

Application Development Framework (ADF)

Teil 1: Überblick

Autor(en): Kersten Mebus, Jürgen Menge

Oracle Deutschland GmbH

Die Entwicklung von Java/J2EE Anwendungen wird durch den Einsatz von Design Pattern (Entwurfsmustern) deutlich vereinfacht. Entwurfsmuster stellen eine wiederverwendbare Vorlage von objektorientierten Konzepten aus anderen Anwendungen bzw. von anderen Entwicklern dar, die in einem spezifischen Kontext einsetzbar sind. Entwickler können sich diese Muster zu Nutze machen, um Aufwände durch Wiederverwendung zu verringern. Neben Entwurfsmustern verwenden Entwickler zum Erstellen von Applikationen mit Standardverhalten häufig Frameworks. Frameworks geben in der Regel die Architektur einer Anwendung vor und implementieren relevante Design Pattern, mit dem Ziel, die Entwicklungsgeschwindigkeit bedeutend zu erhöhen. Das Oracle „Application Development Framework“ (ADF) ist ein Java/J2EE Framework, mit dem Anwendungen einfach, visuell, deklarativ und somit effizienter entwickelt werden können.

Welche Möglichkeiten das Application Development Framework von Oracle bietet, zeigt unsere mit dieser Ausgabe startende Artikelserie. Die nächsten Ausgaben der DOAG News zeigen an Hand eines durchgängigen Beispiels wesentliche Einsatzbereiche von ADF auf :

- *Oracle ADF – der Einstieg in die J2EE-Welt für Forms-Entwickler*
- *Standardisierte Persistenzmechanismen in Verbindung mit ADF*
- *Oracle Web Center Suite - Portale der neuen Web 2.0 Generation mit ADF*

Einführung in ADF

Das Oracle Application Development Framework ist eine Zusammenführung von mehreren Frameworks, um End-zu-End J2EE-Anwendungen auf Basis des Modell-View-Controller (MVC) Entwurfsmusters zu entwickeln. ADF wurde im Sommer 2003 von Oracle als Produkt auf den Markt gebracht und kann im weitesten Sinne als kontinuierliche Weiterentwicklung des im Jahre 1999 von Oracle eingeführten Frameworks *Business Components for Java (BC4J)* betrachtet werden. Dieses Framework ist unter dem Namen *ADF Business Components (ADF BC)* in die ADF-Architektur integriert, die um weitere Frameworks, wie z.B. *Java Server Faces (JSF)*, *TopLink*, *Struts* erweitert wurde.

Die Idee von ADF ist die strikte Trennung zwischen Daten (Modell), die durch die Geschäftslogik gekapselt werden, und grafischer Darstellungsschicht (View) sowie der zur Darstellungsschicht gehörenden Steuerungseinheit (Controller). Dieses Konzept, dargestellt in Abb. 1, hat den Vorteil, dass die Geschäftslogik zum Beispiel mit Hilfe von *Enterprise JavaBeans (EJB)* oder der *ADF Business Components*, etc. entwickelt und somit unterschiedlichen Präsentationstechnologien zur Verfügung gestellt werden kann. Änderungen an der Geschäftslogik lassen die Implementierung der Präsentationsschicht unberührt und umgekehrt.

Die Steuerung innerhalb der Präsentationsschicht einschließlich der Benutzerinteraktion wird über einen Controller erreicht.

Für die Präsentationsschicht kann zwischen *ADF Swing* für Java-Swing Applikationen (Rich Client) und *JavaServer Pages (JSP)*, *JavaServer Faces (JSF)* sowie *ADF Faces* für webbasierte Anwendungen gewählt werden. *Oracle ADF Faces* stellt eine umfangreiche Bibliothek von Benutzeroberflächenkomponenten bereit, die auf dem Standard JSF (JSR-127) basieren. *ADF Faces* bietet eine größere Komponentenvielfalt als JSF an. So enthält *ADF Faces* zum Beispiel hierarchische Tabellen, Baumnavigation, Tabellen mit integrierter Sortierung, partielle Seitenwiedergabe (kein Refresh der kompletten Seite mehr notwendig), clientseitige Validierungen, Verwendung von Skins um das Look und Feel einer Anwendung zu definieren, usw. Diese Komponenten wurden 2007 von Oracle dem Apache MyFaces-Projekt *Trinidad* zur Verfügung gestellt.

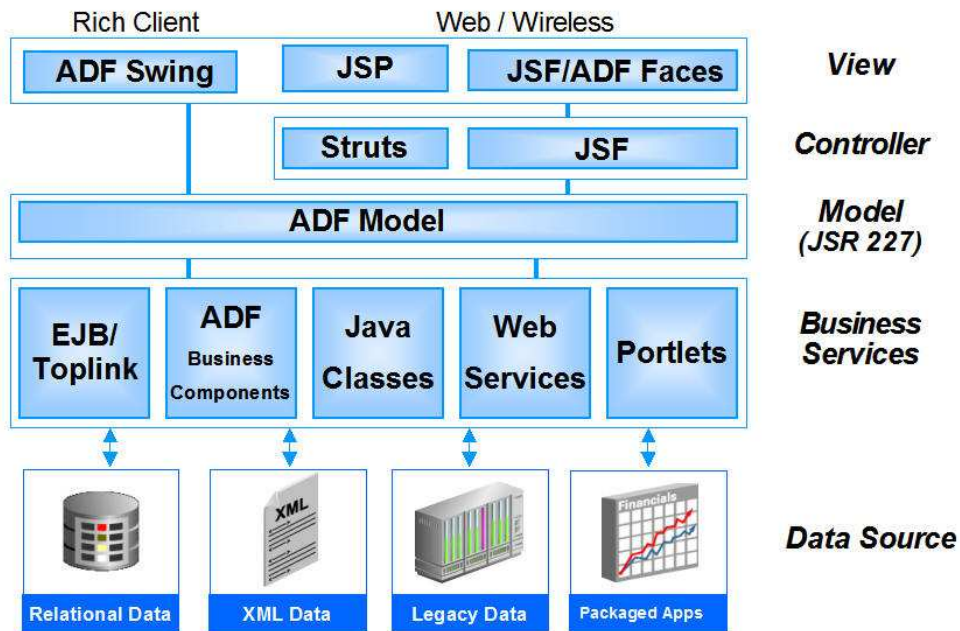


Abbildung 1: ADF-Architektur

Da die Modell-Schicht den zentralen Bestandteil von Oracle ADF bildet, wird sie im folgenden Abschnitt näher erläutert.

Die ADF Modell-Schicht

Das ADF-Modell ist eine Abstraktionsschicht zwischen grafischer Benutzeroberfläche (View) und der zugrundeliegenden Geschäftslogik (Business Services). Das ADF-Modell implementiert den Java Specification Request JSR-227 „A Standard Data Binding & Data Access Facility for J2EE“. Der Vorteil in der Nutzung von ADF ist, dass das ADF-Modell die Dateninhalte über die Geschäftslogik einem Userinterface ohne eigenen Programmieraufwand zur Verfügung stellt, d.h. in eine Präsentationsschicht bindet (Data Binding). Neben dem Data Binding ist das Data Control eine weitere Schicht des ADF-Modells. Data Controls sind eine Abstraktion der Business Services, die der Bindingschicht den konsistenten Zugriff auf Daten unterschiedlicher Services ermöglichen. Daher beschreibt ein Data Control das Public Interface eines Business Service. Data Bindings verbinden somit UI-Komponenten mit Daten oder Aktionen, die die Data Controls zur Verfügung stellen. Data Controls und Data Bindings werden mit Hilfe von XML-Metadaten automatisch definiert.

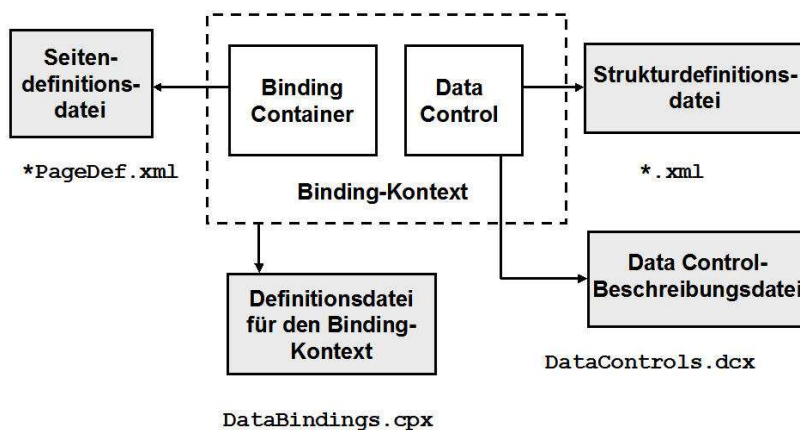


Abbildung 2: Objekte und Metadateien des ADF-Modells

Alle in einer Seite (JSF, JSP oder ADF Faces) verwendeten Bindings (siehe Abb. 2) werden in einer Seitendefinitionsdatei beschrieben und in einem Binding Container zusammengefasst. Es gibt einen Binding Container pro Seite. Der Binding-Kontext ist das Objekt (Handle), über das der Client auf die Data Binding-Schicht zugreift. Die Datei *DataBindings.cpx* ordnet jede Seite ihrer Seitendefinitionsdatei und den von ihr verwendeten Data Controls zu. Jeder Binding-Kontext, jeder Binding Container und jedes Data Control-Element verfügt über eine eigene Metadatei. Eine Metadatei ist eine XML-Datei, die alle anwendungsspezifischen Informationen enthält, die das ADF-Modell zur Laufzeit benötigt (siehe Abb. 2).

Data Controls werden in der ADF-Entwicklungsumgebung des *Oracle JDeveloper* durch das Anklicken mit der rechten Maustaste auf die Business Service Komponente (Ausnahme: ADF Business Components) und der Kontextmenüauswahl *Create Data Control* erzeugt. Für die *ADF Business Components* werden die Data Controls im *JDeveloper* automatisch zur Verfügung gestellt. Alle Data Controls einer Applikation werden visuell in einer *Data Control Palette* einschließlich ihrer Methoden, Parameter, Attribute, Collections und integrierten Operationen aufgelistet. Diese Komponenten werden mittels Drag & Drop auf die zu entwickelnde Benutzeroberfläche im visuellen Editor des *JDeveloper* gezogen. Anschließend muss noch ausgewählt werden, wie die Komponente dargestellt werden soll, zum Beispiel als Tabelle, Form, Textfeld, ListBox, etc.. So wird in Abb. 3 beispielhaft die Sichtweise eines Reisebürokunden (hier: *KundenView1*) durch Drag & Drop aus der Data Control Palette in den grafischen Editor als Formular (ADF Form) dargestellt. Das Data-Binding wird automatisch erstellt, wenn einer Seite die datengebundene Komponente (*KundenView1*) mittels Drag & Drop hinzugefügt wird (siehe Abschnitt: *Vorstellung des Demo-Beispiels*).

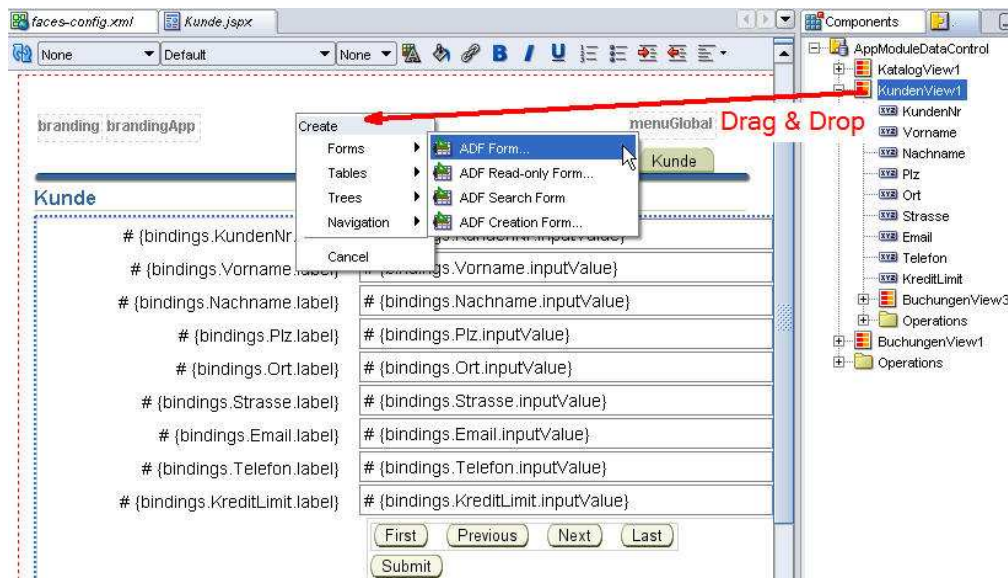


Abbildung 3: Erstellen einer datengebundenen Komponente ohne Programmieraufwand

Entwicklungs- und Laufzeitumgebung für Oracle ADF

Der *Oracle JDeveloper* ist für ADF-Anwendungen die am besten geeignete, integrierte Entwicklungsumgebung (IDE). Das kostenlose Werkzeug ermöglicht eine deklarative, grafische sowie programmatische Arbeitsweise. Eine Vielzahl von Funktionen, wie z.B. die Nutzung von ADF-spezifischen Wizards, bietet eine durchgängige Unterstützung aller Ebenen der ADF-Architektur.

Über Plug-Ins kann eine Versionskontrolle (*Subversion* u.a.) und ein Built-Management auf Basis von *ant* und *Maven* integriert werden. Die entwickelten Applikationen lassen sich mittels des eingebetteten J2EE-Containers (*OC4J*) unmittelbar im *JDeveloper* testen.

Notwendige Sicherheits-Überprüfungen, die die Authentifizierung und Autorisierung der Benutzer umfassen, können in ADF-Applikationen auf Basis des Standards *Java Authentication and Authorization Service* (JAAS) implementiert werden. In diesem Fall sorgt der J2EE-Container dafür, dass alle Aktionen eines Benutzers innerhalb des festgelegten Sicherheits-Kontextes ausgeführt werden. Die erforderlichen Informationen über Benutzer, Rollen und Rechte (*Credentials*) können in Dateien oder in einem LDAP-Verzeichnis verwaltet werden.

Für die Übernahme in eine Test- oder Produktionsumgebung müssen ADF-Web-Applikationen auf einem Applikations-Server installiert (*deployed*) werden. Dazu sind in der Laufzeit-Umgebung verschiedene ADF-Bibliotheken erforderlich. Dabei ist zu beachten, dass die eingesetzten Applikations-Server für den Betrieb von ADF zertifiziert sein müssen. Sollen relationale Datenbanken (SQL-92) für die Persistenzierung genutzt werden, ist im Falle des Einsatzes der *ADF Business Components* oder von *Oracle TopLink* auch die Zertifizierung der Datenbanken zu beachten. Alle entsprechenden Informationen sind im Oracle Technet bei den Produktinformationen zu *Oracle JDeveloper/ADF* zu finden.

Aktueller Status und Ausblick Oracle ADF 11g

Oracle ADF ist eine der Basis-Technologien im Projekt *Fusion*, in dem die neue Generation der betriebswirtschaftlicher Software von Oracle nach den Prinzipien einer Service-orientierten Architektur (SOA) entwickelt wird. Daneben wurden in den vergangenen Jahren eine große Zahl von Kunden-Projekten mit ADF erfolgreich umgesetzt.

Mit der bereits in einer vierten Vorab-Version (*Technical Preview 4*) vorliegenden Software von Oracle ADF 11g bleiben die bereits von ADF 10g bekannten Bausteine erhalten, werden aber u.a. durch wesentliche Neuerungen erweitert:

- Mit den *ADF Faces Rich Client Components* werden zusätzliche interaktive Oberflächen-Elemente auf Basis von AJAX (Asynchrones JavaScript and XML) bereitgestellt.
- Ein leistungsfähiger Template-Mechanismus erlaubt die flexible Gestaltung von Seiten und Seiten-Fragmenten.
- Wiederkehrende Abläufe (z.B. Login-Prozess) können als Task Flow definiert und in einer Anwendung wiederverwendet werden.

Vorstellung des Demo-Beispiels

Während der Artikelserie wird durchgängig das Beispiel einer fiktiven Reisebuchungs-Applikation verwendet.

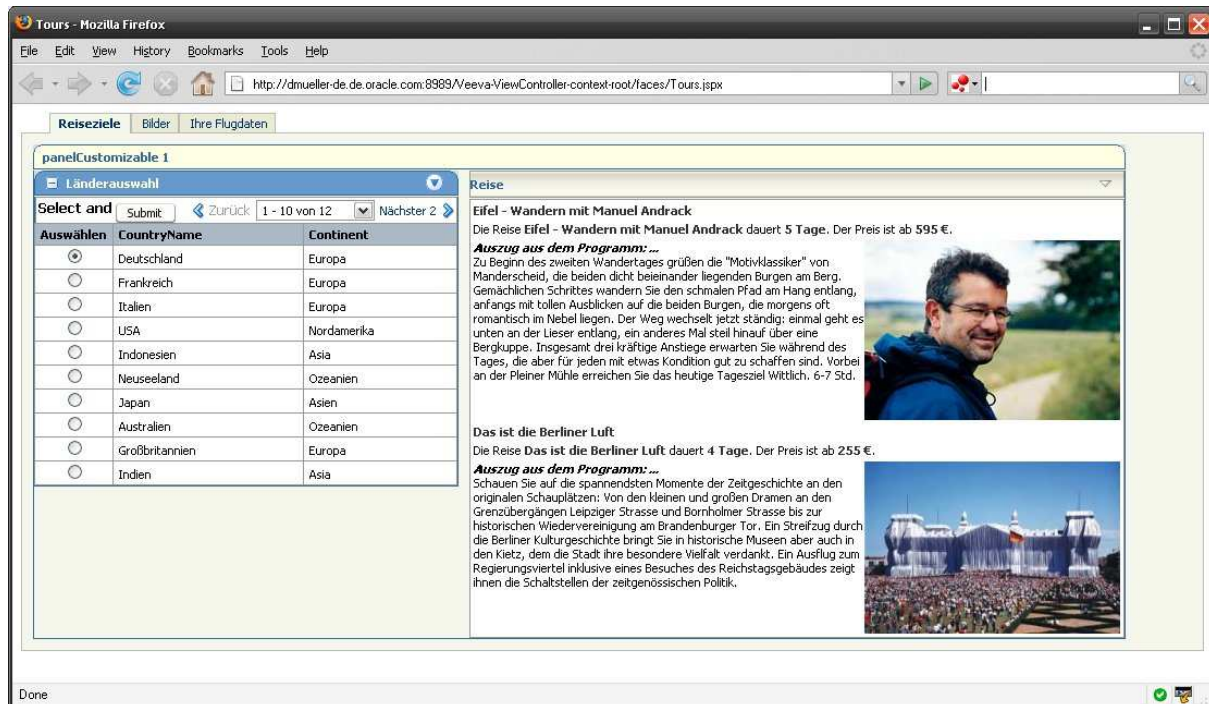


Abbildung 4: Reisebuchungs-Applikation

Diese Applikation soll von drei verschiedenen Anwendergruppen genutzt werden:

- **Kunden**
- **Reisebüros** als Vermittler und
- **Veranstalter** als Produkthersteller.

Der Kunde sucht in Daten und Inhalten, die der Veranstalter über das Reisebüro bereitstellt. Dies bedeutet, dass verschiedene Integrationsszenarien betrachtet werden müssen, da die Inhaltstypen und Quellen vielseitig sind. Über das Reisebüro wird der Buchungsprozess ausgeführt.

Das Datenmodell der Anwendung besteht aus den 3 Datenbank-Tabellen KUNDEN, BUCHUNGEN und KATALOG.

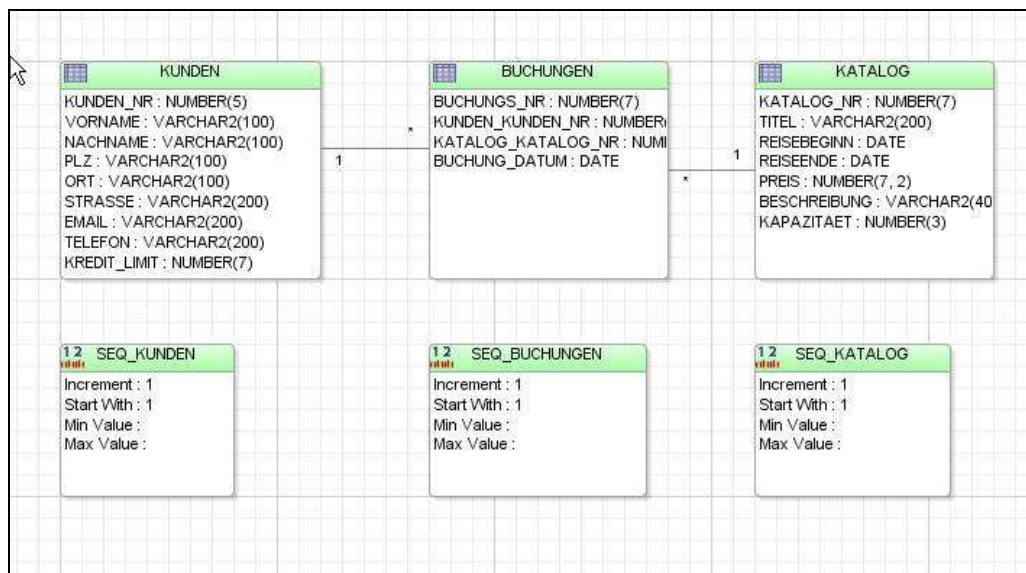


Abbildung 5: Datenmodell der Reisebuchungs-Applikation

Der Übersicht halber und zum besseren Verständnis der ADF-Grundlagen wird die Applikation nicht in allen technischen Einzelheiten beleuchtet. So wird zum Beispiel auf die Darstellung einer Implementierung der Sicherheits-Mechanismen verzichtet.

Zusammenfassung

Das Oracle Application Development Framework ist eine Zusammenführung von mehreren Frameworks, um Java-EE-Anwendungen auf Basis des Modell-View-Controller (MVC) Design Patterns zu entwickeln. Das Framework bietet eine durchgehende Unterstützung für die Erstellung der grafischen Benutzeroberfläche einschließlich des Kontrollflusses, der Implementierung der Geschäftslogik sowie der Datenzugriffsschicht an. Ziel des Frameworks ist es, die Anwendungsentwicklung anhand eines visuellen und deklarativen Entwicklungsansatzes zu vereinfachen und zu beschleunigen. Auf Grund der offenen Architektur können Entwickler ihre ADF-Anwendungen jederzeit durch selbstgeschriebenen Code erweitern bzw. Funktionalität und Verhalten des Frameworks anpassen. Das Application Development Framework können sowohl erfahrene als auch nicht erfahrene Java-EE-Programmierer produktiv einsetzen. Folgende Zielgruppen lassen sich identifizieren:

- Entwickler mit einem prozeduralen oder 4GL-Hintergrund, die Entwicklungsumgebungen, wie Oracle Forms, Visual Basic, etc. einsetzen
- Entwickler mit objektorientiertem Programmierwissen und/oder einem Java-EE-Hintergrund

In der nächsten Ausgabe der DOAG News werden wir mit Teil 2 „Oracle ADF – der Einstieg in die J2EE-Welt für Forms-Entwickler“ unsere Reihe fortsetzen.

Nützliche Links

Produktseite Oracle ADF im OTN

<http://www.oracle.com/technology/products/adf>

Produktseite Oracle JDeveloper im OTN

<http://www.oracle.com/technology/products/jdev>

Oracle ADF Learning Center

<http://www.oracle.com/technology/products/adf/learnadf.html>

Deutsche ADF Community

<http://www.oracle.com/global/de/community/adf>

Kontaktadressen:

Michael Bräuer

michael.braeuer@oracle.com

Kersten Mebus

kersten.mebus@oracle.com

Jürgen Menge

juergen.menge@oracle.com

Detlef Müller

detlef.mueller@oracle.com