

Das APEX Migrationsprojekt bei der Union Investment

Niels de Bruijn
MT AG
Ratingen

Schlüsselworte:

Oracle Application Express 3.2.1 (APEX), jQuery, SAP Business Objects XI, MS Access 2007, Oracle Forms 6i.

Einleitung

Die Union Investment betreibt das Oracle basierte System „OraDIS“, das zur verteilten Pflege primär von Stammdaten dient. Vor der Migration wurden hierfür als Frontend MS Access 2007 sowie Oracle Forms 6i eingesetzt. Ein fehlender Support für Oracle Forms 6i und die schlechten Reaktionszeiten von MS Access 2007 waren die maßgeblichen Auslöser für das Migrationsprojekt. In einem „Proof of Concept“ wurde daraufhin Ende 2009 eine Realisierung mittels APEX geprüft. Anschließend wurden im Zeitraum April bis Mitte August 2010 in einem Festpreisprojekt etwa 160 funktionale Dialoge nach APEX 3.2.1 migriert. Seit dem 26. August 2010 wird die Anwendung durch 100 Endanwender produktiv verwendet.

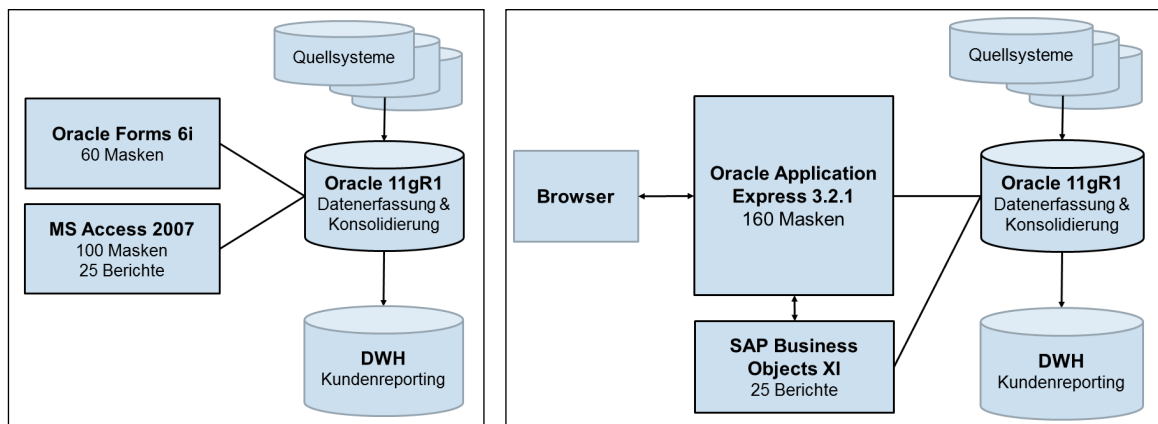


Abb. 1: Alt versus Neu.

Warum APEX?

Die Frage stellte sich nachdem klar war, dass die bestehende Lösung nicht mehr beibehalten werden konnte. Oracle Forms 10g/11g wäre eine Alternative gewesen, wären nicht die Lizenzkosten so hoch gewesen. Für APEX selber fallen keine Lizenzkosten an, es ist nicht von Plugins im Browser abhängig, bietet viele Komponenten ohne Programmieraufwand an und ist für PL/SQL-Entwickler schnell erlernbar. Allesamt Kriterien, die eine Entscheidung für APEX erleichtert haben.

Vom Client-Server ins Web

Während der Proof-of-Concept-Phase wurde schnell klar, dass die Migration nach APEX eine große Änderung in der Bedienung mit sich bringt, worauf die Endanwender vorbereitet sein müssen. Eine Client-Server-Anwendung verhält sich nun mal anders als eine Webanwendung. Merkbar wird dies beispielsweise dadurch, dass man in einer Webanwendung explizit speichern muss. In MS Access wird dagegen beim Verlassen des Feldes implizit gespeichert. Auch das Locking-Verhalten ist unterschiedlich: Während in MS Access ein Datensatz nicht geändert werden kann, wenn dieser bereits vorher durch einen anderen Endanwender geändert wurde, muss diese Prüfung in einer Webarchitektur in der Anwendung stattfinden. Auch die Authentifizierung ändert sich: aus einem Datenbankbenutzer wird ein Anwendungsbrowser. Der aktuelle Benutzer in der Datenbank mittels „USER“ abfragen funktioniert mit einer APEX-Anwendung nicht mehr. Stattdessen ist „v('APP_USER')“ zu verwenden.

Die zweite Herausforderung stellte die Integration von Business Objects Berichten dar. Über APEX können etwa 25 vordefinierte PDF/Excel Berichte abgefragt werden, die vorher mit MS Access abgebildet waren. Anfänglich wurde versucht, den Aufruf über eine Java Klasse in der Datenbank zu ermöglichen, jedoch waren die Antwortzeiten von Business Objects dabei nicht akzeptabel. Im zweiten Lauf wurde stattdessen eine URL-Integration vorgenommen: Die Datenbank fungiert als Webclient für den Business Objects Server und leitet das Dokument weiter an den Browser. Obwohl die Architektur von Business Objects nicht auf Geschwindigkeit ausgerichtet ist, wurden die Antwortzeiten von durchschnittlich 7 Sekunden pro Aufruf durch die Fachbereiche noch als „akzeptabel“ eingestuft.

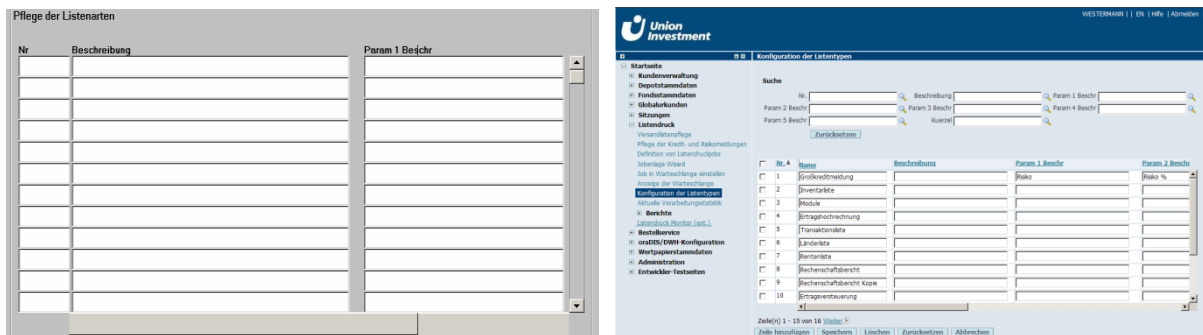


Abb. 2: Von Oracle Forms 6i nach Oracle Application Express 3.2.1.

Unterschiede zwischen MS Access und APEX

Auch während der Realisierung wurde das Entwicklungsteam noch mit Funktionalitäten konfrontiert, die zwar in MS Access verwendet wurden, jedoch schwer oder gar nicht mit der Webtechnologie realisierbar sind.

Beispiel: Ein Kontextmenü über die rechte Maustaste aufzurufen, um einen Filter auf ein bestimmtes Feld zu setzen, funktioniert zwar in MS Access, jedoch nicht so in APEX. Auch Checkboxen mit drei Statusmöglichkeiten (leer, angekreuzt, grauer Hintergrund) gibt es im Webumfeld nicht als Standard. Ebenso nicht im Standard enthalten ist eine Auswahlliste, in der mehrere Informationen ausgerichtet dargestellt werden. Auffällig ist auch, wie wenig platzsparend die Elemente auf einer APEX-Seite sind, wenn eine Standardvorlage verwendet wird. Folge: Der Endanwender muss viel häufiger scrollen als eigentlich notwendig ist.

In MS Access 2007 können Anpassungen nur dann vorgenommen werden, wenn die Datei nicht gerade durch einen anderen Entwickler geöffnet ist. Und sind die Entwicklungen fertig, dann muss aus der Entwicklungsdatenbank die Abnahmedatenbank gemacht werden. Die Folge ist ein aufwändiges Ändern von Umgebungsvariablen in der Access-Datei. Mit APEX hat man diese Probleme nicht. Besonders der Rolloutprozess ist in APEX sehr komfortabel und sogar auf Seitenebene möglich.

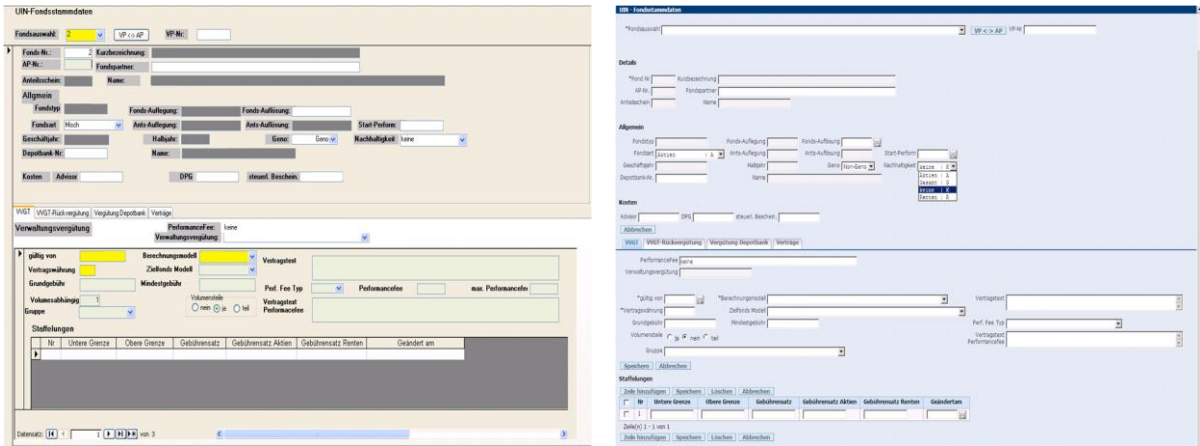


Abb. 3: Von MS Access 2007 nach Oracle Application Express 3.2.1.

Das „Model-View-Controller“-Prinzip mit APEX

Ein APEX-Objekt operiert niemals direkt auf einer Tabelle. Stattdessen wurden zunächst funktionale Views bereitgestellt, die durch seitenspezifische Views referenziert werden. Nur diese seitenspezifischen Views werden in APEX neben seitenspezifischen Packages auf der zugehörige Seite referenziert. Diese Struktur kommt der Wartbarkeit der Anwendung zugute und reduziert den Aufwand bei einem potenziellen Austausch der Benutzeroberfläche.

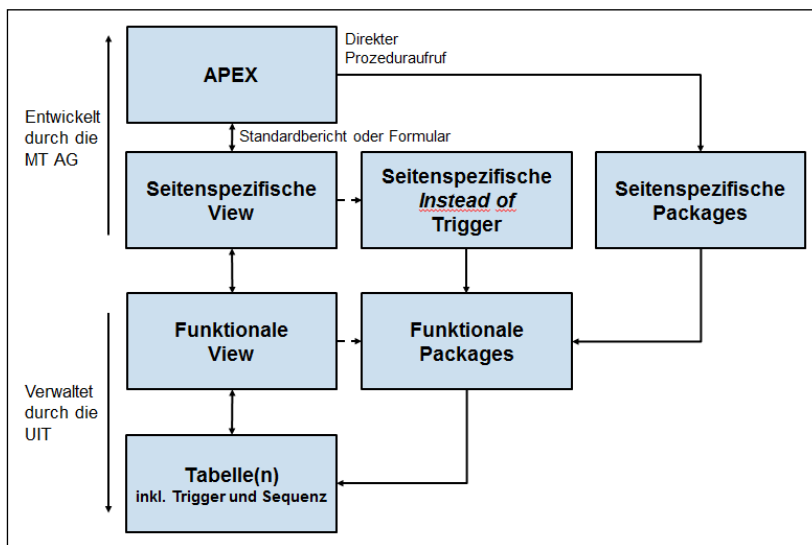


Abb. 3: Saubere Trennung von der Oberfläche, die Geschäftslogik und das Datenmodell.

Rapid Application Development

Sehr wichtig während der Realisierung war die Verwendung von Standardkomponenten in APEX. Formulare und Berichte wurden so oft wie möglich über Assistenten in APEX erzeugt. Im Falle eines Formulars spart man hier viel Zeit im Vergleich zu einem manuellen Vorgang.

Die logische Begrenzung auf nur einen View für die automatische Verarbeitung in APEX kann mit dem Einsatz von einem „Instead-of“-Trigger auf den View gelöst werden. So erhält man die Flexibilität für die Datenverarbeitung, ohne dabei auf den Einsatz von APEX-Assistenten verzichten zu müssen.

Nicht in allen Fällen war der Einsatz eines Assistenten möglich. So kam es öfter vor, dass mehrere „Single-Row“-Formulare, tabellarische Formulare oder interaktive Berichte pro Seite erwünscht waren. Nur leider können aktuell derartige Komponenten nur einmal pro Seite über den Assistenten erstellt werden. Mehrere sind zwar möglich, aber ab dem zweiten muss man dann auf den Assistenten verzichten und die Elemente, Prozesse und die Prüfsumme manuell erstellen.

Ohne jQuery geht es nicht

An vielen Stellen musste mit der JavaScript-Bibliothek „jQuery“ Nachhilfe in Anspruch genommen werden. Ob Designkorrekturen oder clientseitige Validierungen: Ohne den Einsatz von jQuery wären bestimmte „Ungereimtheiten“ in APEX nur schwierig zu beheben gewesen. Nehmen wir einmal folgendes Beispiel: In APEX werden die Eingaben erst dann überprüft, wenn der Endanwender das Formular versendet. Einfache Validierungen wie „korrektes Datumsformat verwendet“ oder „nicht mehr als 20 Zeichen erlaubt“ können jedoch durchaus während der Eingabe erfolgen. Für derartige Validierungen wurde deshalb auf Basis von jQuery eine Palette mit generischen Funktionen bereitgestellt und in der Anwendung verwendet.

Mantis: nicht nur für Bugtracking

Als Ticketingsystem wurde das Open Source Tool „Mantis“ sowohl bei der Union Investment als auch bei der MT AG eingesetzt. Alle Masken wurden in diesem System als Ticket hinterlegt, damit der Status pro Maske jederzeit ersichtlich war. Pro Ticket wurde eine fachliche Beschreibung der Maske hinterlegt. Nach und nach wurden die Masken, verteilt in vier Arbeitspakete, umgesetzt und anschließend durch einen anderen Mitarbeiter des Projektteams getestet. Nach der Fertigstellung eines Arbeitspakets erfolgte jeweils die Auslieferung der Anwendung sowie der geänderten Datenbankobjekte an die Union Investment. Dort wurde die Anwendung in die Abnahmeumgebung eingespielt und getestet. Gefundene Fehler wurden in Mantis als Unterticket für die Maske registriert, abgearbeitet und mit der nächsten Auslieferung wieder mit ausgeliefert.

Lessons learned

Für derartige Migrationsprojekte gilt auch für APEX die 80/20 Regel. 80% lassen sich sehr schnell realisieren, wenn die Masken über Assistenten in APEX erstellt werden können. Die restlichen 20 % lassen sich zwar realisieren, erfordern jedoch einen erheblichen Mehraufwand. Wichtiger ist jedoch die Erkenntnis, dass APEX-Projekte sich gut kalkulieren lassen. Und es sind durchaus auch größere und wartungsfreundliche Webanwendungen mit APEX realisierbar – ein korrekter Umgang mit dem Produkt ist dabei natürlich vorausgesetzt.

Kontaktadresse:

Niels de Bruijn
MT AG
Balcke-Dürr-Allee 9
D-40882 Ratingen

Telefon: +49 (0) 173 233 24 06
Fax: +49 (0) 21 02 309 61-50
E-Mail: niels.de.bruijn@mt-ag.com
Internet: www.mt-ag.com/apex