Datenvisualisierung mit dynamischen Pivot-Tabellen in der Praxis

Timo Hahn
virtual7 GmbH
Karlsruhe

Schlüsselworte
Pivot-Tabellen, dynamische GUI, ADF, JDeveloper11g, Tabellen, BindingLayer, Iterator, Cubic Editor, Datenvisualisierung

Einleitung
Als die führende und größte international tätige Fenstermarke Europas bietet INTERNORM anspruchsvollen Kunden richtungsweisende Lösungen für Fenster und Türen. Seit der Gründung im Jahr 1931 hat das Familienunternehmen mit Firmensitz in Traun/OÖ mehr als 20 Millionen Fenstereinheiten produziert. Heute beschäftigt INTERNORM mehr als 1.950 Mitarbeiter. Von der eigenen Extrusion der Profile über die Fenster-Fertigung bis zur Isolierglasproduktion werden alle Arbeitsschritte in den drei österreichischen Produktionswerken Traun, Sarleinsbach und Lannach durchgeführt.

Im Rahmen eines ADF Projekts „Internorm Salesbook“ wurde für die Firma INTERNORM ein Konzept zur dynamischen Erstellung von Auswertungen mittels Pivot Tabellen erstellt und implementiert. Basierend auf Auftragsdaten, die in einem J.D. Edwards System für Mandanten gesammelt und tagesaktuell für die Weiterverarbeitung in einer flachen Datentabelle in einer Oracle Datenbank bereitgestellt werden, soll der Benutzer selbstdefinierte Auswertungen erstellen können.

Die flache Datentabelle umfasst dabei ca. 50 Attribute. Es ist dabei dem Benutzer überlassen, welche Attribute aus der flachen Datentabelle er in der Pivot Tabelle anzeigen und wie diese zu aggregieren sind. Ebenso kann der Benutzer die Reihenfolge der Attribute und die zur Aggregation anzuwendende Funktion selbst auswählen und speichern

Detaillierter ‚Use Case'
Die INTERNORM stellt für Mandanten Daten über deren Kunden und Aufträge in einer flachen Tabellenstruktur bereit, die über ein J.D. Edwards System gesammelt und vorausgewählt werden. Die Daten des J.D. Edwards System werden periodisch in eine flache Datentabelle in eine Oracle DB geladen. Die hieraus resultierende Tabelle besteht aus mehr als Spalten, die alle notwendigen Daten für die vom Kunden gewünschte Auswertung beinhaltet. Die Daten in der Tabelle stehen allerdings nicht aggregiert bereit, so dass der Anwender die gewünschte Aggregation selber auswählen und die Daten entsprechend aufbereiten können muss.

Jeder Anwender kann aus der Menge der angebotenen Spalten eine Liste von Spalten auswählen, über die er die Datenmenge kaskadierend gruppieren will. Das heißt, jeder Spalte der vom Benutzer erstellten Attributliste ist eine Ordnungszahl zugewiesen, die die Reihenfolge der Gruppierung angibt. Somit werden zuerst die Daten der Spalte mit der Ordnungszahl 1 herausgefiltert bzw. gruppiert, anschließend innerhalb der so entstehenden Datenmenge die Daten mit der Ordnungszahl 2 gruppiert usw. Als Beispiel könnte z.B. die Länderkennung die Ordnungszahl 1, dem Bundesland die Ordnungszahl 2 und der Stadt die Ordnungszahl 3 zugewiesen werden, um eine Gruppierung nach Land. Bundesland und Stadt zu erhalten.

Aus dem Rest der noch nicht ausgewählten Attribute kann der Anwender die Attribute wählen, die in der Detailansicht auf unterster Ebene als Tabelle, der Detailansicht, dargestellt werden sollen. Für jedes in der Detailansicht sichtbares Attribute (Spalte) kann eine Aggregation (z.B. Summe oder Durchschnitt) für eine Berechnung hinterlegt werden. Das Ergebnis der Aggregation soll über der entsprechenden Spalte in der Vatergruppe angezeigt werden.

Die Auswahl der zu gruppierenden Spalten aus der Menge der verfügbaren Spalten ist nicht Bestandteil dieses Vortrags, kann aber leicht mit ADF realisiert werden. Aus den Informationen kann ein Meta-Objekt gebildet werden, welches personalisiert in der Datenbank abgespeichert werden kann. Die so erzeugten Meta-Objekte dienen später in der Oberfläche zur Auswahl einer benutzerdefinierten, aggregierten Darstellung, also einer dynamischen Pivot-Tabelle. Im Weiteren wird ein solches Meta-Objekt als ‚Bookmark' bezeichnet. Eine genauere Beschreibung folgt unten.

Lösungsszenarien
Es wurden 3 verschiedene Lösungsvarianten untersucht:

1. Programmatisches Pivot-Tabellen-Model
   - Nachteile:
     - schwer zu implementieren
     - erlaubt nicht die volle Funktionalität des Cubic-Editors, Aggregation und Abschnittsweisen Datenladens welches Standart-Pivot-Tabellen bereitstellen
2. Kontinuierliche Anpassung eines ‚Dummy' Pivot-Tabellen-Model pro Anforderung
   - Nachteile:
     - Änderungen an den ‚Bindings' die einen ‚Cubic Editor' verwenden sind nicht trivial
     - kein ‚caching' der Bindings für die einzelnen Anforderungen
3. Erzeugung eines Pivot-Tabellen-Binding einmalig pro neuer Anforderung

Nach der Untersuchung der drei Lösungen wurde die dritte Lösung erfolgreich implementiert. Im Folgenden wird nach einer Beschreibung des Meta-Models diese Lösung beschrieben.

Metadata-Objekt bzw. Bookmark
Ein Metadata-Objekt bzw. Bookmark beschreibt wie eine vom Anwender zusammengestellte dynamische Piviot-Tabelle später aussehen soll. Ein Bookmark pro Anwender wird dabei als 'Standard' markiert und wird immer dann angezeigt, wenn keine andere Auswahl durch den Benutzer erfolgte.

Ein Bookmark beschreibt dabei

- welche Spalten der flachen Datentabelle gruppiert werden sollen
- in welcher Reihenfolge die Gruppierungen angezeigt werden sollen
- wie innerhalb einer Gruppe die Daten sortiert werden sollen, wobei nach mehreren Spalten auf-/absteigend sortiert werden kann
- welche Aggregation für eine Spalte herangezogen werden soll. Dabei sind zurzeit die folgenden Aggregatsfunktionen implementiert: KEIN, SUMME oder DURCHSCHNITT. Dies kann aber leicht auf alle innerhalb einer Pivot-Tabelle möglichen Aggregationen erweitert werden.

Lösung im Detail
Für jedes Meta-Objekt oder Bookmark wird beim ersten Aufruf dynamisch ein 'Pivot Tabel Binding' mit den im Bookmark definierten Attributen, deren Reihenfolge und der gewählten Aggregation gebildet. Ein neues ViewObject und ein neuer Iterator für dieses ViewObject werden dynamisch angelegt um die Daten von der DB abzurufen. Das ViewObject verwendet einen aus den Bookmark generierten SQL String zur Abfrage der Daten. Es werden nur die im Bookmark definierten Daten über das ViewObject geladen.

Alle Namen der ViewObjecte, Iteratoren und PivotTableBinding, die für ein Bookmark gebildet werden, werden gecached und innerhalb der Session wiederverwendet.

Einige Implementierungsdetails sind:

1. Um in der Pivot-Tabelle die aggregierten Daten zusammen mit den Gruppen anzeigen zu können, wird jeder Zeile eine UID hinzugefügt. Dazu wird die ROWNUM, die bei Verwendung einer Oracle DB zur Verfügung steht, verwendet. Diese wird immer als letzte Spalte der im Bookmark definierten Attribute automatisch angefügt. Gleichzeitig wird die ROWNUM als ‚Drill Path' für die Definierten kaskadierenden Gruppen verwendet. Allerdings wird diese Spalte nicht in der Oberfläche angezeigt, sondern sie wird mittels ‚header stamping' ausgeblendet.
2. Für die Aggregation können im Bookmark Funktionen wie NONE, AVG oder SUM definiert werden. Für die Aggregation NONE verhält sich die Piviot-Tabelle merkwürdig, wenn nur genau eine Zeile zu aggregieren ist. In diesem Fall werden alle Daten der Zeile an die Aggregatsfunktion geliefert. Zur Lösung des Problems werden in diesem Fall, wenn die Aggregation NONE ausgewählt wurde, in der gruppierten Zeile die Werte deren Spalten mit NONE belegt. Über ‚cell stamping' werden diese Werte in der Oberfläche ausgeblendet.
3. Wenn ein angezeigter Aggregationspfad NULL Werte enthält, verhält sich die Pivot-Tabelle in einer nicht vorhersehbaren Weise. Um dieses Problem zu umgehen wird anstatt direkt auf die Werte mittels SQL zuzugreifen, immer noch die SQL Funktion NVL(TO_CHAR(col), '_NULL_') im SQL Statement verwendet. Damit wird statt des Wert NULL ein String '_NULL_' ausgegeben, der dann in der Oberfläche mittels ‚cell stamping' unterdrückt wird bzw. als leer angezeigt wird.

Ablauf einer Erzeugung einer dynamischen Pivot-Tabelle
Nachfolgenden wird kurz beschrieben wie beim Wechsel eines Bookmark eine neue Pivot-Tabelle erzeugt und angezeigt wird. Abb. 1 zeigt im oberen Drittel einen Teil der Ausgangsseite (ohne die zugehörige Tabelle), der blaue Pfeil deutet den Wechsel des Bookmark an. Unterhalb des blauen Pfeil ist das Resultat, also die Anzeige des neuen Bookmark, dargestellt.

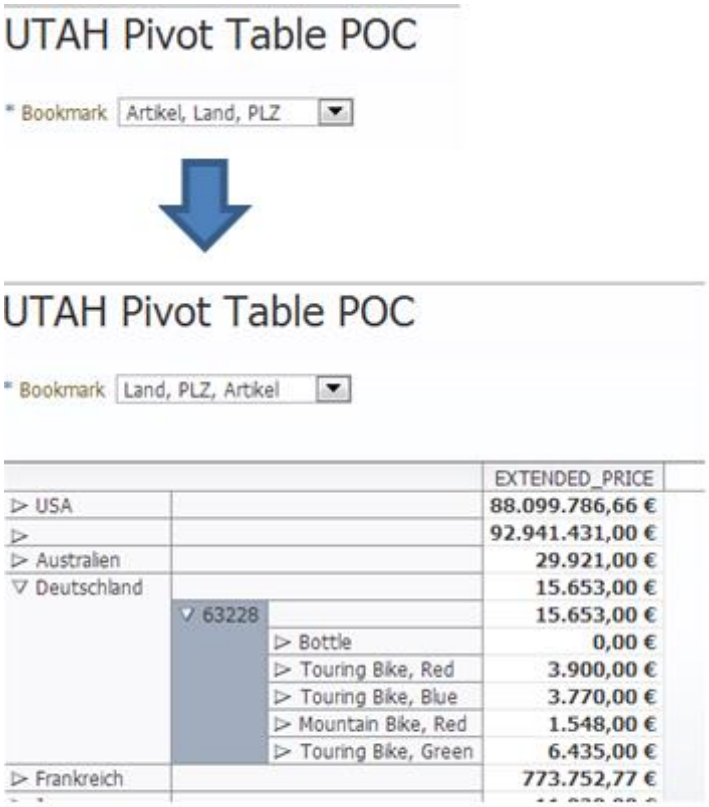

Abb. 1: Wechsel eines Bookmark in der Oberfläche

1. Der Wechsel des Bookmark löst einen Event aus, der prüft, ob der ausgewählte Bookmark bereits verwendet wurde. Ist dies der Fall wird auf diese Daten zurückgegriffen und mit Punkt 5 die Verarbeitung abgeschlossen.
2. Über eine Methode im ApplicationModule wird ein neues ‚View Definition Object' (ViewDef) angelegt und mit den Daten aus dem zum Bookmark gehörenden BookmarkEntry (BME) geladen. Aus diesen BME Daten wird ein SQL Statement generiert (Abb. 2)
3. Dieses SQL-Statement wird in ein ViewDef eingetragen, aus dem dann das ViewObject für die Pivot-Tabelle erzeugt ins Datenmodel des ApplicationModules eigetragen wird.

BME_COLUMN_NAME	BME_DISPLAY_POS	BME_SORT_POS	BME_IS_GROUPING_COL	BME_AGGREGATION_TYPE
COUNTRY_NAME	1	1	1	0
ZIP	2	2	1	0
ARTICLE_DESC	3	4	1	0
EXTENDED_PRICE	4	0	0	1

```
SELECT bme_column_name,
  bme_display_pos,
  bme_sort_pos,
  bme_is_grouping_col,
  bme_aggregation_type
FROM bookmark_entry
WHERE bme_bm_id = 29;
SELECT ROWNUM                                AS ROW_INTERNAL_UID ,
  NVL(TO_CHAR(COUNTRY_NAME), '_NULL_') AS COUNTRY_NAME ,
  NVL(TO_CHAR(ZIP), '_NULL_')          AS ZIP ,
  NVL(TO_CHAR(ARTICLE_DESC), '_NULL_') AS ARTICLE_DESC ,
  EXTENDED_PRICE                       AS EXTENDED_PRICE
FROM ORDER_TABLE
ORDER BY EXTENDED_PRICE ASC ,
  COUNTRY_NAME ASC ,
  ZIP ASC ,
  ARTICLE_DESC DESC
```

Abb. 2: Ein BookmarkEntry wird zu SQL Statement verarbeitet

4. Im ViewController wird nun mit diesem ViewObject ein Pivot-Tablen-Binding erzeugt und ein PivotTableIterator angelegt. Dazu wird erneut der BME verwendet, der die Informationen über den ‚Drill Path' (die Reihenfolge der zu gruppierenden Attribute) und die Aggregation enthält (Abb. 3).

```
<pivotTable IterBinding="PivotTableViewIterator_29" id="PivotTableModel_29"
            xmlns="http://xmlns.oracle.com/adfm/dvt">
  <pivotTableDataMap>
    <rows>
      <item value="COUNTRY_NAME"/>
    </rows>
    <drills type="INSERT"/>
    <hierarchies>
      <item value="COUNTRY_NAME" location="BEFORE">
        <child value="ZIP" label="ZIP"/>
      </item>
      <item value="ZIP" location="BEFORE">
        <child value="ARTICLE_DESC" label="ARTICLE_DESC"/>
      </item>
      <item value="ARTICLE_DESC" location="BEFORE">
        <child value="ROW_INTERNAL_UID" label="ROW_INTERNAL_UID"/>
      </item>
    </hierarchies>
    <columns>
      <data aggregateDuplicates="true" defaultAggregateType="NONE">
        <item value="EXTENDED_PRICE" aggregateType="SUM"/>
      </data>
    </columns>
    <pages/>
  </pivotTableDataMap>
</pivotTable>
```

Abb. 3: Aus dem BookmarkEntry erstelltes Pivot-Tabellen-Binding

5. Abschließend wird die Pivot-Tabelle mittels eines ‚refresh' aktualisiert und die Daten in der Oberfläche angezeigt.

Diese Vorgehensweise lässt sich natürlich auch auf eigene Tabellenstrukturen anwenden. Nicht verschwiegen werden soll, dass dazu allerdings der Aufruf von internen APIs notwendig ist. Da es sich bei den verwendeten Methoden aber um APIs handelt, die sehr dicht an den öffentlich zur Verfügung stehenden handelt, sollte die Nutzung in Ordnung gehen (oder gegebenenfalls leicht an deren Änderung angepasst werden können).

Vor Einsatz einer solchen Lösung in einer Produktionsumgebung sollte mittels eines ‚Service Request' (SR) eine Veröffentlichung der verwendeten internen APIs nachgefragt werden.

Referenzen

Oracle® Fusion Middleware Web User Interface Developer's Guide for Oracle Application Development Framework:
http://docs.oracle.com/cd/E23943_01/web.1111/b31973/dv_crosstab.htm#ADFUI11818

Rich Client Demo:
http://jdevadf.oracle.com/adf-richclient-demo/faces/components/pivotTable.jspx

Shay Shmeltzer: "Changing the Aggregation of an ADF Pivot Table":
https://blogs.oracle.com/shay/entry/changing_the_aggregation_of_an

Kontaktadresse:

Timo Hahn
Virtual7 GmbH
Zeppelin Straße 2
D-76185 Karlsruhe

Telefon: +49 (721) 619 017 59
Fax: +49 (721) 619 017 29
E-Mail hahn@virtual7.de
Internet: www.virtual7.de