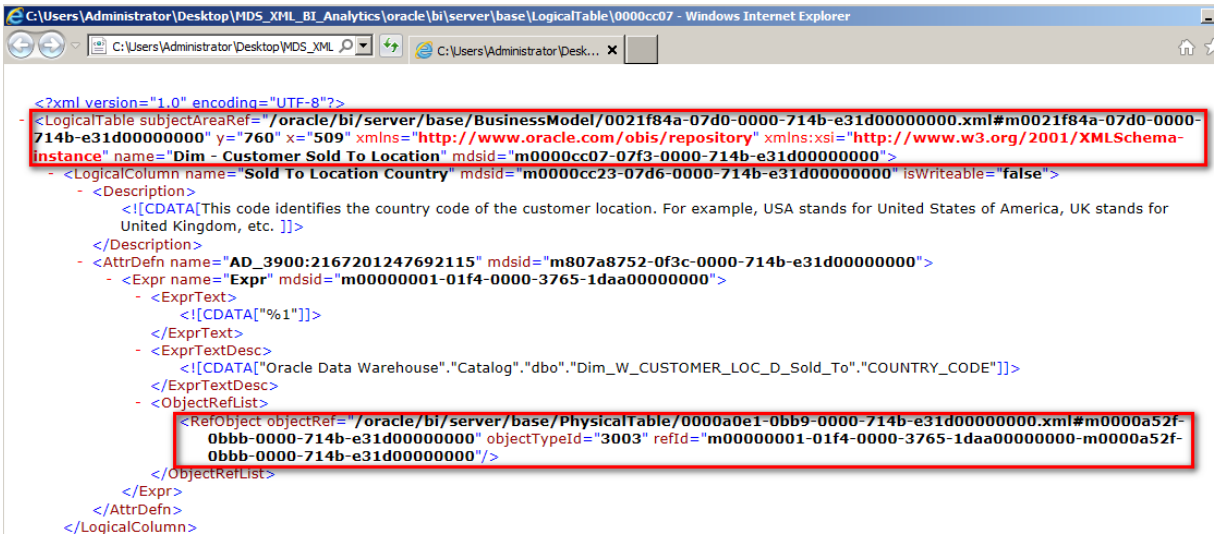

Abb. 4: Beispiel Logical Table mit Verweisen

Am Rande sei erwähnt, dass die Struktur und der Aufbau der MDS XML-Dateien sich deutlich unterscheiden von der XML-Struktur, die der BI-Server nutzt. Die xsd-Dateien, die die Strukturen und Zusammenhänge zwischen den Objekten definieren, finden sich unter ORACLE_HOME/bifoundation/server/bin und heißen xudml_mds_admin.xsd, xudml_mds.xsd und xudml_mds_core.xsd.

Fallbeispiel Metdatenverwaltung mit Subversion

Die neuen MDS XML-Dateien können in eine Source Code-Verwaltung wie Subversion integriert werden. Hierzu muss zunächst ein solches System zur Verfügung stehen. Im Fallbeispiel wurde eine einfache Struktur in der Source Code-Verwaltung definiert:
Zur Konfiguration des Administratorwerkzeuges muss im Menü „Extras“ der Punkt „Optionen“ und in dem sich öffnenden Fenster der Bereich "Quellkontrolle" ausgewählt werden. Es erscheint ein Fenster über die die Bearbeitung einer bestehenden Konfiguration oder das Anlegen einer Konfiguration möglich ist. Grundsätzlich kann das Administrationswerkzeug mit beliebigen Source Code-Verwaltungswerkzeugen arbeiten.

Im Fallbeispiel legen wir eine neue Konfigurationsdatei an. In der obersten Zeile ist eine XML-Datei anzugeben, in der die Konfiguration gespeichert wird – in diesem Fall Test.xml. Im Standard stehen die Dateien unter dem angegebenen Verzeichnis zur Verfügung:
ORACLE_INSTANCE/config/OracleBIServerComponente/coreapplication_obisn

Mit dem Speichern muss zunächst eine leere Datei angelegt werden. Danach können die Befehle für das Anlegen neuer Verzeichnisse oder Dateien und für den Check-out angegeben werden. Für Subversion steht ein Template von Oracle auf Oracle Technet zur Verfügung (scm-conf-svn.template.xml). Über die Schaltfläche "Wird geladen" kann ein solches Template eingelesen werden. Danach ist der Dialog mit den korrekten Befehlen für ein Subversion Repository ausgefüllt.

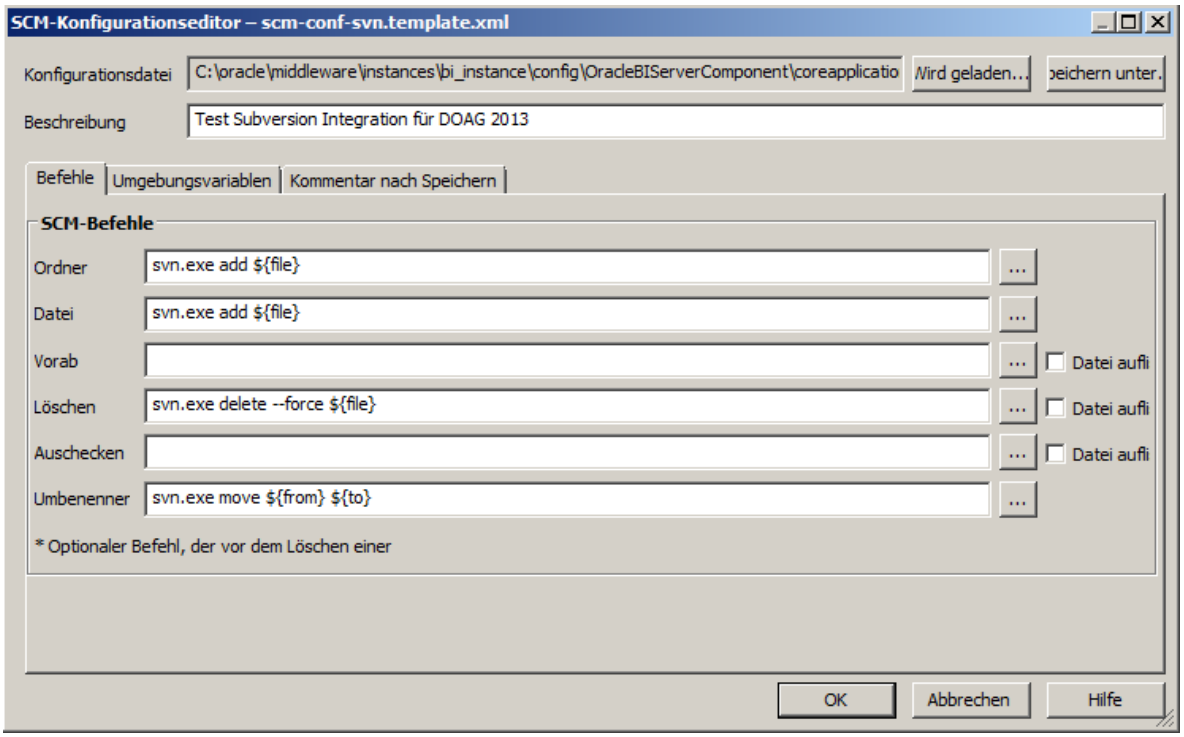

Abb. 5: Konfiguration des Administratorwerkzeugs

Im nächsten Schritt muss ein leeres Verzeichnis z.B. „/trunk/Repository/“ aus Subversion auf dem lokalen PC ausgechecked werden. Dies kann im Falle Subversion mit „TortoiseSVN“ erfolgen. Als nächstes muss das RPD-File als MDS XML-Struktur in exakt dieses Verzeichnis gespeichert werden. Dadurch werden in diesem Verzeichnis neue Verzeichnisse angelegt und darin die zuvor beschriebenen generierten XML-Dateien. Danach wird der „Commit“ für die neu angelegten Verzeichnisse und Dateien über „TortoiseSVN“ durchgeführt. Je nach Größe des OBIEE Repository und Verfügbarkeit des Subversion Servers kann dies einige Zeit dauern (siehe Abbildung 6).

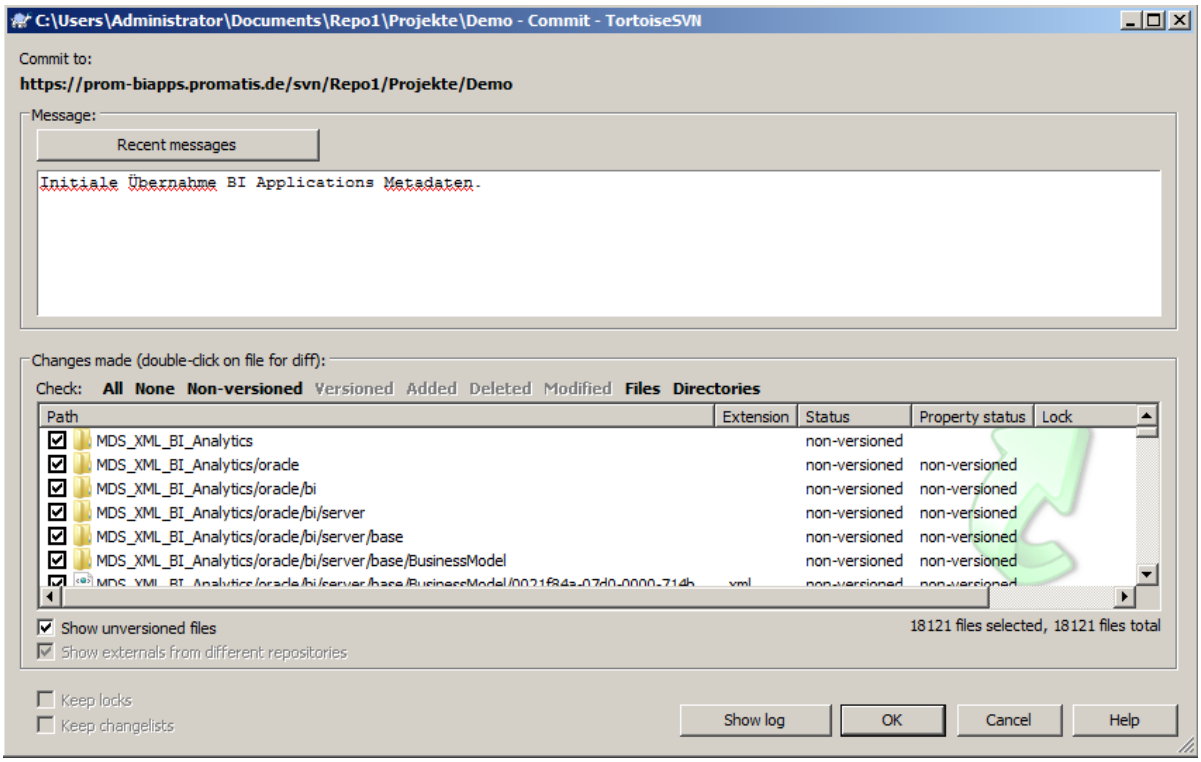

Abb. 6: Initiale Übertragung der Dateien nach Subversion

Alle neuen Dateien können direkt auf dem Subversion Server überprüft werden. Das Repository steht unter Source Code-Verwaltung.

Im nächsten Schritt wird ein Benutzer für das bestehende Verzeichnis einen Check-out durchführen und auf seinem Rechner eine lokale Kopie der Dateien erzeugen. Beim Arbeiten mit Dateien, die über Subversion verwaltet werden, sollte vor jeder Bearbeitung ein Update durchgeführt werden, um sicherzustellen dass die neuste Version verfügbar ist. Danach muss das Verzeichnis mit dem Administrationswerkzeug über Datei --> Öffnen --> MDS XML geöffnet werden. Beim ersten Öffnen wird der Benutzer gefragt, ob das Verzeichnis unter Quellverwaltung steht oder nicht (siehe Abbildung 7).

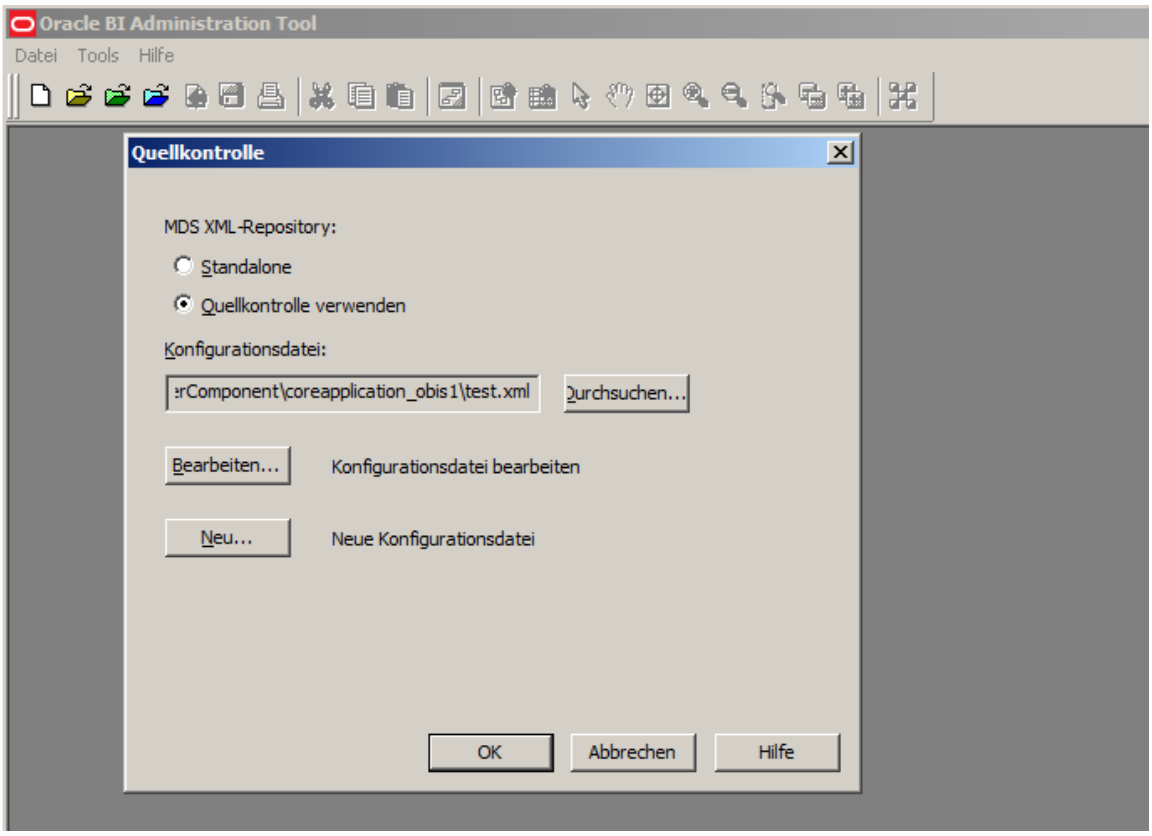

Abb. 7: Öffnen von Metadaten, die unter Quellverwaltung stehen

Im Falle, dass das OBIEE Repository in Subversion verwaltet wird, muss die Konfigurationsdatei angegeben werden, die zu Beginn des Fallbeispiels erzeugt wurde. In der Kopfzeile des Administrationswerkzeugs kann man erkennen, in welchem Modus das OBIEE Repository geladen wurde (siehe Abbildung 8).

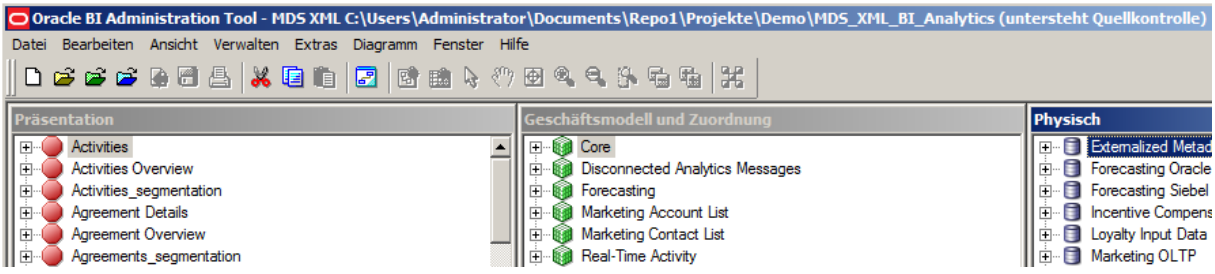

Abb. 8: Administrationswerkzeug mit einem veralteten Subversion OBIEE Repository

Werden im OBI Repository Änderungen vorgenommen so werden alle betroffenen Dateien identifiziert. Beim Speichern werden nach der Konsistenzprüfung und einem Speicherdialog die Aktionen für die Source Code-Verwaltung angezeigt. In diesem Fallbeispiel wurden 2 neue Dateien erzeugt: 1241 Dateien wurden angepasst und mehrere Dateien werden gelöscht (siehe Abbildung 9).

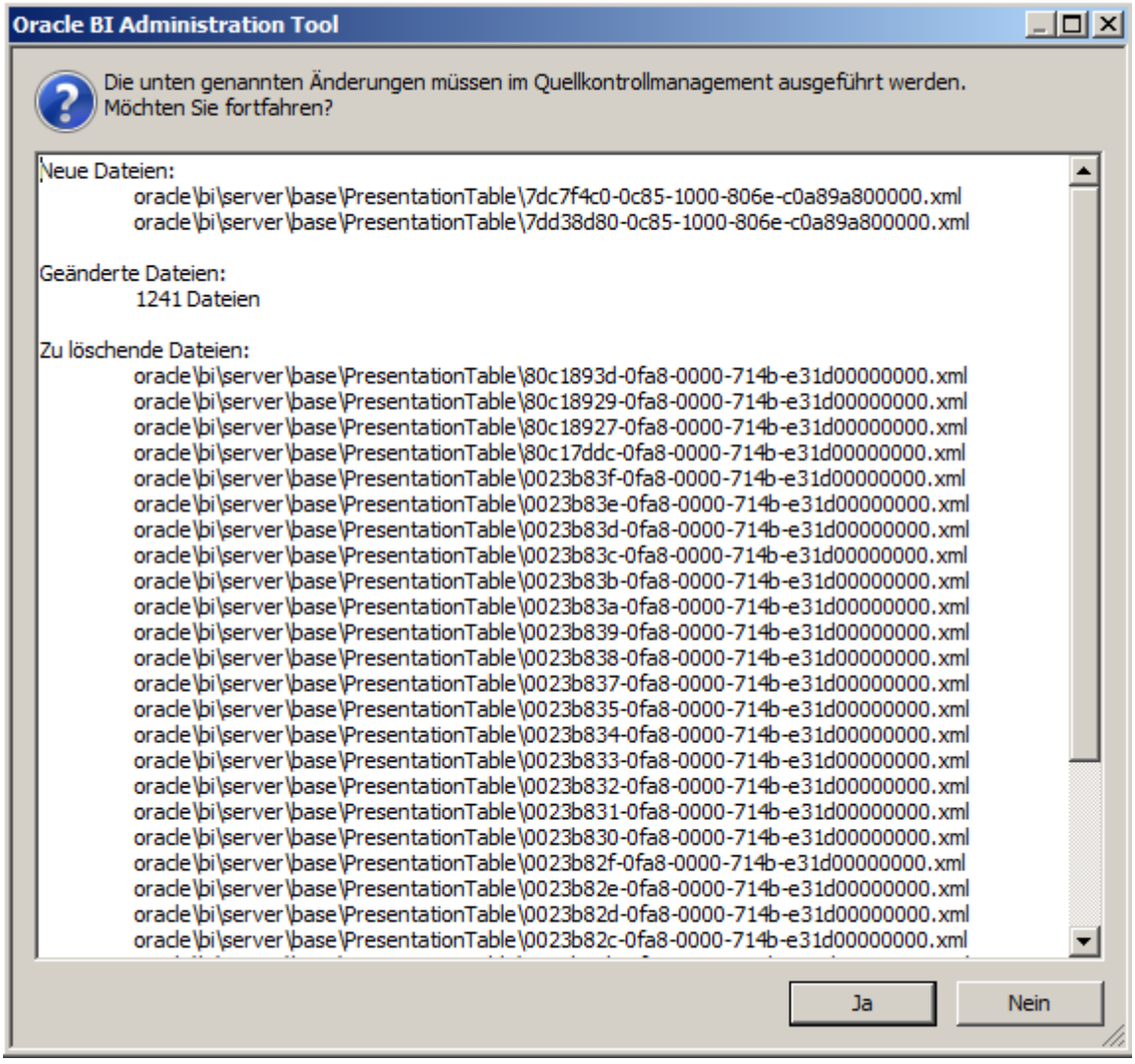

Abb. 9: Subversion Aktionen durch die Änderungen am OBIEE Repository

Die Dateien wurden zwar in Subversion gespeichert, der Commit-Befehl muss allerdings außerhalb des Administrationswerkzeuges manuell durchgeführt werden. Erst nach dem „Commit“ stehen die Änderungen anderen Benutzern zur Verfügung.

Zusammenfassung und Ausblick

Das OBIEE Repository unter Source Code-Kontrolle zu stellen, ist aus Sicht des Autors eine sinnvolle Funktion. Die hier vorgestellte Speicherung des OBIEE Repository mittels MDS XML ermöglicht zumindest eine „halbautomatische“ Integration mit Servern wie Subversion. Für ein professionelles Arbeiten sollte Oracle möglichst rasch den eingeschlagenen Weg weiter verfolgen und ausbauen. Beim Test mit einer sehr großen RDP-Datei, hat das Laden der MDS XML-Dateien sehr lange gedauert. Dies ist nicht akzeptabel. Da das Feature insbesondere bei großen Projekten sinnvoll ist.

Der Merge von mehreren parallel geänderten Dateien ist sehr aufwändig und kann nur in sehr wichtigen Fällen durchgeführt werden. In der Regel ist bei parallelen Änderungen zu entscheiden, welcher Benutzer „gewinnt“. Ein Vorteil des Verfahrens ist, dass Änderungen transparenter werden

als bei der Nutzung von RPD-Dateien. So können Anpassungen an den Metadaten zumindest über die XML-Strukturen eingegrenzt werden, auch wenn dies sehr mühsam ist.

Zusammenfassend muss man sagen, dass ein Änderungsmanagement sehr wohl mit dieser Funktionalität möglich ist. In großen Projekten mit vielen Administratoren und einer großen RPD-Datei kann zumindest derzeit nur das bewährte MUD-Verfahren genutzt werden.

Kontaktadresse:
Michael Weiler
PROMATIS software GmbH
Pforzheimer Str. 160
76275 Ettlingen

Telefon:	+49 (0) 7243-2179 17
Fax:	+49 (0) 7243-2179 99
E-Mail:	michael.weiler@promatis.de
Internet:	www.promatis.de • www.horus.biz • www.prociris.biz