

Die Forderungen der Fachabteilungen nach Prozess-Abbildung in IT-Systemen haben in den letzten Jahren stark zugenommen. Dabei steht nicht nur die schnelle Umsetzung, sondern auch die Flexibilität für nachträgliche Anpassungen im Fokus. Apex ermöglicht es, genau diese Anforderungen umzusetzen. Allerdings sollte auch dort nie der Grundsatz der Einheitlichkeit aus Sicht des Anwenders verloren gehen.

Standardisierung mit Apex

Thomas Zielbauer, MAN Truck & Bus AG

Findet der Anwender in neuen Applikationen eine gewohnte Benutzerführung, reduziert sich der Schulungsbedarf beziehungsweise die Schulungsintensität erheblich. Es liegt also nahe, über eine Standardisierung der Anwendungen in Apex nachzudenken. In Apex gibt es mehrere Möglichkeiten diesen Ansatz anzugehen.

Wir könnten die Anforderungen des Fachbereichs in jeweils separaten Anwendungen mit demselben Look & Feel (Theme) umsetzen, was es uns ermöglicht, ein einheitliches Design mit jedoch getrennten Funktionalitäten an den Kunden zurückzugeben. Oder aber wir verfolgen den Ansatz der absoluten Standardisierung und integrieren verschiedene Projekte in eine Art „Framework-Anwendung“. Das hätte den Vorteil, dass Benutzer mit mehreren Berechtigungen sich nicht durch unterschiedliche Anwendungen bewegen müssten. Ebenso könnte standardisierte Funktionalität auf alle dort integrierten Teilsysteme vererbt werden. Natürlich würde sich die übergreifende Systemanmeldung auch mit einem Single-Sign-on-Ansatz decken, jedoch bringt der Framework-Ansatz Vorteile mit sich, die sich in der Praxis mit Oracle Apex als eindeutig interessanter herausstellen.

Dieser Ansatz der UI-Standardisierung brachte das Unternehmen des Autors vor einigen Jahren auf die Idee, mit Apex eine Framework-Applikation zu programmieren, also eine Anwendung, in der unterschiedlichste Projekte und Kundenanforderungen umgesetzt werden können. Eine Anwendung, die es auch den Entwicklern vereinfachen würde, schnell und flexibel auf Kunden-Anforderungen zu reagieren – aber

dabei auch niemals den definierten Standard verlassen zu müssen. Sozusagen Individualität im Standard.

Konzept-Erstellung

Diese und weitere Fragen prägten das Konzept grundlegend: Wie viele Variablen braucht der Fachbereich tatsächlich und wie soll der Zugriff darauf erfolgen? Wie kann eine Navigation umgesetzt werden, die es uns erlaubt zu bestimmen, was (beispielsweise Gemeinkosten oder Qualitätsdaten) über wen (entspricht dem Auswertungsobjekt wie etwa Kostenstelle oder Abteilung) angesehen werden kann? Mithilfe welcher Komponenten wird sich das alles umsetzen lassen?

Das endgültige Konzept gliederte sich in vier funktionale Bereiche (siehe Abbildung 1):

1. Variablen, die mithilfe einer jQuery-Funktion ein- und ausblendbar sein sollen
2. Eine Navigationsleiste, die ein jQuery-basiertes Dropdown-Menü enthalten soll (dynamisch, abhän-

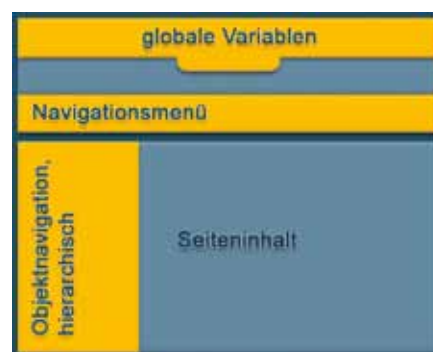


Abbildung 1: Das strukturierte, visuelle Konzept

gig von der Rollenzuweisung des angemeldeten Benutzers)

3. Die Auswertungsobjekt-Navigation, die die Organisationsstruktur der MAN Truck & Bus AG als Baumstruktur darstellt und erst bei Änderungsbedarf eingeblendet werden soll
4. Ausreichend Platz für den eigentlichen Inhalt der aktuellen Apex-Seite

Umsetzung des Konzepts in Apex

Der Schlüssel ist ein eigenes Theme und natürlich die Globalisierungsmöglichkeit in Apex mithilfe der Seite 0. Inhalte, die auf Seite 0 in Apex erstellt werden, projizieren sich selbst auf jede geöffnete Seite. Regionen, die beispielsweise im Seiten-Template definiert sind, lassen sich auf Seite 0 mit Leben füllen. Auf diese Weise entstehen das individuelle Benutzermenü, die Kostenstellen-Hierarchie als Baumstruktur und die „globalen“ Variablen auf jeder Seite in der Anwendung. Wie die Abbildungen 2 und 3 zeigen, ist das gesamte visuelle Konzept im minimierten Zustand sehr kompakt gehalten. Dadurch gewinnt man Platz für den eigentlichen Seiteninhalt.

Umgesetzt wurden die ein-/ausblendbaren Ebenen als „<div> HTML“-Elemente. Die Cascading-Stylesheet-Eigenschaft (CSS) „display“ wechselt durch das Klick-Ereignis bei diesen Elementen per JavaScript von „none“ auf „block“ und umgekehrt. Dafür lässt sich wunderbar die in jQuery integrierte Funktion „toggle“ nutzen, die mit oder ohne Animation verwendet werden kann. Die HTML-Elemente selbst wurden per CSS absolut positioniert, um nicht durch die weiteren HTML-

Da es das Ziel war, unternehmensweit einige wenige Kennzahlen aufzubereiten, die lediglich zur stärkeren Identifizierung mit dem Unternehmen beitragen sollten, fiel die Entscheidung auf animierte Business-Objects-Xcelsius-Diagramme. Diese werden über eine sich täglich ändernde XML-Datei automatisiert mit Daten befüllt. Per JavaScript und CSS wurde eine Art Diagrammwürfel in die Einstiegsseite integriert. Eine Seite des Würfels enthält weitere Navigationsmöglichkeiten in verschiedene Richtungen. Damit können unterschiedliche Diagramme animiert gewechselt werden. Mit dieser Technik konnten mehrere Diagramme auf dem Platz von einem in die Startseite integriert werden. Selbstverständlich enthält diese Einstiegsseite aber auch funktionale Bereiche wie beispielsweise eine Übersicht darüber, welche Daten sich derzeit im System befinden, oder auch eine Möglichkeit zur Passwortänderung.

Berechtigungskonzept der Anwendung

Zur Steuerung der Rechte empfiehlt es sich, den Apex-Standard zu benutzen. Dieser ist vielseitig anwendbar und sollte die meisten Anforderungen erfüllen. Die Überprüfung der einzel-

nen Rollen wurde sowohl auf der Seite selbst als auch in unserem Navigationsmenü platziert. Damit lässt sich nicht nur die Anwendung absichern, sondern auch das Menü individuell für den Anwender erzeugen. Diese Form der Personalisierung trägt erheblich zur Übersichtlichkeit der Anwendung bei. Die Überprüfung der Berechtigung kann ohne Bedenken auf einmal je Session reduziert werden. Gerade Anwendungen mit vielen Navigationszugriffen oder Seitenwechseln profitieren hiervon. Für den Anwender entsteht dadurch lediglich ein annehmbarer Nachteil: Wurde einem Anwender eine neue Rolle zugesprochen, wird diese erst nach erneuter Anmeldung aktiv (siehe Abbildung 5).

Rückwirkende Betrachtung

Die Framework-Anwendung ist seit knapp zwei Jahren im produktiven Betrieb mit vereinzelt Design-Anpassungen. Einige Projekte brachten neue Funktionen mit sich, die in den Framework-Standard integriert werden konnten. Die Anzahl der darin umgesetzten Projekte aus den Fachabteilungen nahm konsequent zu. Das Wachstum der Anwendung erfolgte also mehrdimensional. Aus Sicht des An-

wenders trat ein, was als Ziel definiert war: Sehr hohe Akzeptanz und nahezu kein Schulungsbedarf. Aus Entwicklersicht wurden die Erwartungen ebenso erfüllt: Neue Anforderungen können sehr zielgerichtet umgesetzt werden und im Standard integrierte Funktionen wie beispielsweise der Warnhinweis, dass eine Seite nicht ausreichend dokumentiert wurde, tragen erheblich zur Qualität der Anwendung bei (siehe Abbildung 6).

Integration eines Kundenprojekts in die Framework-Applikation

Ein interessantes Projekt aus dem Fachbereich war die Analyse von Qualitätsdaten. Als Datenquelle diente ein auf einer Oracle-Datenbank basierendes Data Warehouse. Das System sollte in der Lage sein, Produktionsfehlermeldungen zu überwachen und zu katalogisieren. Sobald aus der Datenmenge ersichtlich wurde, dass etwa ein Problem bei einer Konstruktionsgruppe vorlag, sollte der Prozess zum Abstellen des Fehlers ausgelöst werden. Die Protokollierung des Abstellprozesses sollte durch das System vollzogen werden. Während des Abstellprozesses trat ein zeitlich definiertes Regelwerk in Kraft, das den Anwender anhand festgelegter Standards durch den Prozess begleitete. Außerdem sollte es dem Kunden möglich sein, die einzelnen Prozessschritte mit Kollegen zu teilen. Das Hauptaugenmerk lag damit von Anfang an auf optimierter Analysefunktionalität, softwareseitiger Prozess-Unterstützung sowie einer klaren Kommunikation des Systems mit dem Anwender.

Die Katalogisierung des Datenbestandes erfolgte mithilfe von Oracle-Text. Um eine komfortable Suche mit bestmöglichen Suchergebnissen zu realisieren, wurde auf Schlüsselmerkmale ein Textindex gelegt. Der Textindex, der mit dem nachfolgenden Befehl erzeugt wurde, gibt bei dessen Verwendung eine Suchergebnisbewertung (score) zurück.

```
create index IDX_INDEXNAME on
TABELLE (INDEXSPALTE) indextype
is ctxsys.context
```

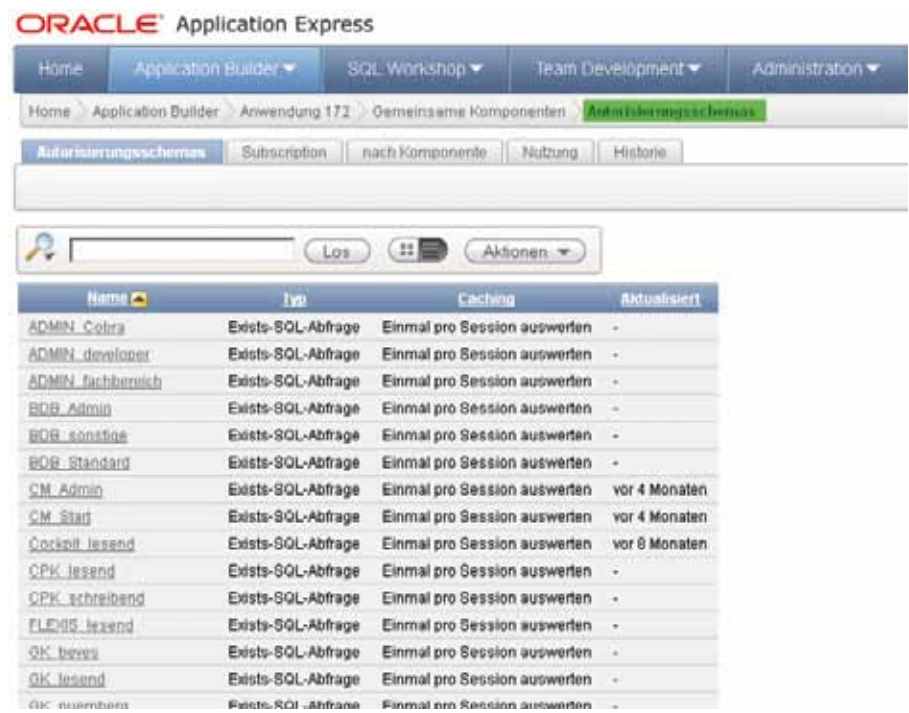


Abbildung 5: Autorisierungsschemata in Apex



Abbildung 6: Fehlerbehandlung serienmäßig



Abbildung 7: AnyChart5 bietet zahlreiche Einsatzmöglichkeiten

```

select
  SCORE(1) as Punkte,
  SPALTE_A,
  SPALTE_B,
  INDEXSPALTE
from TABELLE
where
  contains(INDEXSPALTE, '%' || :P1_Apex_TEXT_ELEMENT_SUCHE || '%', 1) > 0

```

Listing 1

Dieser „Score“ bewertet den Grad an Entsprechung des Ergebnisses zum Suchbegriff in Punkten. „Score“ lässt sich wie eine normale Spalte sortieren. Damit können die am ehesten dem Suchbegriff entsprechenden Ergebnisse zuerst aufgelistet werden. Eine Abfrage nach einem Suchbegriff erfolgt über den SQL-Befehl „contains“ (siehe Listing 1).

Nachdem die gezielte Navigation innerhalb der Daten mit der Umsetzung der Suchfunktion einen wichtigen Schritt nach vorn gemacht hatte, galt es, diesen sinnvoll in die Anwendung zu integrieren. An dieser Stelle nahm der Framework-Gedanke wieder konkrete Formen an: Die benötigten Regionen der Suchfunktion wurden auf Seite 0 gelegt. Allerdings wurde die Publizierung auf bestimmte Seiten beschränkt, da diese nur auf einigen benötigt war.

Die Suchfunktion wurde aufgrund der hohen Qualität der Ergebnisse in einige Berichte als Filtermöglichkeit aufgenommen. Ziel war es unter anderem, ein grafisches Monitoring eines eindeutigen Fehlerbildes zu ermöglichen.

Da die Suchfunktion so programmiert wurde, dass sie, sobald ein Wert selektiert wurde, auch ein konkretes Fehlerbild zurückgeben konnte, wurde dieses in Variablen auf Seite 0 abgelegt. Dadurch war das konkrete, ausgewählte Fehlerbild wiederum für die Filterung der Daten aus dem Data Warehouse griffbereit. Die Abfrage erfolgte also gezielt und damit äußerst performant. Nun war es möglich, auf

Verwalten Sie Ihre Requirements mit IBM Rational DOORS?
Haben Sie ein PLM System von Oracle im Einsatz?
Lassen Sie sich den Mehrwert durch eine Integration nicht entgehen!
Besuchen Sie uns auf der DOAG 2012 Applications.

usb GmbH • Betastr. 13a • 85774 Unterföhring • Telefon 089 / 99894283
http://www.usb-muc.de • info@usb-muc.de

ORACLE Gold Partner



Abbildung 8: Verwenden eines „Region Display Selectors“ für eine Region



Abbildung 9: Die Auswirkung der JavaScript-Anpassung

diese gefilterte Datenmenge ein Reporting nach Wunsch des Fachbereichs aufsetzen.

Apex bietet im Standard bereits eine breite Masse an Möglichkeiten zur Umsetzung eines grafischen Reporting. In Version 4.1 enthält Apex mit AnyChart 5 eine äußerst mächtige Komponente, um Flash-Diagramme dynamisch zu erzeugen. Für die Erstellung dieser ist es möglich, den integrierten Assistenten zu verwenden oder – sofern Kenntnisse über die AnyChart-5-XML-Syntax vorhanden sind – auch den Code beliebig selbst zu schreiben. Aufgrund der besseren Anpassungsmöglichkeiten fiel die Entscheidung, für das grafische Reporting die XML-Syntax selbst zu codieren und Apex lediglich das Füllen des Templates mit Daten zu überlassen (siehe Abbildung 7).

Da das Reporting so konstruiert wurde, dass der Fachbereich verschiedene Sichten auf eine gefilterte Menge an Basisdaten anwenden konnte, lag es nahe, die Datenmenge in eine Collection zu schreiben. Diese konnte nun von den unterschiedlichen Berichten einer Seite als Datenbasis angesprochen werden. Der Performance-Gewinn war dadurch enorm. Um die Seite aber nicht mit zu vielen Berichten zu überladen, wurde ein Standard-Fea-

```

/* ausblenden aller display-selector regionen */
apex.jquery(, #SHOW_ALL').hide();
apex.jquery(, #DISP_REG_1').hide();
apex.jquery(, #DISP_REG_2').hide();
/* einblenden von Region 1 */
apex.jquery(, #DISP_REG_1').show();
/* Initialisierung Region 1 */
apex.jquery(, li.apex-rds-first').removeClass(,apex-rds-selected");
apex.jquery(, li.apex-rds-first').hide();
apex.jquery(, a[href="#DISP_REG_1"]').closest(, li').
addClass(,apex-rds-selected");

```

Listing 2

ture in Apex verwendet: der „Region Display Selector“. Dieser ermöglicht es, bestimmte Regionen einer Seite ein- und ausblendbar zu machen. Haben wir beispielsweise zwei Regionen auf einer Seite und aktivieren für beide den „Region Display Selector“, kann man entweder Region 1, Region 2 oder alle Regionen zugleich anzeigen lassen. Zu beachten ist nur, dass auf der Apex-Seite eine neue Region vom Typ „Region Display Selector“ erstellt werden muss. Diese spezielle Region übernimmt die Steuerung der Sichtbarkeit der Regionen im User Interface. Da auch diese Region vollständig anpassbar ist, lässt sie sich per CSS wunderbar in eine Liste mit Registerkarten-Optik umwandeln.

Wie Abbildung 8 zeigt, wird die Region einfach per Einstellung ein-/ausblendbar gemacht. In der Abbildung ist aber auch das Feld „Static ID“ hervorgehoben. Diese wurde für unsere Regionen fest hinterlegt. Der Grund hierfür ist schnell erklärt: Im Apex-Standard öffnet sich eine Seite mit mehreren ein-/ausblendbaren Regionen immer im Modus „Show all“. Damit würden alle Diagramme beim Öffnen der Seite auf einmal sichtbar sein. Die Seite würde dadurch allerdings überladen wirken. Ziel war, dass zum einen der „Display Region Selector“-Eintrag „Show all“ komplett verschwindet und zum anderen, dass nur eine Region auf einmal angezeigt wird. Da die Region IDs fest definiert wurden, war es möglich, über folgenden JavaScript-Code (siehe Listing 2), der beim Laden der Seite ausgeführt wird, die Sichtbarkeit zu steuern (siehe Abbildung 9).

Mit dieser Methode fiel es nicht schwer, die Anforderungen des Fachbereichs gebündelt, aber dennoch übersichtlich auf einer Themenseite darzustellen. Der Anwender hatte hierbei lediglich eine einmalige, kurze Wartezeit für das Bereitstellen der Daten. Sobald diese geladen wurden, konnte ohne Ladezeiten zwischen den Sichten gewechselt und damit die Qualitätsdaten analysiert werden.

Fazit

Das kurz diskutierte Projekt ist ein gutes Beispiel für die erfolgreiche Integration einer Data-Warehouse-Analyse über eine standardisierte Apex-Anwendung. Es zeigt sehr deutlich, mit welchen Mitteln bereits innerhalb des Apex-Standards gearbeitet werden kann. Außerdem bietet es einen kleinen Einblick in die Möglichkeiten, die durch Anpassungen der Apex-Features entstehen können. Apex bietet nahezu keine Grenzen – auch nicht bei der Definition eigener Standards.

Thomas Zielbauer
MAN Truck & Bus AG
thomas.zielbauer@man.eu

