

Wiederverwendung von bestehendem PL/SQL Code in ADF Anwendungen

Hendrik Gossens

MT AG

D-40882 Ratingen

Schlüsselworte

PL/SQL, ADF, Geschäftslogik, Wiederverwendung

Einleitung

Möchte man eine bestehende Oracle Forms-Anwendung modernisieren und durch eine ADF basierte Anwendung ablösen, so kann die in PL/SQL implementierte Business Logik wiederverwendet werden. Obwohl im Sinne der Schichtenarchitektur eine datenbankzentrierte Geschäftslogik bei der Neuentwicklung von Anwendungen mit ADF ein eher untypisches Szenario darstellen sollte, birgt dieser Ansatz bei der Migration bestehender Anwendungen viele Vorteile. Zum einen kann auf bewährte und getestet Geschäftslogik zurückgegriffen werden, zum anderen verringert sich durch den Wiederverwendungsaspekt die Entwicklungszeit der Anwendung. Darüber hinaus werden der Parallelbetrieb der alten und der neuen Anwendung, sowie eine schrittweise Migration in die "neue Welt" erheblich vereinfacht. ADF-Projekte bieten verschiedene Ansatzpunkte zur Einbindung von PL/SQL Code. Dieses Manuskript stellt mögliche Varianten.

Datenhaltung, Geschäftslogik und Darstellung - Schichtentrennung moderner Java EE-Anwendungen

Im Gegensatz zu alten Oracle Forms-Anwendungen folgen moderne Java EE-Anwendungen dem Paradigma der strikten Trennung von Datenhaltung, Geschäftslogik und Darstellung.

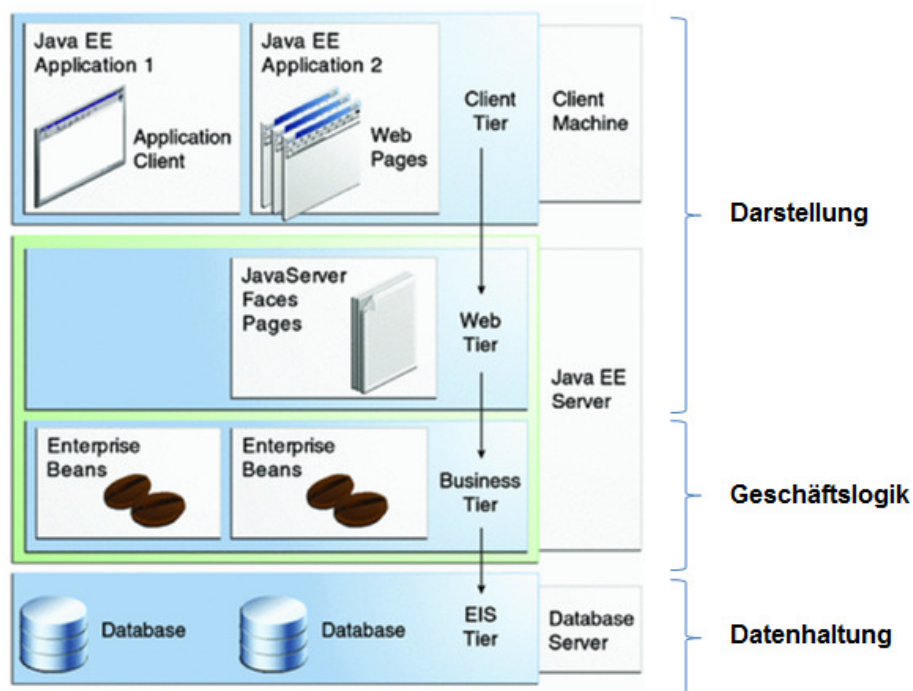


Abb. 1: Java EE Schichtenarchitektur – Trennung von Datenhaltung, Anwendung und Darstellung

Durch die Mehrschichtigkeit ist ein hohes Maß an Modularität gegeben. Dies birgt den Vorteil, dass gegebenenfalls einzelne Implementierungen und Komponenten der Schichten ausgetauscht werden können, ohne eine Re-Implementierung der gesamten Anwendung vornehmen zu müssen. Abbildung 1 zeigt die typische Mehrschichtenarchitektur von Java EE-Anwendungen. Im Gegensatz dazu sind in Oracle Forms-Applikationen Geschäftslogik (in Form von PL/SQL-Code) und Datenhaltung zu einer Schicht zusammengefasst.

Bei ADF handelt es sich um ein Java EE-Framework. Somit besteht zwischen ADF-Anwendungen und Forms-Anwendungen die beschriebene Paradigmen-Divergenz im Hinblick auf die Implementierung von Geschäftslogik: Während die Geschäftslogik von Forms-Anwendungen in Form von PL/SQL-Code in der Datenbank hinterlegt wird, stellt die Geschäftslogik in ADF-Anwendungen eine eigene, von der Datenbank entkoppelte, Schicht dar.

Obgleich es mehrere Möglichkeiten gibt ADF-seitig PL/SQL-Code zu verwenden, soll hier explizit darauf hingewiesen werden, dass die in diesem Manuskript vorgestellten Ansätze nicht für neu zu implementierende Geschäftslogik verwendet werden sollten. Muss Geschäftslogik einer ADF-Anwendung neu implementiert werden, so sollte die Neuimplementierung der im Java EE-Umfeld typischen Trennung von Datenhaltung und Geschäftslogik folgen.

Im Gegensatz dazu kann die Verwendung bestehender PL/SQL-Geschäftslogik sinnvoll sein, wenn „in die Jahre gekommene“, aber bewährte Forms-Anwendungen modernisiert werden sollen. In diesem Fall bietet das Übergehen der strikten Schichtentrennung vor allem folgende Vorteile:

- Kein Aufwand für die Neuentwicklung schon implementierter Geschäftslogik
- Zurückgreifen auf bewährten und getesteten Code
- Paralleler Betrieb von Forms-Anwendungen und neuer ADF-Anwendung möglich
- Schrittweise Migration der bestehenden Anwendung Funktionalität in die ADF-Welt ohne Einbußen der bestehenden Funktionalität möglich.

Um die Möglichkeit einer eventuell zu einem späteren Zeitpunkt gewünschten Transformation der datenbankseitigen Geschäftslogik in eine eigene Schicht offen zu halten, sollte bei der Einbindung von PL/SQL Code in ADF-Anwendungen eine Kapselung des Aufrufs erfolgen. Hierdurch wird prinzipiell eine virtuelle Geschäftslogik-Schicht erzeugt, die dann zukünftig durch Neuimplementierungen ausgetauscht werden kann.

Ansätze zur Verwendung in PL/SQL implementierter Geschäftslogik

Berücksichtigt man den empfohlenen Weg der Kapselung der Integration des PL/SQL Codes in ADF-Anwendungen, so lassen sich verschieden Ansätze identifizieren, wie die bestehende Geschäftslogik in ADF-Anwendungen weiterleben kann:

1. Kapselung des Datenzugriffs über Oracle Business Components

Für diesen Ansatz existieren zwei Möglichkeiten:

- a. Aufruf des PL/SQL-Codes über die `doDML()`-Methode der Entity Object-Implementierungsklasse:

```
static final int attributes[] =
    new int[] { EVENTNO, ORGID, DESCRIPTION, CONTACTNAME,
                STARTDATE, ENDDATE, COMMENTS };
static final int pkAttributes[] = new int[] { EVENTNO };

static final String rowType = "CG$EVENTS_CG$ROW_TYPE";
```

```

static final String indType = "CG$EVENTS_CG$IND_TYPE";
static final String pkType  = "CG$EVENTS_CG$PK_TYPE";

protected void doDML(int operation, TransactionEvent e) {

    try {
        if (operation == DML_INSERT) {
            super.insertUpdate(attributes, rowType, indType,
                "BEGIN events_pkg.ins(?, ?, TRUE); END;");
        } else if (operation == DML_UPDATE) {
            super.insertUpdate(attributes, rowType, indType,
                "BEGIN events_pkg.upd(?, ?, TRUE); END;");
        } else if (operation == DML_DELETE) {
            super.delete(pkAttributes, pkType,
                "BEGIN events_pkg.del(?, ?, TRUE); END;");
        } else // unlikely but you never know
            super.doDML(operation, e);
    } catch (SQLException ex) {
        throw new JboException(ex);
    }
}

```

(Quelle: <http://www.oracle.com/technetwork/articles/muir-designer-085584.html>)

- b. Eigene Methoden in der Implementierungsklasse des Application Module schreiben (ApplModuleImpl). In der Methode wird dann nativ über JDBC auf die PL/SQL-Logik zugegriffen. Dabei sollte für den Aufruf von Funktionen ein CallableStatement verwendet werden, um die Rückgabewerte entgegennehmen zu können. Für den Aufruf von Funktionen ist ein PreparedStatement zu verwenden.

2. *Kapselung der PL/SQL-Logik in Web Services.*

Für diesen Ansatz existieren ebenfalls zwei Möglichkeiten:

- a. Web Services innerhalb der Datenbank. Auf diese Weise erhält man native SOAP Services.
- b. Zwischenschalten des APEX-Listeners. Hierdurch kann man über RESTful Services auf die PL/SQL Logik zurückgreifen.

Die Webservices können dann entweder über das WebService-DataControl oder über WebService-Proxy Klassen in die ADF-Anwendung integriert werden. In der Praxis hat sich dabei die Verwendung von Webservice Proxies als flexibler erwiesen.

Weiteregehende Literatur/Links

- Vortrag DOAG Development 2012: Jürgen Menge - Oracle ADF - bestehende PL/SQL-Logik ersetzen oder wiederverwenden ?
- <http://www.oracle.com/technetwork/java/javaee/overview/index.html>
- <http://www.oracle.com/technetwork/articles/muir-designer-085584.html>
- <http://www.youtube.com/watch?v=gOUUIRjDpU&feature=plcp>
- http://download.oracle.com/otn/java/appexpress/1.1/docs/AELIG/E21058_01.pdf

Kontaktadresse:

Hendrik Gossens

MT AG

Balcke-Dürr-Allee, 9

D-40882 Ratingen

E-Mail hendrik.gossens@mt-ag.com

Internet: www.mt-ag.com