„Managed Code“ mit OWB? Methoden und Wege

Bernhard Rosenberger
MT AG
Frankfurt

Schlüsselworte
Oracle, OWB, Oracle Warehouse Builder, Tcl, OMB, OMB*Plus, Versionsverwaltung, Sourcecode Management, Subversion, CVS

Einleitung
Größere Entwicklungsprojekte mit dem OWB stellen Arbeitsgruppen vor eine organisatorische Herausforderung. Das Tool bietet keine voll ausgereifte Versionskontrolle für die entwickelten OWB Objekte. Die Dokumentation der Entwicklungshistorie ist jedoch eine wichtige Aufgabe um hohe Qualität im Development Lifecycle eines DWH Systems sicherzustellen.

OWB Managed Code
Die Entwicklung und das Deployment von Mappings und Workflows, so möchte man meinen, ließe sich mit dem OWB ‚straight forward‘ betreiben. Anforderungen aufnehmen – Lösungskonzept erarbeiten – Mappings implementieren – deployen – testen – iterative Fehlerbehebung – Deployment in der Produktion – Fertig!
Das mag in kleineren Umgebungen mit einem oder evtl. auch zwei Mitarbeitern auch weitgehend zutreffen. Aber auch nur, wenn es den Entwicklern gelingt parallele Anforderungen der Stakeholder eines DWH zu serialisieren und so eine linear fortschreitende Entwicklung zu garantieren.

Wie geht man damit um, wenn in der Produktion ein gravierender Fehler auftritt, der sofort behoben werden muss? Erschwerend käme dann noch dazu, dass die betreffenden Mappings und Workflows bereits weiterentwickelt worden sind. Es gilt zu verhindern, dass die Entwicklungsstände auseinanderlaufen.

Sofern man nun über keine zweite Entwicklungsumgebung als Werkbank verfügt, muss der Entwicklungsstand gesichert und der Produktionsstand importiert werden, um dringend erforderliche Reparaturen durchführen zu können. Sind die Arbeiten abgeschlossen und die Korrekturen auf die Produktion übertragen, so ist es mit einem Reimport des Entwicklungsstandes nicht getan. Jetzt müssen alle durchgeführten Änderungen in den in Entwicklung befindlichen Sourcen nachgezogen werden. In der klassischen Softwareentwicklung bezeichnet man diesen Vorgang als Merge.

In größeren Organisationen ist es nicht unüblich, dass mehrere solcher Anforderungen gleichzeitig an ein Entwicklungsteam gestellt werden, dann ergeben sich daraus erhebliche Mehraufwände was die Koordination der Implementierungsarbeiten einzelner Aufgaben angeht. In Abbildung1 ein Beispiel, das die mögliche Komplexität bei der Releasekoordination veranschaulichen soll.

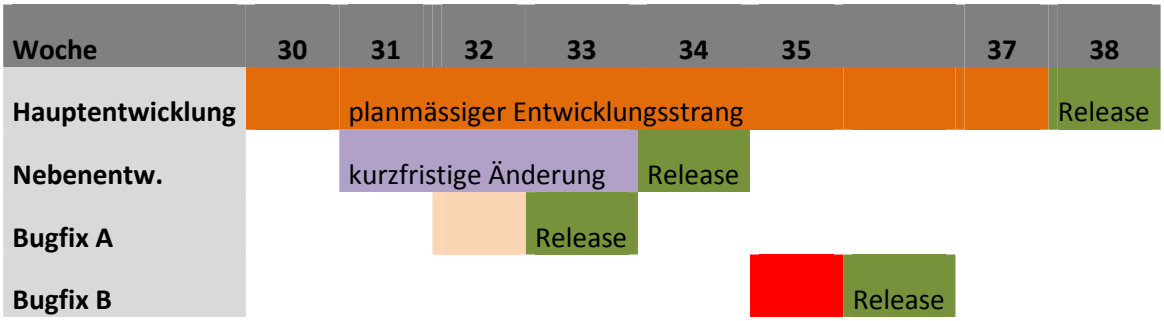

Abb. 1: Herausforderung Parallelentwicklung im OWB

Parallelentwicklung mit OWB?

Der OWB erlaubt lediglich die Entwicklung einer Version eines Objekts zur gleichen Zeit. Um unabhängig voneinander entwickeln zu können, sind vorzugsweise mehrere Entwicklungsumgebungen zu verwenden. Es sind dann zusätzlich die unterschiedlichen Entwicklungsstände über Merge Aktivitäten zu koordinieren.

Ist nur eine OWB Installation als Entwicklungswerkbank zur Verfügung, dann müssen die erforderlichen Arbeiten im Team zeitlich genau abgestimmt werden. Es ist sicherzustellen, dass jedes OWB first class object (Mappings, table operators, view operator etc.) immer nur für einen abgestimmten Zeitabschnitt von einem der Entwicklungsstränge (Branches) exklusiv zur Bearbeitung reserviert ist.

Durch eine Serialisierung der Arbeiten ähnlich des Multitaskings bei einem Single Prozessor System ist eine Bewältigung der Parallelentwicklung machbar. Die manuelle Bearbeitung eines oder mehrerer first class objects in einem Entwicklungsschritt ist hier wie ein Prozessortask zu sehen. Das Bereitstellen der jeweils erforderlichen Entwicklungsstände in den OWB (MDL Import) und das Zurücksichern des fertigen Entwicklungsergebnisses (MDL Export) müssen vor und nach jedem Entwicklungsschritt erfolgen um den Bearbeitungskontext sicherzustellen (Abbildung 2).

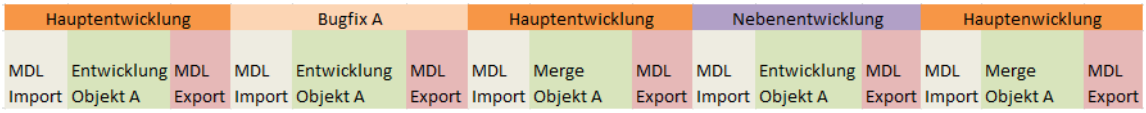

Abb. 2: Multitasking im OWB

Die zeitliche Koordination der Entwicklungsschritte/Tasks ist durch das Team einzuplanen und die Bereitstellung der passenden MDL-Sourcen aus dem jeweiligen Entwicklungsbranch sicherzustellen. Hierbei ist hohe Sorgfalt erforderlich, um Fehler zu vermeiden die möglicherweise erst in der Testphase entdeckt werden können z.B. wenn versehentlich ein Mapping eines falschen Branches importiert, bearbeitet und wieder weggespeichert worden ist. Mit den Planungsaufgaben ist oft eine manchmal auch mehrere Personen in der Rolle als „Release-Manager“ betraut.

Versionierungssysteme

Was liegt also näher, als hier Ausschau nach geeigneten Tools und Hilfsmitteln zu halten, die dem Entwicklungsteam eine systematische Vorgehensweise ermöglichen. Seit langem finden Versionsverwaltungssysteme (z.B. SVN, CVS oder ähnliche) Einsatz in der klassischen

Softwareentwicklung, die für die zuvor beschriebenen Branch und Merge Anforderungen extra geschaffen worden sind.

Kernleistungsmerkmale von Versionsverwaltungssystemen sind:

Commit	Erzeugt aus einer Arbeitskopie eine gültige Version einer Datei
Diff	Vergleicht zwei Dateien miteinander und markiert die Unterschiede: was wurde hinzugefügt, geändert oder gelöscht
Label, Tag	Kennzeichnet eine Menge von Dateien als zu einem Zeitpunkt gültige Version
Branch	Kopie eines gemeinsamen Dokumentenstamms zu einem bestimmten Zeitpunkt; Erlaubt eine unabhängige Weiterentwicklung
Merge	Zusammenführung von Dateien eines Branches zum gemeinsamen Dokumentenstamm, die unabhängig weiterentwickelt worden sind
File Locking	Erlaubt das Sperren von Dateien zum Zwecke Ihrer exklusiven Bearbeitung; Verhindert Merge Probleme

Wie kann man einen solchen Entwicklungskomfort für OWB Objekte (Tables, Mappings, etc.) erreichen?

Collections und Snapshots

Die Antwort der Firma Oracle auf die oben beschriebenen Herausforderungen sind Collections und Snapshots.

Mit Collections werden Pakete von first class objects definiert, jedoch können die Collectionelemente im OWB nicht historisiert werden, weil es sich lediglich um Referenzen auf den aktuellen Stand handelt.

Was das Festhalten von Versionsständen angeht, erlaubt der OWB Change Manager das Erstellen von Snapshots eines Releasestandes, der sich auch später mit einem Neueren Stand (Snapshot) vergleichen lässt (Diff). Mit dem Change Manager lässt sich ein Snapshot in eine andere Datenbank übertragen und auch wiederherstellen. Für eine Gewährleistung lückenloser Versionierung jedoch empfiehlt die Oracle Dokumentation den Snapshot Export und ein Einchecken in ein Versionierungstool.

Bis zur OWB Version 10gR2 haben schlechte Performance und einige Bugs/Features des Change Managers dafür gesorgt, dass sich viele DWH Entwicklungsteams gegen einen Einsatz von Snapshots entschieden haben.

Jenseits der Oracle Option seien im folgenden zwei alternative Verfahren zum Versions- und Releasemanagement vorgestellt, die eine strukturierte Arbeitsweise unterstützen und so dem Entwicklungsteam bei Parallelentwicklungen das Leben erleichtern.

A: Generierung der OWB Objekte mit Hilfe von Tools und OMB-Tcl Skripts sowie Möglichkeiten des Reverse Engineering bestehender Mappings

Dieses Verfahren baut darauf, alle OWB Objekte wie Mappings und Workflows und die dazugehörigen Elemente mittels Tcl Skript zu generieren.

Der OWB bietet mit der OMB-API eine Programmierschnittstelle, die eine skriptgesteuerte Erstellung von OWB Objekten erlaubt. Die OMB-API ist ein Paket aus Java Funktionalitäten, die in die Skriptsprache Tcl integriert ist. Eine im OWB integrierte als auch eine extern startbare Tcl-Shell steht hierfür zur Verfügung (OMB*plus siehe auch Abbildung 3).

Damit eröffnet sich die Möglichkeit Aufbau und Konfiguration von OWB Projekten mittels Skripting durchzuführen. Der Skripting Code lässt sich optimal mit Sourcecodeversionierungssystemen wie CVS oder SVN verwalten. Da der Tcl Code als ASCII basierter Text vorliegt, steht der Nutzung aller Funktionalitäten der Versionierungssysteme offen. So sind z.B. mit Hilfe der Diff Funktionalität Unterschiede zwischen zwei Versionsständen gut dokumentiert. Für den Merge zweier Branches wiederum stehen in den Versionsverwaltungssystemen entsprechende Automatismen unterstützend zur Verfügung.

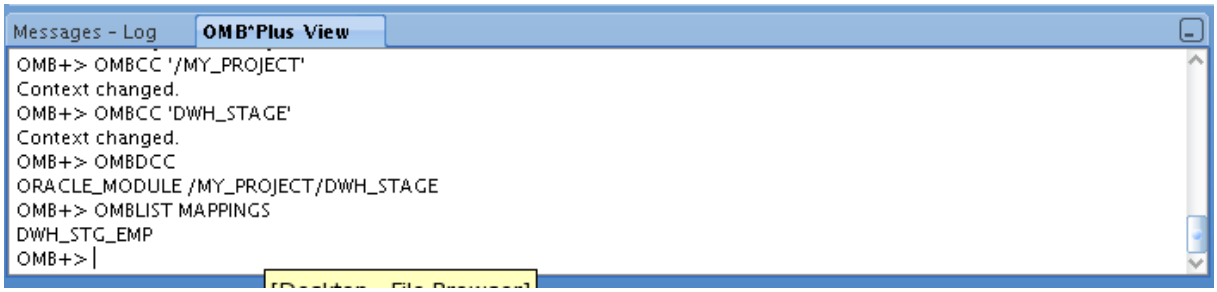

*Abb. 3: OMB*Plus Shell eingebettet im OWB*

Ein Neuaufbau eines DWH Systems, sozusagen auf der grünen Wiese ist hiermit komfortabel realisierbar (full build). Bei bestehenden Systemem müssen die OWB Objekte mittels Reengineering Methoden zuerst einmal in Tcl Sourcecode umgewandelt werden, wenn man bei der Weiterentwicklung von den Diff Funktionalitäten profitieren möchte.

Eine einfachere Variante besteht in der inkrementellen Weiterentwicklung eines bestehenden Systems (inkrementeller build). Man verliert dabei die Möglichkeit einen Diff des Skriptcodes gegenüber der vorhergehenden Version durchführen zu können.

Weitere Vor- und Nachteile beider Verfahren werden im Vortrag dargestellt und sind vor einem Einsatz gut abzuwägen.

B: Entwicklung im OWB und Sourcecodemanagement von MDL Files mit Hilfe von SVN

Das Funktionsprinzip ist hier die Speicherung von MDL Exports aller first class OWB Objekte in einem Versionierungssystem. Parallele Entwicklungsstränge werden als Branch geführt. Auf diese Weise sind alle Versionsstände der Sourcen strukturiert für alle Teammitglieder gleichermaßen im Zugriff. Die Markierung zusammengehöriger Dateien für ein Release mittels eines sogenannten Labels (oder auch Tag genannt) und die Vergabe von Kommentaren beim Einchecken der MDL-Exports erleichtern die Suche nach den jeweils notwendigen Versionsständen.

Sehr hilfreich ist die Verwendung von Sperren (Locks) wie es z.B. SVN bietet. Wenn eine Sperre auf eine ausgecheckte Datei gesetzt ist, sorgt SVN dafür, dass diese Datei von keinem weiteren Anwender des Versionsverwaltungssystems bearbeitet werden kann. Auf diese Art und Weise können Konflikte durch das gleichzeitige Bearbeiten der gleichen Datei elegant vermieden werden.

Eine manuelle Handhabung erfordert vom einzelnen Entwickler die routinemäßige Beachtung einiger Arbeitsschritte:

- Manueller MDL Export jedes veränderten first class objects in ein Sandboxverzeichnis des Versionsverwaltungssystems
- Einchecken des MDL Exports in die Versionsverwaltung
- Vergabe eines Kommentars beim einchecken
- Manuelle Aktualisierung des Versionierungssystems
- Ausschecken eines zu bearbeitenden first class objects, Aufhebung des Schreibschutzes und Markierung für exklusiven Zugriff
- Manueller Import eines first class objects in den OWB

Eine Integration dieser Funktionalitäten in den OWB ist also wünschenswert, um die erforderlichen administrativen Arbeiten zur Versionsverwaltung soweit wie möglich zu vereinfachen.

Experts zur Integration der Versionsverwaltung

Der Einsatz von Experts im OWB ermöglicht eine weitgehende Automatisierung der notwendigen Arbeitsschritte. Experts lassen sich einfach im Kontextmenü des OWB einbinden und ermöglichen so eine bequeme Handhabung der Versionsverwaltung.

Folgende Expert Funktionalitäten sind hier wünschenswert:

Add	Hinzufügen eines neuen Objekts im SVN Repository und initialer Commit
Checkin	MDL Export des markierten Objekts; SVN Commit mit Kommentareingabe; Schreibschutz in lokaler Sandbox wird gesetzt
Checkout	Update des markierten Objekts aus SVN in die lokale Sandbox; Import des MDL Files in den OWB; SVN Locking auf aktuellen Benutzer setzen, dabei wird Schreibschutz der lokalen Datei aufgehoben; Ist im SVN Repository bereits eine Sperre vorhanden (Lock), dann bricht der Vorgang mit Fehlermeldung ab
Delete	Löschen eines OWB Objekts aus dem OWB und aus dem SVN Repository
Tagging	Setzen eines Tags für einen aktuellen Releasestand basierend auf eine Collection von OWB Objekten

Eine Integration von Funktionalitäten zur Versionskontrolle im Kontextmenü des OWB ist in Abbildung 4 dargestellt.

Darüberhinaus erforderliche administrative Aufgaben wie z.B. Aufheben eines Locks eines erkrankten Mitarbeiters oder das Zusammenstellen eines Releases können weiterhin durch Bordmittel des jeweiligen Versionsverwaltungstools durchgeführt werden. Eine Implementierung als zusätzliche Experts ist in Planung.

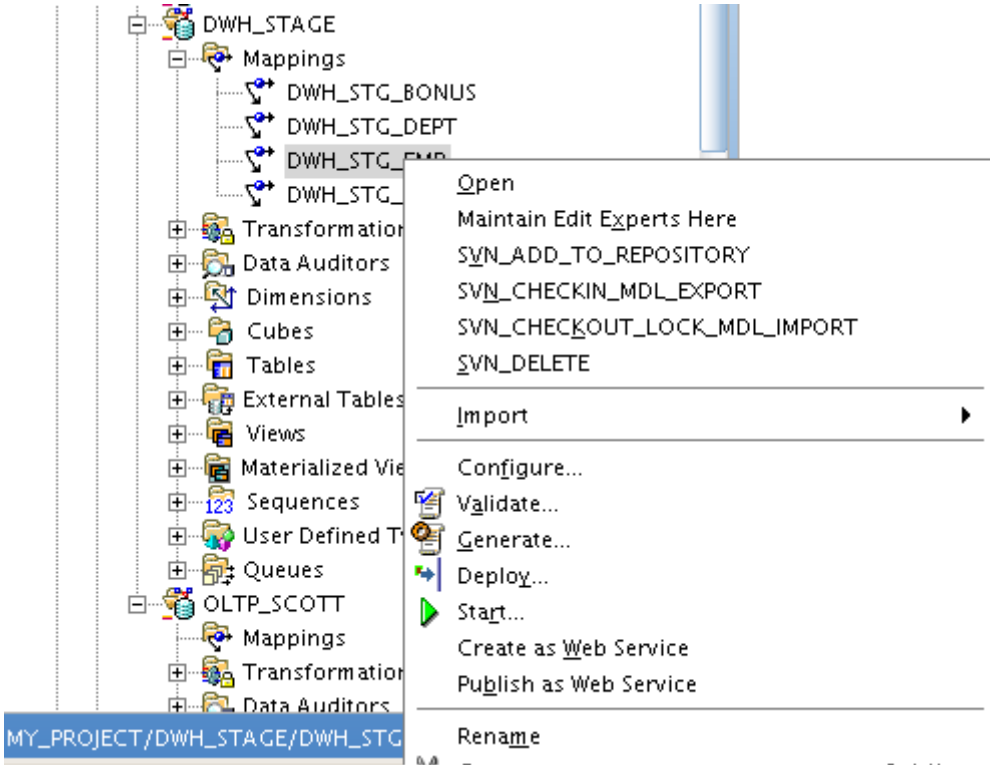

Abb. 4: Versionsverwaltung mit Experts

Zusammenfassung

- Verwendung externer Versionsverwaltungstools ermöglichen eine lückenlose Dokumentation des Implementierungsprozesses
- Parallelentwicklungen mit dem OWB sind möglich, sollten jedoch durch zusätzliche Hilfe einer Versionsverwaltung unterstützt werden
- Versionsverwaltung für die OWB Entwicklung ist mit externen Tools realisierbar. Ein Framework aus Tcl-Skripts und OWB Experts erleichtert die Handhabung
- Die beschriebenen Verfahren spiegeln unsere Erfahrungen aus mehreren Kundenprojekten wieder

Kontaktadresse:
Bernhard Rosenberger
MT AG
Solmsstr. 10
D-60486 Frankfurt

Telefon: +49 69 2649243-20
Fax: +49 2102 309 61-10
E-Mail bernhard.rosenberger@mt-ag.com
Internet: www.mt-ag.com