

Ein Reporting Service mit APEX und BI Publisher

Perry Pakull
Trivadis AG
Glattbrugg

Schlüsselworte

Oracle Application Express, Oracle BI Publisher, Template Builder für Word, Reporting.

Einleitung

Obwohl Oracle Application Express und Oracle BI Publisher integrierte Lösungen für die Parametereingaben eines Reports enthalten, sind komplexe Anforderungen weder mit APEX noch mit BI Publisher allein zu erfüllen. Erst die Kombination der beiden Produkte ermöglicht die Entwicklung einer geeigneten Reporting Applikation, um anwenderfreundliche Eingabeformulare für den Aufruf der BI Publisher Reports umzusetzen. Der Vortrag vermittelt Projekterfahrungen, die bei der Entwicklung der Reporting Service Applikation gesammelt wurden. Dazu gehören APEX Formular Entwicklung, BI Publisher Web Service Integration und die Erstellung von BI Publisher Berichten mit dem Template Builder für Word.

Das Projekt

Das Projekt hat eine längere Vorgeschichte. Die Zielsetzung im ersten Projekt ist die Evaluierung der aktuellen Oracle Technologien Application Express und BI Publisher. Zur Überprüfung der technischen Machbarkeit wird eine kleine Beispielapplikation in der Umgebung des Kunden und mit den Daten des Kunden erstellt. Die APEX Applikation besteht aus einem einfachen Formular zur Eingabe von Parametern, die als Suchkriterien in der Datenbank gespeichert werden. Auf den Geschäftsdaten wird eine View angelegt, um die Daten für den Report abfragen zu können. Das Datenmodell des BI Publisher Berichts verwendet diese View direkt und setzt die Suchkriterien als Parameter ein. Das Layout ist eine einfache tabellarische Liste mit Kopf und Fußzeile. Der Aufruf des Berichts aus APEX heraus erfolgt mit URL-Parametern. Der BI Publisher erzeugt ein PDF-Dokument, das im Browser des Anwenders angezeigt wird.

Das Ergebnis ist überzeugend und so wird ein weiteres Projekt aufgesetzt, um einen kompletten Prototyp zu erstellen. Der Prototyp soll zeigen, dass die bestehenden Kundenanforderungen mit diesen Technologien realisierbar sind. Die Funktionen einer bereits existierenden Reporting Lösung sind die Grundlage für die Anforderungen an den Prototyp. Die Parametereingabe in der APEX Applikation wird erweitert. Die Suchkriterien werden nach der Eingabe strukturiert als Auftrag mit Parametern und definierten Ausgaben in der Datenbank abgelegt. Die benötigten Geschäftsdaten für den Report stellt ein PL/SQL Package bereit. Die Daten werden in einer eigenen Tabelle für den BI Publisher Bericht abgespeichert. Das BI Publisher Datenmodell greift auf diese Tabelle zu und der Bericht kann mit verschiedenen Layouts aufgerufen werden.

Der Prototyp und die darin enthaltenen Möglichkeiten überzeugen alle Beteiligten, so dass der Auftrag für die Realisierung eines kompletten Systems erteilt wird. Die Komplexität der Kundenanforderungen im Bereich Parametereingabe und die Komplexität des Datenmodells der Geschäftsdaten zeigen aber auch, dass der Prototyp noch erweitert werden muss.

Ein Reporting Service

Ein Report benötigt in der Regel ein vorgelagertes Formular zur Eingabe der definierten Parameter. Reporting Systeme wie der BI Publisher bieten dem Anwender die Möglichkeit Werte für die Parameter direkt einzugeben oder aus Wertelisten auszuwählen. Je mehr Geschäftslogik und Unterstützung für den Anwender hier gefordert ist, desto weniger eignen sich diese integrierten Parametereingabeformulare. Also wird die Parametereingabe in einer vorgeschalteten Applikation

realisiert, die entsprechende Möglichkeiten für die Umsetzung von Geschäftslogik und Anwenderunterstützung bietet. An dieser Stelle soll der Begriff Reporting Service genauer betrachtet werden. Die Zusammenstellung der Themen in Abbildung 1 verdeutlichen die Kundenanforderungen an die Parametereingabe und die Ausgabedefinitionen.

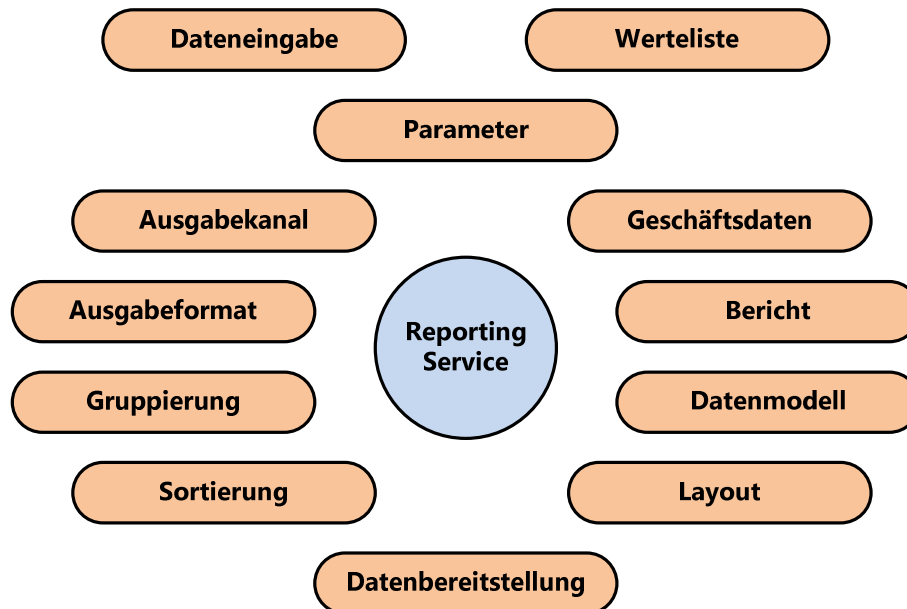


Abb. 1: Reporting Service

Zu den Anforderungen bei der Anwenderunterstützung gehören Assistenten bei der Dateneingabe wie zum Beispiel Kalender und Wertelisten. Bei vielen Wertelisten müssen mehrere Werte auswählbar sein und der Anwender muss sehen können, welche Optionen er ausgewählt hat. Der Inhalt von Wertelisten muss in Abhängigkeit von anderen Parametern dynamisch angepasst werden können. Bei solchen Abhängigkeiten beginnt der Bereich der Geschäftslogik. Die Eingabe von Parametern kann erforderlich oder optional sein. Parameter sind nicht immer als direkte Eingabewerte für den Report geeignet, sondern bilden nur Optionen, die dann in einen oder mehrere interne Parameterwerte umgesetzt werden.

Ein Reporting Service bietet nicht nur optimale Unterstützung bei der Parametereingabe, sondern auch bei der Definition der Auswertungsergebnisse. Bei den Auswertungen kann der Anwender zwischen verschiedenen Darstellungen, Gruppierungen, Sortierungen und Formaten wählen. Natürlich kann er auch festlegen, wo die Auswertung erstellt wird und in welchem Format. Mögliche Ausgabekanäle sind der Bildschirm, das Dateisystem oder der Anhang einer E-Mail. Mögliche Ausgabeformate sind PDF, CSV, Excel oder XML. Auch diese Parameter haben Abhängigkeiten.

Die Anwendung der eingegebenen Parameter auf die Geschäftsdaten zur Selektierung der Reportdaten ist dann ein weiterer Bereich, der zum Reporting Service gehört. Die Datenbereitstellung bezieht sich hier auf die Reportdaten, die dann im Bericht aufbereitet werden und nicht auf die erzeugten Auswertungsergebnisse.

Die Berichte haben ein eigenes Datenmodell, um die bereitgestellten Reportdaten zu selektieren und diverse Layouts für das Rendering der Daten.

Lösungsansatz und System Architektur

Die eingesetzten Technologien APEX und BI Publisher eignen sich gut, um die zuvor beschriebenen Themen und Anforderungen abzubilden. Der Lösungsansatz für die Reporting Applikation besteht aus drei getrennten Komponenten. Eine APEX Applikation für die Parametereingabe, PL/SQL Packages

für die Datenbereitstellung und BI Publisher Berichte mit SQL-Datenzugriff und RTF-Layouts für die Auswertungen. Die Komponenten müssen möglichst unabhängig voneinander eingesetzt und entwickelt werden können. Abbildung 2 gibt eine Übersicht der implementierten System Architektur mit Ergänzungen verschiedener logischer Komponenten.

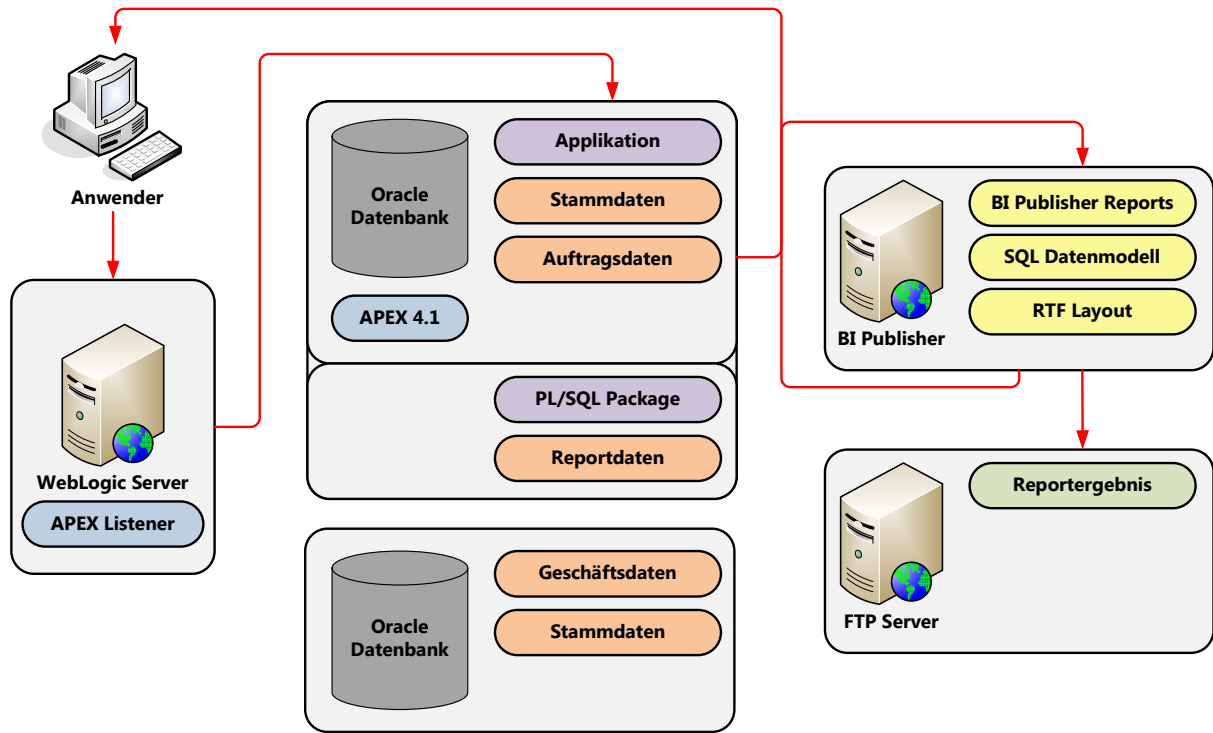


Abb. 2: System Architektur

Die APEX Applikation

Der Lösungsansatz mit möglichst modularen und unabhängigen Komponenten gilt auch für die einzelnen Reporting Services. Die Reporting Applikation hat nur wenige zentrale Seiten. Dazu gehören die Anmeldung, die Mandantenauswahl und eine Übersicht der verfügbaren Reporting Services und der ausgeführten Aufträge.

Listenauswahl			Aufträge						
Gruppe	Service	Beschreibung	Anwender	Auftrag	Gruppe	Mandant	Service	Erstellt am	Status
Gruppe 1	Service 1	Beschreibung 1	PEP	1	Gruppe 4	XXXXX	Service 7	15.08.2012	Reportdaten erstellt
	Service 4	Beschreibung 4							
Gruppe 2	Service 3	Beschreibung 3							
	Service 5	Beschreibung 5							
Gruppe 4	Service 15	Beschreibung 15							
	Service 25	Beschreibung 25							
	Service 7	Beschreibung 7							

Abb. 3: Übersicht Reporting Services und ausgeführte Aufträge

Auf dieser zentralen Seite, die aus zwei einfachen APEX Reports besteht, wählt der Anwender die gewünschten Services aus und wird zur entsprechenden Parametereingabe geführt. Der zweite APEX

Report zeigt die ausgeführten Aufträge und verzweigt auf eine weitere zentrale Seite, zur Anzeige der Auswertungsergebnisse.

Jeder Reporting Service ist ebenfalls unabhängig aufgebaut und besteht aus einer eigenen Seite für die Parametereingabe und einer eigenen Bestätigungsseite. Die Entscheidung für diese Lösung wurde zu Beginn des Projektes getroffen. Die Reporting Applikation besteht zurzeit aus 25 verschiedenen Services mit insgesamt ca. 250 Parametern. Einige Services verwenden die gleichen Parameter bzw. Parametergruppen. Solche Parametergruppen sind jeweils in einer Region auf der Seite zur Parametereingabe angeordnet. APEX unterstützt leider keine HTML-Fragmente bei den Shared Components, die es ermöglichen würden Teile einer APEX Page mehrfach zu verwenden. Die Entwicklung der Parametereingaben mit einer einzigen zentralen Seite für die alle Reporting Services wurde verworfen, weil die Seite nur von einem Entwickler bearbeitet werden kann.

Die Erfassung der Ausgabedefinitionen wiederum ist eine zentrale Seite mit einer Tabular Form. Die Seite wird mit service-spezifischen Stammdaten gesteuert. Der Anwender wird von der Eingabeseite der Parameter über die Ausgabedefinitionen zu einer finalen Bestätigungsseite geführt, die nochmals alle Eingaben anzeigt und durch Hinweise ergänzt.

Die erfassten Daten werden bis dahin in APEX Collections temporär im Session Context gehalten. Sobald der Anwender die Eingaben bestätigt werden die Parametereingaben als Auftrag in der Datenbank gespeichert. Das Datenmodell für die Auftragsdaten in Abbildung 4 ist generisch, jeder Auftragsparameter wird mit Name und Wert als eigener Datensatz gespeichert.

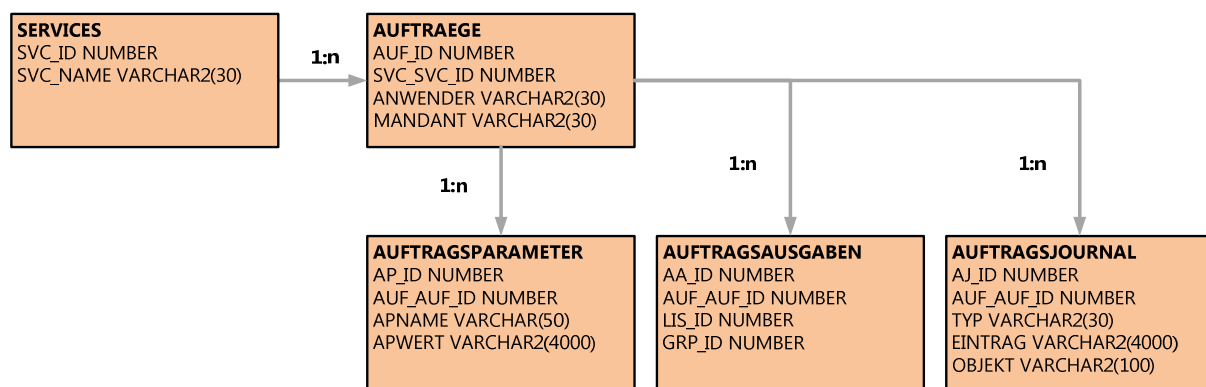


Abb. 4: Datenmodell Auftragsdaten

Die Anbindung der APEX Applikation an die Datenbereitstellung ist direkt und synchron. Das definierte PL/SQL Package für den Reporting Service wird aufgerufen.

Sobald die Datenbereitstellung abgeschlossen ist, sieht der Anwender den Auftrag in der Auftragsübersicht und kann auf die zentrale Seite zur Anzeige der Ausgaben navigieren. Jetzt erfolgt der Aufruf der BI Publisher Berichte für eine Anzeige der Auswertungen am Bildschirm.

Datenbereitstellung mit PL/SQL

Die Datenbereitstellung ist keine vollständig unabhängige technische Komponente, sondern besteht aus PL/SQL Packages in der Datenbank. Sie ist eher als logische Schicht zu verstehen. Die Packages sind unabhängig von der APEX Applikation. Einziger Parameter ist die Auftragsnummer. Die Auftragsdaten wurden vorher in der Datenbank gespeichert. Jeder Reporting Service hat ein eigenes Package mit eigener Logik zur Anwendung der Eingabeparameter. Für die Selektierung der Geschäftsdaten werden die Auftragsparameter gelesen und in ein dynamisches SQL-Statement umgesetzt. Das SELECT-Statement wird über einen REF-Cursor ausgeführt und die Daten in eine Tabelle geschrieben. Jeder Reporting Service hat eigene Datenstrukturen für die Reportdaten.

Die PL/SQL Packages verwenden eine einheitliche Struktur und haben zwei sichtbare Methoden, eine Funktion für die Erzeugung des SELECT-Statements und eine Prozedur zur Bereitstellung der Reportdaten. Der nachfolgende PL/SQL Code ist ein Beispiel zur Veranschaulichung der einheitlichen Strukturen.

```
package rsl
is
    function createreportsqlstatement(p_auf_id in number) return varchar2;
    procedure createreportdaten(p_auf_id in number);
end package;
```

Berichte mit BI Publisher

Die konsequente Trennung von Datenmodell und Layout bei den BI Publisher Berichten hat Vorteile und Nachteile. Ein Datenmodell kann für mehrere Layouts verwendet werden, ein Layout kann verschiedene Datenmodelle nutzen. Die Entwicklung der beiden Komponenten ist in diesem Projekt ebenfalls getrennt. Der Bericht und das Datenmodell werden auf dem Server angelegt und entwickelt. Die RTF-Layouts werden mit dem Template Builder für Word erstellt. Die Anbindung der BI Publisher Berichte ist genauso einfach gehalten wie bei der Datenbereitstellung. Der einzige Parameter der Datenmodelle ist die Auftragsnummer. Die Entwicklung der Berichte ist insgesamt aufwendig, da Änderungen immer an mehreren Stellen mit verschiedenen Programmen vorgenommen werden müssen. Um neue Felder in einen Bericht aufzunehmen muss das PL/SQL Package für die Datenbereitstellung und die Tabelle für die Reportdaten angepasst werden. Danach kann das Datenmodell des BI Publisher Berichts angepasst werden, um anschließend neue XML-Beispieldaten zu generieren. Jetzt erst kann die Aktualisierung des RTF-Layouts mit dem Template Builder für Word vorgenommen werden.

Gruppierungs- und Sortierungsfelder der Berichte werden durch den Anwender vorgegeben. Für den Reporting Service ist eine jeweils Liste mit den verschiedenen Optionen definiert. Die Umsetzung erfolgt in den Datenmodellen. Die gewählte Option wird abgefragt und der Wert des entsprechenden Feldes selektiert. Im Datenmodell hat dieses Feld immer den gleichen Namen. Das Layout verwendet für die Gruppierung und Sortierung immer dieses Feld. Formatierungen werden ebenfalls im Datenmodell vorgenommen. Das Layout enthält so wenig Logik wie möglich.

Für den Prototyp wurden die ersten einfachen Layouts noch mit dem BI Publisher Layout Editor erstellt. Der Template Builder für Word bietet jedoch mehr Funktionalität und eine bessere Unterstützung bei der Entwicklung durch die Assistenten. Der erste Entwurf wird mit den Assistenten erzeugt und anschließend angepasst. Dabei werden die Formularfelder für die Wiederholungsgruppen automatisch angelegt. Die Positionierung der Daten erfolgt immer über Word Tabellen. Erweiterungen erfordern XPATH, XSL oder XSL-FO Kenntnisse. Syntaxfehler sind nur schwer zu finden, auch wenn eine Validierungsroutine zur Verfügung steht.

Schnittstellen

Die Reporting Applikation hat zwei Schnittstellen. Der Aufruf der Applikation wird aus einer anderen Dialog Applikation per URL gestartet. Der Anwender ist dort bereits angemeldet. Beim Aufruf wird ein zusätzlicher Parameter übergeben. Die APEX Applikation ruft einen Web Service mit diesem Parameter auf, der den Anwender verifiziert und autorisiert. Die Anmeldung in APEX erfolgt automatisch mit einem technischen APEX Benutzer. Der Web Service liefert die Anwenderdaten einschließlich der Berechtigungen für Reporting Service Gruppen und Mandanten.

Die zweite Schnittstelle integriert die APEX Applikation mit dem Oracle BI Publisher. Der Aufruf der Berichte erfolgt über die Web Service Schnittstelle des BI Publishers und wird in der Datenbank realisiert. Diese Schnittstelle wird benötigt, um eine zusätzliche Anmeldung auf dem BI Publisher zu vermeiden. Ein weiterer Vorteil ist die Unabhängigkeit von der APEX Applikation. Aufträge können so auch über ein Scheduling System aufgerufen werden.

Erfahrungen und Fazit

Oracle Application Express und Oracle BI Publisher vermitteln leicht das Gefühl, dass Applikationen schnell und einfach realisiert werden können. Für Oracle Application Express gilt das nur, solange vorhandene Standardelemente ohne Anpassungen verwendet werden. Erweiterungen mit jQuery und JavaScript für dynamische Elemente, Anpassungen der Templates für die Oberfläche und spezielle Validierungen erfordern deutlich höheren Aufwand. Die Kontrolle und Pflege von mehr als 50 APEX Seiten ist ebenfalls aufwendig. Die Anzahl der Seiten wird aber noch weiter steigen. Doch auch dafür bietet APEX mit dem Repository und den APIs bereits heute Möglichkeiten, diese Schwierigkeiten in den Griff zu bekommen. Die Unterstützung durch integrierte Tools für die Entwicklung und Administration ist gut. Die APEX Entwicklungsumgebung bietet sehr viele Elemente und Funktionen für die Gestaltung der Oberflächen.

Umsteiger von Oracle Reports werden die integrierte Entwicklungsumgebung Report Builder beim BI Publisher schmerzlich vermissen. Die Erstellung der RTF-Layouts mit dem Template Builder für Word ist am Anfang gewöhnungsbedürftig. Die Assistenten sind hervorragend, aber Anpassungen oder Erweiterungen der XPATH, XSL oder XSL-FO Syntax sind schwer und erfordern ebenfalls deutlich höheren Aufwand. Die Möglichkeiten Layouts zu gestalten sind insgesamt beeindruckend. Im Projekt konnte bisher nur ein kleiner Teil der verfügbaren Elemente für Formulare, Tabellen und Grafiken genutzt werden.

Die ersten Schritte sind leicht, Erweiterungen erfordern Fachwissen und praktische Erfahrung. Diese Oracle Technologien sind modern und zeitgemäß, sehr flexibel, reichhaltig an Funktionen und vor allem offen. Stabilität, Performance und Skalierbarkeit sind gegeben. Die Software wird ständig weiter entwickelt und verbessert. Die Reporting Lösung bietet dem Kunden eine solide Basis für die Zukunft. Es lohnt sich also hier zu investieren.

Kontaktadresse

Perry Pakull
Trivadis AG
Europa-Strasse 5
CH-8152 Glattbrugg

Telefon: +41-44-808 70 20
Fax: +41-44-808 70 21
E-Mail: perry.pakull@trivadis.com
Internet: www.trivadis.com