Des Kaisers neue Kleider – Entwicklung dynamischer Oberflächen mit ADF

Markus Klenke
TEAM GmbH
Paderborn

Schlüsselworte
Oracle ADF, RAD, Wiederverwendbarkeit, Deklarative Komponenten, Einsteiger

Einleitung

Rapid Application Development ist in aller Munde. Ob nun mit Forms oder Apex, bei vielen Anforderungen kann mit Hilfe dieser Tools sehr schnell ein Prototyp oder ein Showcase erstellt werden. An dieser Stelle muss man sich jetzt eine Frage stellen: Warum ist ADF häufig nicht in der Liste dieser Tools zu finden. Ist das Framework zu mächtig und träge um eine solche Anforderung zu erfüllen?

Um zu zeigen, dass ADF für jegliche Komplexitätsstufen ebenfalls eine Antwort mitbringt, wenn man sich nur auf das Framework einlässt, wird in einer Demo erklärt, wie sich die Komponenten von ADF am sinnvollsten nutzen lassen um nicht nur "rapid", sondern auch effektiv zu entwickeln.

Als Beispiel sollen dynamisch generierte Stammdatendialoge designt werden und in einer Rahmenapplikation bereitgestellt werden. Desweiteren werden einzelne Komponenten erstellt, die die spätere Weiterentwicklung der Applikation deutlich beschleunigen und vereinheitlichen.

Stammdatenbearbeitung mit Oracle ADF

In vielen Projekten beginnen die Entwicklungen mit einer Machbarkeitsstudie und einer Eingewöhnungsphase, in der Entwickler sowie Endanwender sich von einem Framework als Arbeitsumgebung bzw. von dem Look and Feel der Endanwendung überzeugen sollen. Somit müssen gerade zu Beginn einer Entwicklungsphase die Weichen auf möglichst schnelle und einfache Entwicklung bei guter Visualisierung und Bedienbarkeit gestellt sein.

ADF bietet genau für diesen Aspekt einen sehr stark auf Drag and Drop fokussierten Entwicklungsansatz. Gerade Stammdaten-Elemente, welche stark auf Geschäftsregeln und meistens weniger auf die Darstellung optimiert werden können von diesem Konzept profitieren. Zu Beginn jeder ADF Applikationsentwicklung wird das Datenmodell erstellt.

Dieses dient allerdings nicht nur dazu eine simple Datenreplikation zu erhalten, sondern vereint die Validierung von Geschäftsregeln mit der groben Vorgabe, wie sich Datenfelder auf der Oberfläche zu verhalten haben, miteinander. So können mit ADF auf sehr deklarative Weise schon in der Modellerstellung die Hauptaufgaben für die Stammdatenverarbeitung festgelegt werden ohne mit der Oberflächenentwicklung zu beginnen.

Ist die Entwicklung am Modell abgeschlossen gibt es für den Oberflächenentwickler diverse Möglichkeiten mit ADF eine Oberfläche zu erstellen. Die einfachste Möglichkeit ist es, die generierung der Oberflächenelemente vom Framework selbst übernehmen zu lassen. Dazu kann eine

einzige Seite erstellt werden und je nach Bedarf entweder eine Tabellen- oder Formularstruktur unterschiedlichster Daten erzeugt werden.

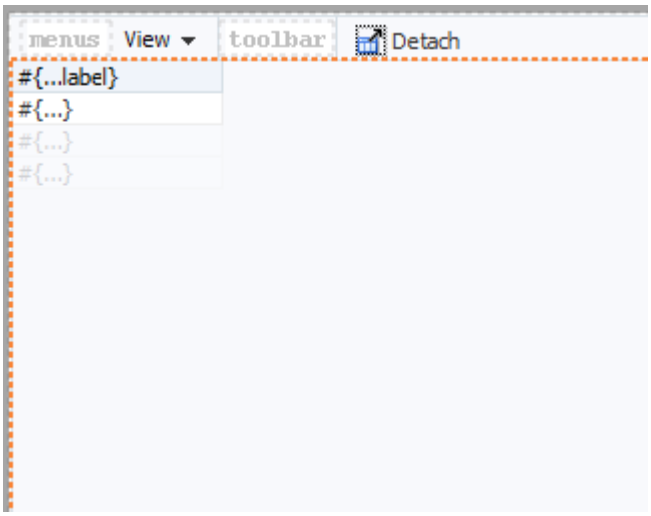

Abb. 1: Darstellung einer dynamisch erstellten Tabelle im JDeveloper

Diese Möglichkeit ist sicherlich die schnellste und einfachste, allerdings hat man die Oberflächen mit dieser Methode gerade zu Beginn der ADF Entwicklung nicht zu einhundert Prozent unter Kontrolle. Es ist daher häufig ratsam, für jedes Stammdatum eine eigene Oberfläche zu entwickeln, diese allerdings auf die verschiedenen Nutzungsebenen zuzuschneiden um sie somit später einfacher in den Kontext der „echten“ Applikation zu bringen.

Um eine Gleichförmigkeit zu erhalten und die Wartbarkeit der Applikation von vornherein sicherzustellen, sollten alle Oberflächen auf einem Stammdaten-Oberflächentemplate basiert sein. Sollten sich Änderungswünsche an Layout, Farbgebung oder anderen Oberflächeneigenschaften ergeben, ist die Änderung nur an diesem Template nötig.

Um einen hohen Wiederverwendungsgrad zu erhalten, bietet es sich an für jeden Stammdatendialog einen Task-Flow zu erstellen, also eine ADF Komponente, welche es erlaubt, einen eigenen Kontext für den Stammdatenbereich zu erstellen. So können an Task-Flows zum Beispiel Eingabe- und Ausgabeparameter eingestellt werden, um den Flow so kontextfrei wie möglich zu machen. Es können als Parameter auch ganze Modelleinheiten genutzt werden, was für den Einsatz des ersten beschriebenen Ansatzes nötig ist.

Um ein möglichst breites Spektrum an verschiedenen Möglichkeiten darzustellen, wie Stammdatendialoge erstellt werden können, werden in der Live-Demo beide Wege beschrieben und ausgearbeitet. Ein Dialog wird nach dem Verfahren des „Ein Task-Flow pro Stammdatendialog“ erstellt, ein weiterer wird vollkommen dynamisch erzeugt und bekommt alle benötigten Informationen als Parameter übergeben.

Hat man sich für ein Vorgehen entschieden, können die Stammdatendialoge für sich entwickelt werden und in einer prototypischen Rahmenanwendung eingebettet werden um beispielsweise vom Endnutzer getestet werden.

Entwicklung dynamischer deklarativer Komponenten
Sind die ersten Erfahrungen mit ADF gemacht und der Prototyp in fertigem oder weit fortgeschrittenem Zustand, so gilt es nun den Weg von der Entwicklung mit der heißen Nadel hin zu einer Entwicklung einer Unifikation bzw. Verallgemeinerung zu finden. Ab diesem Zeitpunkt wird es enorm wichtig, alle Komponenten auf eine gemeinsame Basis zu stellen.

Während zu Beginn die Notwendigkeit von Templates noch relativ offensichtlich ist, stellen sich zum jetzigen Zeitpunkt etwas technischere Fragen. Werden die Standardkomponenten von ADF genutzt oder werden mehrere Komponenten oder ein eigener Layoutcontainer erschaffen, um so die Arbeit aller Mitarbeiter im Projekt zu vereinfachen und gegebenenfalls zu beschleunigen.

Ein typisches Beispiel für eine Komponente, die immer wieder eine sinnvolle Verwendung findet, ist eine Dropdown-Liste von Elementen, welche einen Parameter setzt. Natürlich gibt es von ADF die Möglichkeit solche Listen anzulegen, jedoch ist die Wiederkehr dieser Komponente so hoch, dass der jeweilige Aufwand innerhalb eines Projektes nicht zu unterschätzen ist. Daher macht es Sinn, eine Komponente zu erstellen, die einfach parametrierbar ist und in all diesen Fällen genutzt werden kann um beispielsweise Parameter aus einer Master-Tabelle oder Konstanten aus einer Konfigurationstabelle zu erstellen.

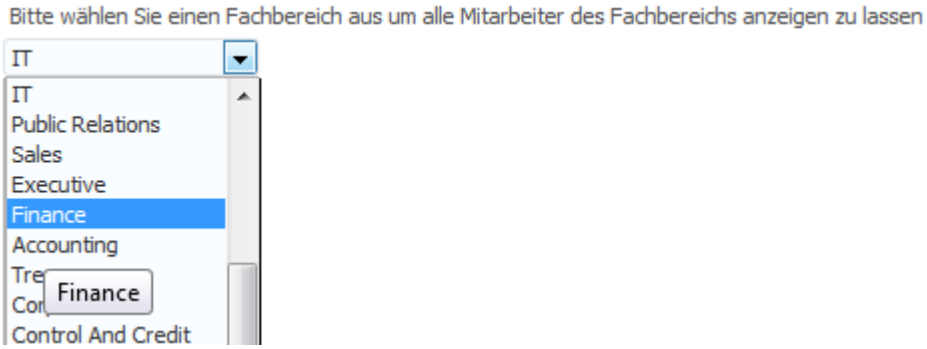

Abb. 2: Rendering einer deklarativen Komponente zur Auswahl eines Parameters

Komponenten dieser Art werden in einem Basismodul entwickelt und können somit in allen anderen Bereichen der ADF Entwicklung genutzt werden.

In der Demo soll eine Komponente dieser Art von Grund auf erstellt werden. Diese Komponente wird zusammen mit einem Seitentemplate als Library deployt und von ein er weiteren Komponente eingeladen. In der Demonstration wird beschrieben, wie sich das Erstellen und Benutzen von deklarativen Komponenten anfühlt und auf welche Besonderheiten man bei der Entwicklung jener achten sollte.

Um die Funktionalität der Komponente sicherzustellen und eine effiziente Arbeitsweise aufzuzeigen, wird an diesem Teil der Weg des deklarativen Entwickelns hin und wieder verlassen um weitere Facetten des Entwickelns mit ADF zu zeigen. So wird die Datenanbindung der List-Of-Values programmatisch über eine Funktion am Application Module erstellt, diese dann aber wieder über das Seitentemplate deklarativ an die Oberfläche gebunden wo sie dann von der von uns erstellten Komponente aufgerufen wird. Ein beispiel für den Zugriff auf die Bindings des Page Templates kann so aussehen:

```
DCBindingContainer templateContainer =
(DCBindingContainer)ADFUtils.getBindings().get("ptb1");

DCIteratorBinding constantIterator =
templateContainer.findIteratorBinding("ConstantLookupIterator");
```

Ist die Komponente fertiggestellt, so wird eine Haupteinstiegsseite erstellt, in der die Komponente genutzt wird um einen Parameter auszuwählen und somit die Darstellung der zuvor erstellten Stammdatendialoge zur Laufzeit zu verändern.

Ziel des Vortrags ist es, den Zuhörern einige Mittel aufzuzeigen, die Entwicklung mit ADF zu beschleunigen. Dabei werden nicht nur Themen angesprochen, welche für Anfänger relevanz haben, sondern es wird auch auf Komponenten und Arbeitsweisen eingegangen, welche eventuell auch für erfahrene Entwickler Möglichkeiten bieten ihr Arsenal an Funktionalitäten von ADF zu erweitern.

Kontaktadresse:
Markus Klenke
TEAM GmbH
Hermann Löns-Straße, 88
D-33104 Paderborn Schloß-Neuhaus

Telefon: +49 (0) 5254-8008-55
Fax: +49 (0) 5254 8008-19
E-Mail mke@team-pb.de
Internet: www.team-pb.de