

Elvid Muslim

Dekod telekom d.o.o., T.Marinić 2, 10430 Samobor
+385 98 1746 135
elvid.muslim@dekod.hr
www.dekod.hr

Ivan Trpčić

Dekod telekom d.o.o., T.Marinić 2, 10430 Samobor
+385 98 374 519
ivan.trpcic@dekod.hr

Sandra Skenderović

Dekod telekom d.o.o., T.Marinić 2, 10430 Samobor
+385 98 472 005
sandra.skenderovic@dekod.hr

SAŽETAK

Svima je poznato koje su prednosti za razdvajanjem razvojne, testne i produkcijske okoline za potrebe izrade skladišta podataka.

U timu za razvoj skladišta podataka s takvom konfiguracijom okolina, vrlo je bitno da postoji kvalitetan način odrade VC-a (Version Controla).

Predavanje obrađuje metode kojima se može izvesti VC s OWB-om te njihove međusobne prednosti i nedostatke.

UVOD

Zbog čega nam je potrebno razdvajanje razvojne, testne i produkcijske okoline za potrebe izrade skladišta podataka?

Najjednostavniji odgovor bi bio da se nikad ne bi smjelo raditi izmjene na produkcijskoj okolini ukoliko se prije toga nismo dobro uvjerali da su izmjene ispravno implementirane i testirane. Čak i male izmjene su podložne da postanu bug i uzrokuju ozbiljne probleme.

Razvojna okolina sadrži zadnje verzije objekata. Na ovoj okolini se radi razvoj potrebnih objekata i tek nakon uspješnog razvoja objekti bi se trebali prebacivati na testnu okolinu.

Testna okolina se koristi od strane više tipova korisnika. Uz programere koji testiraju svoje izmjene i dorade ovdje također su u test uključeni i krajnji korisnici koji provjeravaju odgovaraju li napravljene izmjene njihovim poslovnim pravilima i potrebama. Također za testnu okolinu potrebno je da čim više odgovara produkcijskoj okolini (kako po pitanju kvalitete podataka, tako i po pitanju performansi) radi što kvalitetnijeg testiranja.

Prilikom ovakve konfiguracije okolina bitno je da se čim kvalitetnije upravlja promjenama na objektima. Zbog toga nam je potreban čim kvalitetniji Version Control System.

Tema ovog predavanja biti će prikaz nekih od mogućih realizacija VCS u Oracle Warehouse Builderu.

VERSION CONTROL SYSTEM (VCS)

VCS je softver koji omogućuje upravljanje promjenama dokumenata, programa, slika i drugih informacija koje se pohranjuju u obliku računalnih datoteka. Promjene se obično identificiraju prema inkrementalnom broju ili slovu koje je također poznato kao broj revizije.

VCS je postao sastavni dio prilikom razvoja skladišta podataka i od njega se očekuje da može zadovoljiti sljedeće kriterije:

- podrška rada u timu – potrebno je da podržava da tim programera koji može biti geografski odvojen može raditi na istom skupu objekata (datoteka) bez da si međusobno utječu na rad
- upravljanje promjenama – promjene se mogu pregledavati, dograđivati ali i vraćati stare verzije ukoliko se ustanovi da za tim ima potrebe
- praćenje vlasništva – svaka verzija bi trebala biti povezana s imenom programera koji je implementirao navedenu izmjenu
- praćenje promjena – svaku promjenu bi trebalo biti moguće dokumentirati s kratkim opisom izmjena koje su napravljene s obzirom na prijašnju verziju

UPRAVLJANJE METAPODACIMA U ORACLE WAREHOUSE BUILDERU

Oracle Warehouse Builder omogućuje izradu snapshot-a koji služe za izradu sigurnosne kopije metapodataka, praćenje povijesti promjena metapodataka i za usporedbu različitih verzija metapodataka. Snapshot sadrži sve informacije o objektima i njihovim međusobnim relacijama. Dok objekt u workspaceu može imati samo jednu trenutnu definiciju, istodobno može postojati više snapshota koji sadrže njegove promjene kroz određeni vremenski period.

Oracle Warehouse Builder omogućuje dva tipa snapshota koji se mogu kreirati ovisno o funkcionalnosti:

- Full Snapshot – snapshot koji omogućuje kreiranje sigurnosne kopije i njezino vraćanje
- Signature Snapshot – snapshot koji omogućuje povijesne podatke za usporedbu

Snapshotovi se spremaju u bazu, ali postoji i mogućnost da se snapshotovi spremaju na disk u obliku datoteka.

Također s obzirom na spremanje objekata koji pripadaju vršnom objektu za koji se izrađuje snapshot moguće je snapshotove podijeliti u dvije sljedeće kategorije:

- No Cascade Snapshot – u snapshot se sprema isključivo definicija označenoga objekta i u slučaju objekta koji samo sadrži druge objekte kao što je npr. modul ili projekt
- Cascade Snapshot – u snapshot se uključuju i objekti koji spadaju u kontekst objekta za koji se radi snapshot. Ukoliko se sprema snapshot modula koji sadrži dvije tablice tada u ovom slučaju će se u snapshot spremiti modul uključujući i dvije tablice koje mu pripadaju.

Tablica 1 pokazuje koje su mogućnosti snapshota s obzirom na gornje dvije podjele:

Tablica I: Kombinacije snapshotova

TIP SHAPSHOTA	REZULTAT SNAPSHOTA NAD MODULOM "A" KOJI SADRŽI DVIJE TABLICE : TABLICA1 I TABLICA2
Signature, Cascade Snapshot	Snapshot uključuje informacije o modulu A, Tablici 1 i Tablici 2. Snapshot se ne može koristiti za vraćanje svih triju objekata; snapshot se može koristiti jedino za usporedbu.
Full, Cascade Snapshot	Snapshot uključuje modul A, Tablicu1 i Tablicu2. Snapshot se može koristiti za vraćanje objekata ili za usporedbu .
Signature, No Cascade Snapshot	Snapshot uključuje jedino modul A Snapshot se ne može koristiti za vraćanje objekta, može se jedino koristiti za usporedbu. Također ukoliko se neka od tablica promijeni u tom slučaju usporedba ne prikazuje da se modul A promijenio.
Full, No Cascade Snapshot	Snapshot uključuje jedino modula A Snapshot se može koristiti za vraćanje modula A ali ne i njegovih tablica. Ukoliko se neka od tablica promijeni, tada usporedba neće pokazivati da se modul A promijenio.

Upravljanje snapshotovima

Za upravljanje snapshotovima koristi se Metadata Change Manager unutar kojeg je moguće međusobno uspoređivati, brisati, spremati snapshotove u datoteke, učitavati snapshotove iz datoteka...

Da bismo bolje ilustrirali rad metadata change managera poslužit ćemo se primjerom. Na slici 1 dan je popis tablica i njihove veze koje će se koristiti kod punjenja određene tablice za potrebe ovog predavanja. Tablice koje se koriste za punjenje standardno se kreiraju u HR shemi prilikom instalacije Oracle baze podataka.



Slika 1: ER Dijagram

Iz tablica sa slike 1 napuniti će se denormalizirana tablica TGT_EMPLOYEES koja će sadržavati podatke iz gornjih tablica.

U nastavku su popisane tablice koje se koriste za potrebe predavanja i njihov opis odnosno izgled podataka:

Tablica II: Tablica DEPARTMENTS

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700

Tablica III: Tablica REGIONS

<i>REGION_ID</i>	<i>REGION_NAME</i>	<i>MANAGER_ID</i>	<i>LOCATION_ID</i>
1	Europe	200	1700

Tablica IV: Tablica LOCATIONS

<i>LOCATION_ID</i>	<i>STREET_ADDRESS</i>	<i>POSTAL_CODE</i>	<i>CITY</i>	<i>STATE_PROVINCE</i>	<i>COUNTRY_ID</i>
1000	1297 Via Cola di Rie	00989	Roma		IT

Tablica V: Tablica COUNTRIES

<i>COUNTRY_ID</i>	<i>COUNTRY_NAME</i>	<i>REGION_ID</i>
IT	Italy	1

Tablica VI: Tablica EMPLOYEES

<i>EMPLOYEE_ID</i>	<i>FIRST_NAME</i>	<i>LAST_NAME</i>	<i>EMAIL</i>	<i>PHONE_NUMBER</i>	<i>HIRE_DATE</i>	<i>SALARY</i>	<i>DEPARTMENT_ID</i>
165	David	Lee	dlee	4565	10.05.2005	10 000	10

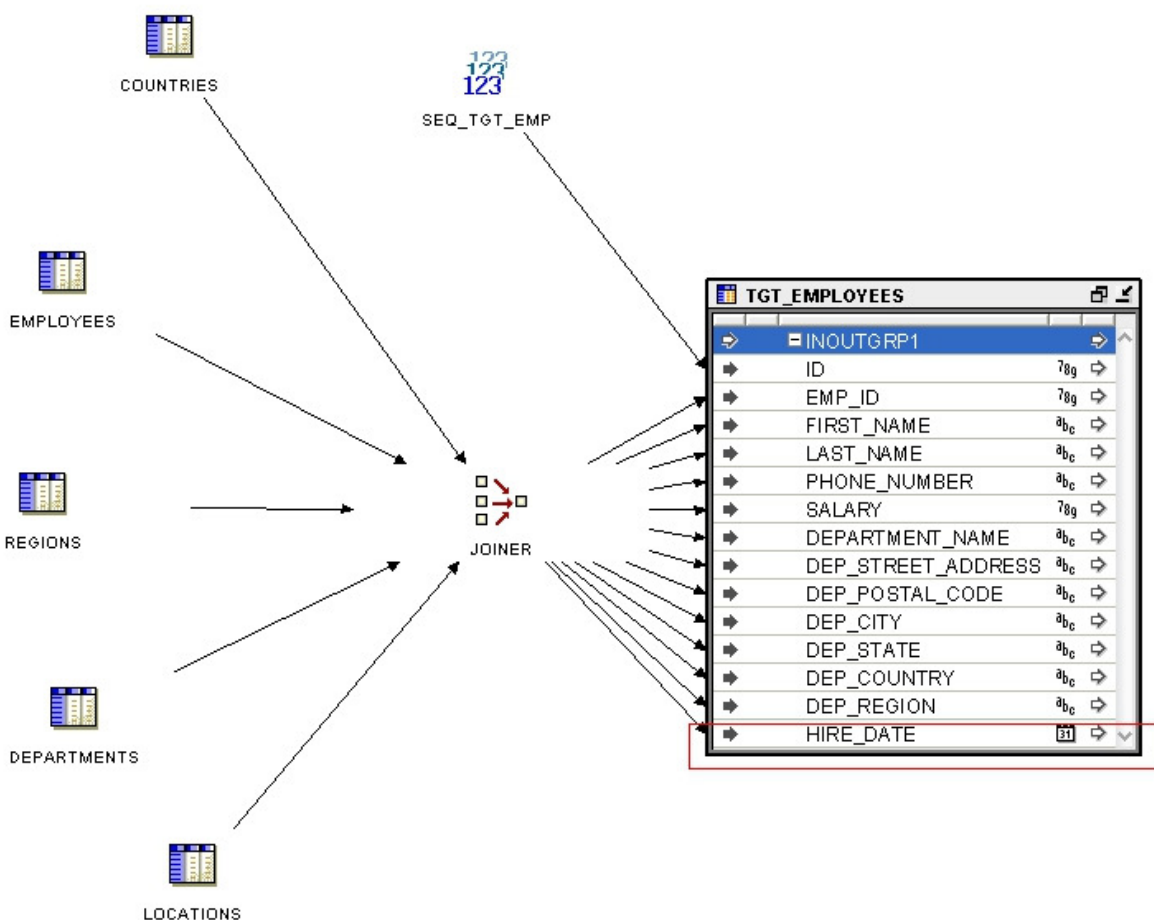
Odredišna tablica TGT_EMPLOYEES

Tablica VII: Tablica TGT_EMPLOYEES

<i>NAZIV KOLONE</i>	<i>OPIS</i>	<i>IZVORNA TABLICA</i>
ID	Primarni ključ tablice	
EMP_ID	ID zaposlenika	EMPLOYEES.EMPLOYEE_ID
FIRST_NAME	Ime zaposlenika	EMPLOYEES.FIRST_NAME
LAST_NAME	Prezime zaposlenika	EMPLOYEES.LAST_NAME
PHONE_NUMBER	Broj telefona	EMPLOYEES.PHONE_NUMBER
SALARY	Plaća	EMPLOYEES.SALARY
DEPARTMENT_NAME	Naziv odjela u kojem zaposlenik radi	DEPARTMENTS.DEPARTMENT_NAME
DEP_STREET_ADDRESS	Adresa gdje se nalazi odjel u kojem zaposlenik radi	LOCATIONS.STREET_ADDRESS
DEP_POSTAL_CODE	Poštanski broj gdje je lociran odjel zaposlenika	LOCATIONS.POSTAL_CODE
DEP_CITY	Grad u kojem je smješten odjel	LOCATIONS.CITY
DEP_STATE	Pokrajina u kojoj je smješten odjel	LOCATIONS.STATE_PROVINCE
DEP_COUNTRY	Zemlja u kojoj se nalazi zaposlenikom	COUNTRY.COUNTRY_NAME

NAZIV KOLONE	OPIS	IZVORNA TABLICA
	odjel	
DEP_REGION	Regija u kojoj se nalazi odjel od zaposlenika	REGIONS.REGION_NAME
HIRE_DATE	Datum kad je zaposlenik primljen u odjel	EMPLOYEES.HIRE_DATE

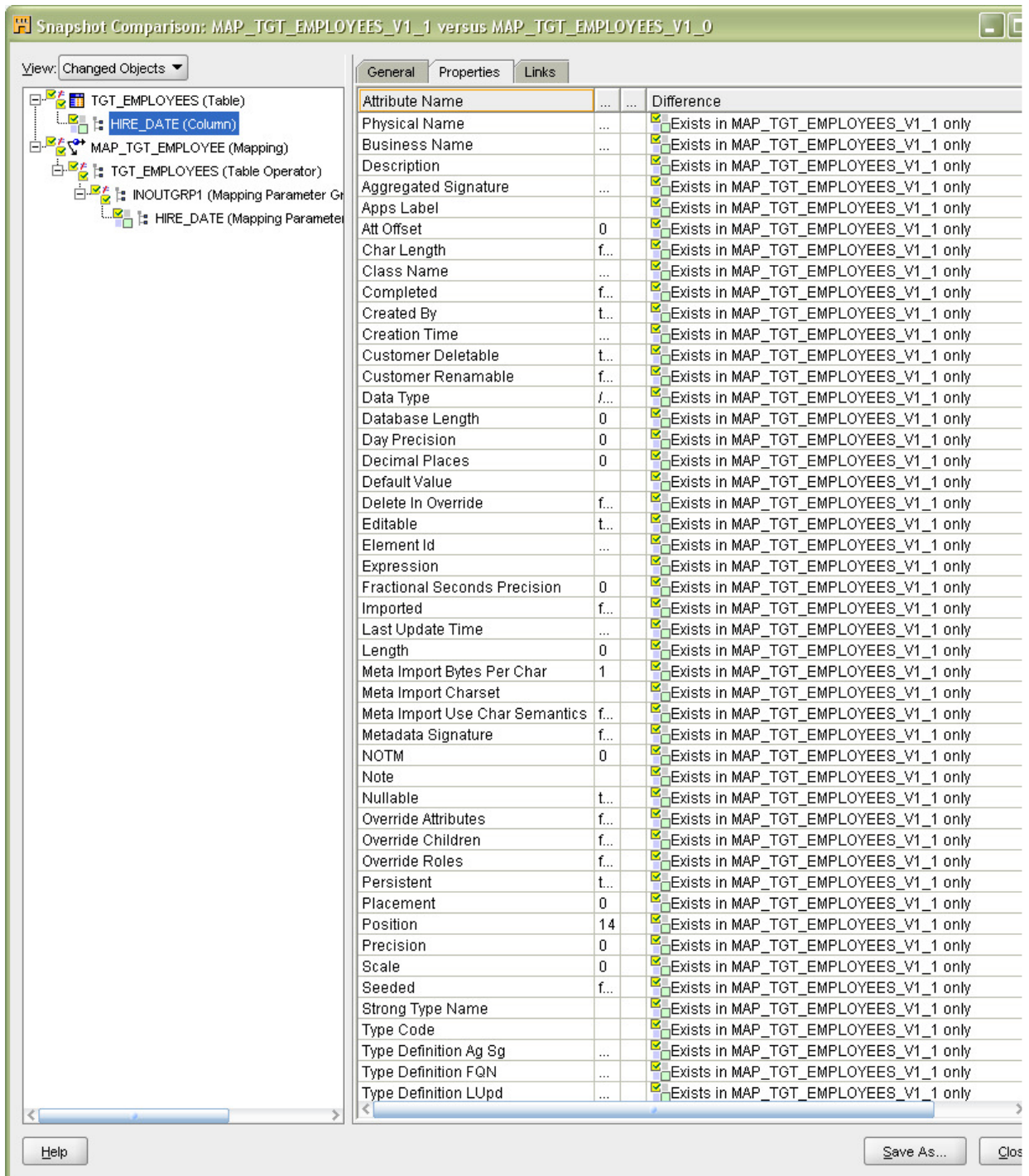
Za potrebe primjera u inicijalnom kreiranju tablice TGT_EMPLOYEES neće postojati kolona HIRE_DATE nego će ona biti dodana u naknadnim izmjenama. Dakle kolona HIRE_DATE će biti dodana između kreiranja dvije verzije snapshota, tako da verzija V1_0 neće sadržavati kolonu HIRE_DATE.



Slika 2: Mapiranje u OWBu

Na slici 2 prikazano je mapiranje koje puni tablicu TGT_EMPLOYEES ukućujući i kolonu HIRE_DATE tako da se ovdje radi o objektu čija verzija odgovara snapshotu MAP_TGT_EMPLOYEES_V1_1. Snapshot MAP_TGT_EMPLOYEES_V1_0 sadrži identičnu logiku jedino što se ne učitava kolona HIRE_DATE.

Ukoliko se preko Metadata Change Managera napravi usporedba između dva snapshota rezultati će pokazati da je jedina razlika u punjenju kolone HIRE_DATE. Spomenuta usporedba prikazana je na slici 3 u nastavku.



Slika 3: Usporedba dva snapshota

Kao što možemo vidjeti unutar Metadata Change Managera međusobna usporedba dva snapshota je primijetila da kolona HIRE_DATE postoji jedino u snapshotu MAP_TGT_EMPLOYEES_v1_1.

UPRAVLJANJE METAPODACIMA KORIŠTENJEM OMBPLUS-a

Sve gore spomenute radnje koje je moguće napraviti korištenjem Metadata Change Managera također je moguće napraviti korištenjem komandi iz OMBPLUS-a. Naravno takvim pristupom moguće je mnogo toga automatizirati.

Kratki primjer koda koji kreira snapshot te novokreirani snapshot uspoređuje s prijašnje kreiranim snapshotom. Razlike između dva snapshota sprema u datoteku diff.xml:

```
OMBCONNECT test_user/test_user@localhost:1521:orcl11gr2;

OMBCC '/MY_PROJECT';

OMBCREATE SHAPSHOT 'MAP_TGT_EMPLOYEES_V1_1_OMB' SET
PROPERTIES(DESCRIPTION, TYPE)
VALUES('Ovo je snapshot tablice i mapiranja TGT_EMPLOYEES kreiran pomocu
OMBPLUSa','FULL')
      ADD TABLE '/MY_PROJECT/TGT/TGT_EMPLOYEES'
      ADD MAPPING '/MY_PROJECT/TGT/MAP_TGT_EMPLOYEES';

OMBCOMMIT;

OMBCOMPARE SNAPSHOT 'MAP_TGT_EMPLOYEES_V1_1' WITH SHAPSHOT
'MAP_TGT_EMPLOYEES_V1_1_OMB'
      OUTPUT TO 'c:/hroug/diff.xml' WRITE CHANGED;

OMBCOMMIT;
```

Nakon što je skripta uspješno završila, datoteka diff.xml ima sljedeći izgled:

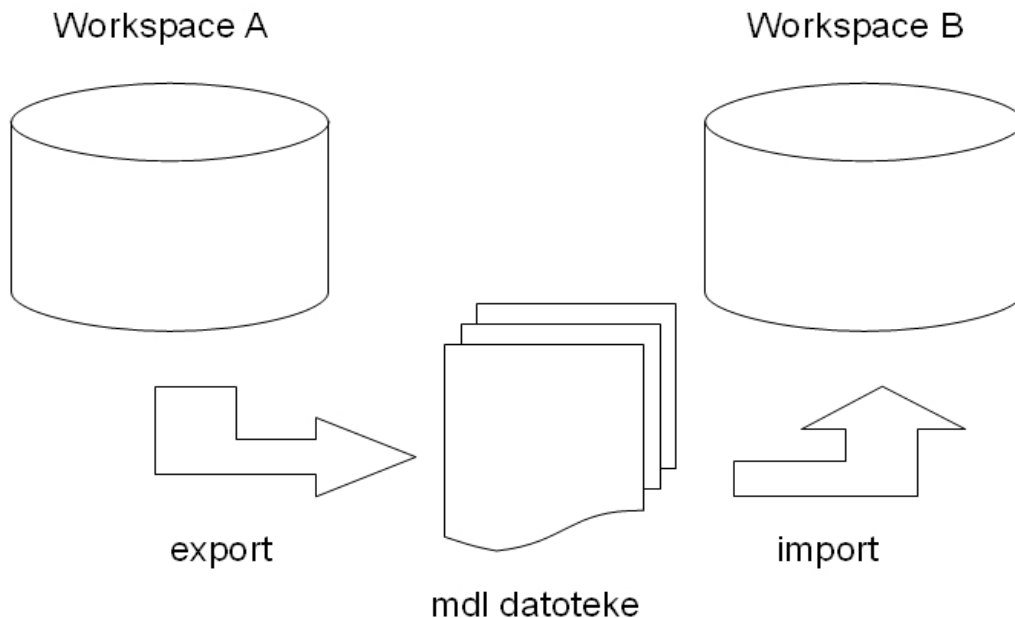
```
<CompareTree TargetSnapshot="MAP_TGT_EMPLOYEES_V1_0"
SourceSnapshot="MAP_TGT_EMPLOYEES_V1_1_OMB"
DiffOf="[TargetSnapshot minus SourceSnapshot]
(MAP_TGT_EMPLOYEES_V1_0 minus MAP_TGT_EMPLOYEES_V1_1_OMB)">
<MetaModelTree TargetSnapshot="MAP_TGT_EMPLOYEES_V1_0"
SourceSnapshot="MAP_TGT_EMPLOYEES_V1_1_OMB"
MetaModelDiffOf="[TargetSnapshot minus SourceSnapshot]
(MAP_TGT_EMPLOYEES_V1_0 minus MAP_TGT_EMPLOYEES_V1_1_OMB)" />
.
.
.
<OWBObject InstanceOf="oracle.owb.mapping.MapAttribute"
PhysicalName="HIRE_DATE" LogicalName="HIRE_DATE"
DiffState="FOUND IN SOURCE ONLY" AggregateState="UNCHANGED"
UUID="09D35F9FB5444F65A33F850921552463"
TypeDescription="Map Attribute"
MyRole="Attributes"
ParentRole="AttributeGroup">
.
.
.
</OWB Object>
</CompareTree>
```

Unutar datoteke diff.xml unutar tag-a OWB Object atribut DiffState je poprimio vrijednost "FOUND IN SOURCE ONLY" dok na vršnom tagu se može pročitati da je atribut SourceSnapshot poprimio vrijednost "MAP_TGT_EMPLOYEES_V1_1_OMB" pa tako se zaključuje da se kolona HIRE_DATE pojavljuje jedino unutar zadnjeg snapshota koji je kreiran pomoću OMBPLUS-a.

VERSION CONTROLING IZVAN ORACLE WAREHOUSE BUILDERA

Osim version controllinga unutar Oracle Warehouse Buildera preko snapshota, moguće je izvesti i version controlling bez korištenja snapshota. U tom slučaju može se koristiti import/export metapodataka pojedinih objekata iz workspacea u OWB-u u mdl datoteke. Kod arhitekture gdje se razvojna, testna i produkcijska okolina nalaze u različitim workspaceovima ovakav način prebacivanja metapodataka između pojedinih okolina je vrlo prikladan.

Na slici 4 prikazan je koncept prebacivanja verzija između dva workspacea.

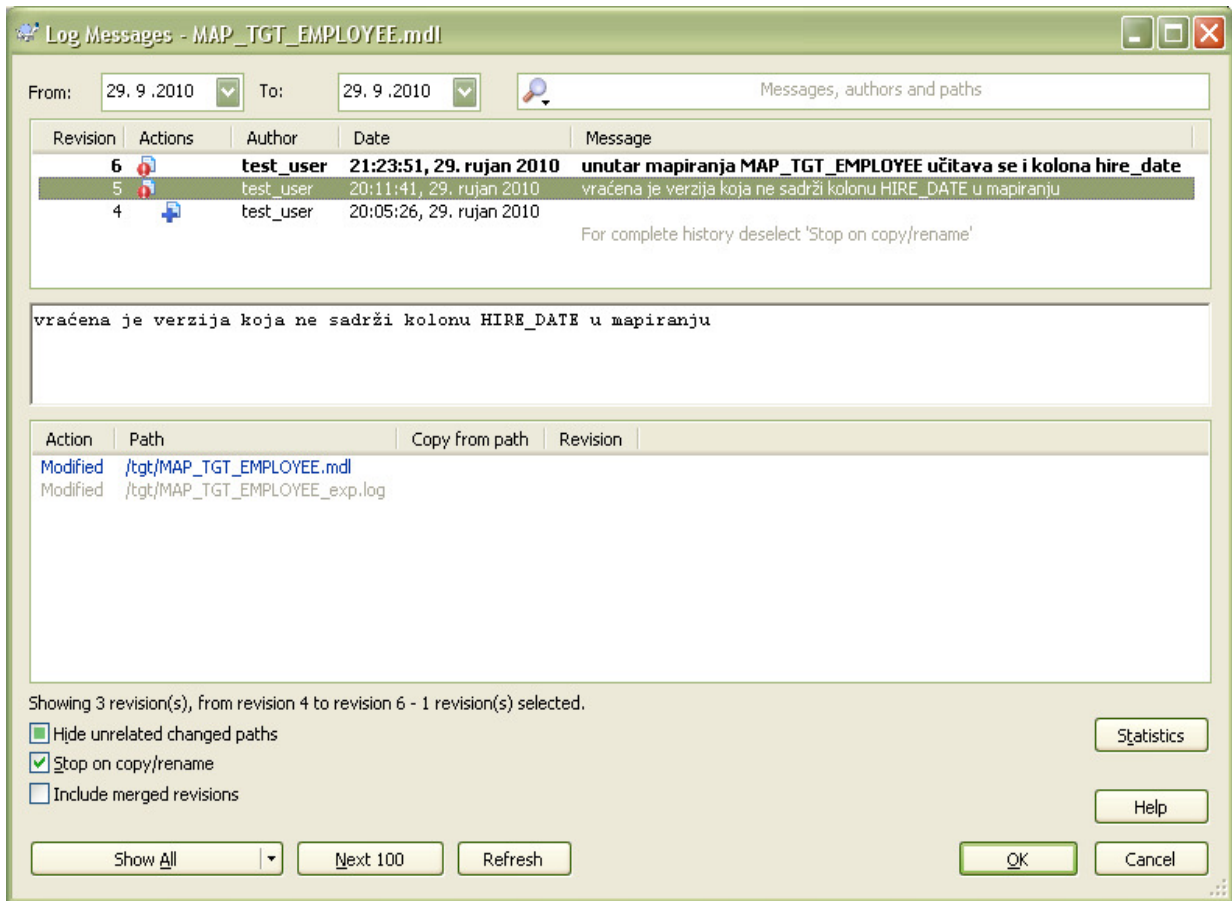


Slika 4: koncept mdl datoteka

Nakon exporta objekata u mdl datoteke nad kreiranim datotekama se može raditi standardni version controlling kao i kod bilo kojih drugih standardnih datoteka. Prilikom ovakvog pristupa poželjno je da je organizacija repozitorija bilo kojeg eksternog VCS usklađena s organizacijom projekata unutar workspacea u samom OWBu. Naravno kod kreiranja mdl datoteka moguće je kreirati datoteke koristeći export wizard iz Design Centra unutra OWBa ali isto tako i koristeći mdl skripte.

```
OMBCONNECT test_user/test_user@localhost:1521:orcl11gr2;  
  
OMBCC '/MY_PROJECT';  
  
OMBEXPORT TO MDL_FILE 'c:/hroug/MAP_TGT_EMPLOYEE.mdl' FROM PROJECT  
'MY_PROJECT' COMPONENTS(MAPPING  
'/MY_PROJECT/TGT/MAP_TGT_EMPLOYEE')  
OUTPUT LOG 'c:/hroug/MAP_TGT_EMPLOYEE.log';  
  
OMBCOMMIT;
```

Ovakvim pristupom moguće je vratiti stari mdl file koristeći eksterni VCS kao na slici 5 te ga zatim importati u repozitorij i na taj način vratiti ispravnu verziju metapodataka unutar workspacea u OWBu.



Slika 5: Prikaz revizija mdl datoteke u eksternom CVSu

Međutim ovdje postoji jedan mali problem. Prilikom kreiranja mdl datoteke koja sadrži metapodatke u samu datoteku se upisuje vrijeme kreiranja mdl datoteke tako da dvije mdl datoteke koje su kreirane nad identičnim objektima u workspaceu će imati različite mdl datoteke kao u primjeru ispod:

```
<?xml version="1.0" encoding="WINDOWS-1250" ?>
<!-- $Header: whdev/2.0/owb/oracle/wh/repos/pdl/metadataloader/export/MDLExportAdmin.java
/main/19 2009/05/21 21:54:08 xxzhang Exp $ -->
<OWBMETADATA Description="Oracle Warehouse Builder Export Data File"
CreationDate="09/12/2010 20:58:32 CEST">
.
.
.
</OWBMETADATA>

<?xml version="1.0" encoding="WINDOWS-1250" ?>
<!-- $Header: whdev/2.0/owb/oracle/wh/repos/pdl/metadataloader/export/MDLExportAdmin.java
/main/19 2009/05/21 21:54:08 xxzhang Exp $ -->
<OWBMETADATA Description="Oracle Warehouse Builder Export Data File"
CreationDate="09/14/2010 18:18:32 CEST">
.
.
.
</OWBMETADATA>
```

Zbog ovakvog načina zapisivanja metapodataka eksterni VCS će detektirati razlike između dvije spomenute mdl datoteke i time uočiti razlike između lokalnog repozitorija i zajedničkog repozitorija iako između objekata čiji su podaci u mdl datoteci nema nikakvih razlika.

ZAKLJUČAK

Kao što je u ovom predavanju pokazano version control u Oracle Warehouse Builderu se može izvesti na nekoliko načina od koji neki nude i mogućnost integracije s eksternim version control sistemima. Uz mogućnosti integracije s eksternim alatima ovakav način version controla nudi i veliku robusnost čija je primjena moguća na razne arhitekture.

Version control u OWB-u zadovoljava potrebe koje se postavljaju pred bilo koji drugi samostalni VCS. Korištenje snapshota za realizaciju je ipak u prednosti zbog gubitaka određenih funkcionalnosti prelaskom na eksterne VCS. Korištenjem skripti u OMB Plusu stvari se mogu dodatno automatizirati i time poboljšati način izvedbe VC.