

APEX, Hilfe – gelungene Benutzerführung (fast) ohne Mehraufwand

Andreas Wismann
WHEN OTHERS
D-41564 Kaarst

Schlüsselworte

APEX, Benutzerführung, User Interface, Hilfe, HTML, jQuery, CSS, JSON

Einleitung

Mit Oracle Application Express erstellen Sie im Handumdrehen sehr ansehnliche und benutzungsfreundliche Web-Applikationen. APEX bietet von Haus aus eine gute Benutzerführung und auch eine vielfältige Unterstützung für Hilfetexte an. So können Sie beispielsweise Hilfetexte und/oder Kommentare direkt hinterlegen für

- Seiten
- Page Items (Eingabefelder), diese Texte werden beim Anklicken des Labels eingeblendet
- Prozesse (für Entwickler)

Dennoch vermisst man in umfangreichen APEX-Anwendungen bisweilen eine schlüssige Benutzerführung und ein Hilfesystem, das seinen Namen tatsächlich verdient. Nicht etwa, weil dies alles so schwierig zu implementieren wäre, sondern weil dazu keine zwingende Notwendigkeit seitens der Entwickler besteht. Diese Aspekte werden im Entwicklungszyklus häufig stiefmütterlich behandelt, entweder weil kein stimmiges Konzept vorhanden ist oder weil man annimmt, dass sich dadurch der Entwicklungsaufwand deutlich erhöht – ohne echten Nutzengewinn. Gerade für Applikationen, die im eigenen Unternehmen verwendet werden, gilt oft: Der Schuster hat die schlechtesten Schuhe.

APEX-Bordmittel sind ein sehr guter Ansatz

Es gibt dabei jedoch einige Einschränkungen. Als Beispiele:

- Hilfetexte sind grundsätzlich einsprachig, da sie deklarativ zum Objekt gehören
- Seiten-Hilfetexte werden von allein nicht angezeigt
- Der Entwickler muss in der Seite bzw. in den Objekteigenschaften ganz nach unten scrollen, um den Developer Comment zu lesen
- Für den Benutzer existiert standardmäßig zwar ein Handbuch, das aber nicht „online“, also kontextsensitiv mit den Standardkomponenten der APEX-Anwendung verknüpft ist.

Es geht im Vortrag also sowohl um die Anliegen von Benutzern wie die von Entwicklern.

Erster Test: Existieren in Ihrer Anwendung überhaupt nicht-triviale Seiten-Hilfetexte?

```
SELECT page_help_text
       , COUNT(*)
FROM (SELECT substr(to_char(help_text), 1, 50) AS page_help_text
      FROM apex_application_pages)
GROUP BY page_help_text
ORDER BY COUNT(*) DESC;
```

Das Ergebnis sieht, seien wir selbstkritisch, typischerweise so oder ähnlich aus:

	PAGE_HELP	COUNT(*)
1	No help is available for this page.	129
2		61
3	Für diese Seite ist keine Hilfe verfügbar.	2
4	This is the Home Page of the Sample Database Appli ...	1
5	Help for this page	1

Das SQL ist übrigens deshalb etwas umständlich formuliert, weil die Spalte HELP_TEXT vom Typ CLOB ist – aha, man kann also beliebig umfangreiche Hilfetexte in APEX hinterlegen – Pluspunkt! „No help is available for this page.“ lautet übrigens (im englischen APEX) der vorgegebene Hilfetext für jede neu erstellte APEX-Seite, wenn der Entwickler ihn nicht ausdrücklich löscht oder durch Sinnvolleres ersetzt.

Nächster Test: Haben Ihre Page Items anklickbare Labels, die dem Benutzer eine verfügbare Hilfestellung suggerieren, können dann aber gar keine Hilfe anbieten?

```
SELECT *
FROM apex_application_page_items
WHERE (item_label_template LIKE '%help%'           -- englisches APEX
      OR lower(item_label_template) like '%hilfe%') -- deutsches APEX
AND item_help_text IS NULL;
```

Lassen Sie sich überraschen – hier tauchen vermutlich etliche Beispiele für „sub-optimale Benutzerführung“ auf... Denn das standardmäßige Label-Template für ein neu erstelltes Page Item verfügt über einen anklickbaren Link, der einen Dialog mit kontextsensitiver Hilfe einblendet. Jedoch hilft das dem Anwender nur, wenn der Entwickler im Feld Item Help etwas Hilfreiches hinterlegt hat, ansonsten bleibt der Link zwar immer noch anklickbar, zeigt aber nur einen leeren Dialog an. Nach mehreren erfolglosen Versuchen dürfte der Anwender anschließend wohl dauerhaft die Lust verloren haben, noch irgendwelche der vermeintlichen Hilfe-Links anzuklicken.

Mit den Erkenntnissen des obigen SQL-Statements haben Sie zwei Möglichkeiten zur Abhilfe:

- Sie können diesen Items ein Label Template zuordnen, das keine Hilfe anbietet, oder
- Sie kümmern sich um die Hilfe.

Ein weiteres APEX-Bordmittel, das Sie kennen sollten

Das APEX-Feature „User Interface Defaults“ wird von vielen Entwicklerteams nicht genutzt. Doch gerade damit lässt sich einiges an Aufwand einsparen, um Ihren APEX-Anwendungen einen konsistenten Hilfe-Layer mitzugeben. Wenn Sie User Interface Defaults verwenden, dann liest APEX beim Erstellen einer neuen Seite oder eines Reports den TABLE COMMENT und die COLUMN COMMENTS aus dem Data Dictionary und verwendet die dort gespeicherten Texte als Hilfetext für die Objekte in der Anwendungsoberfläche. Das ist sehr praktisch, vorausgesetzt, dass Sie Ihre Comments auch pflegen und dass sich diese Comments als Hilfetexte eignen. Es gibt zudem zwei Dinge zu beachten:

- Sie sollten gleich von Beginn an mit User Interface Defaults arbeiten, denn nach dem Erstellen von Seiten oder Items können Sie die „nachgepflegte“ Hilfetexte nicht mehr automatisch nachladen
- Änderungen, die Sie später (nach Erstellung der APEX-Seite) an den Table und Column Comments vornehmen, werden in der Anwendung nicht mehr synchronisiert

Ein Anwendungsbeispiel wird im Vortrag demonstriert.

Eigenleistung: Ein Hilfesystem von Usern für User

Den Hauptteil des Vortrags nimmt die Vorstellung eines Hilfesystems ein, das von den Benutzern selbst gestaltet werden kann. Dabei können Anwender die Inhalte der Anwendungshilfe selbständig erstellen beziehungsweise die vorhandene Hilfe ergänzen. Das Prinzip besteht darin, die Web-Anwendungen teilautomatisiert mit Zugriffspunkten für Hilfetexte zu bestücken, an denen die Benutzer ihre hilfreichen Informationen quasi „anheften“ können, auf die alle anderen Benutzer wiederum Zugriff haben. Durch ein (optionales) Berechtigungssystem kann erreicht werden, dass nicht nur allgemein lesbare Einträge erstellt werden können, sondern auch spezialisierte Hinweise von Benutzer A an Benutzer B,C und D oder von Gruppe X an Gruppe Y möglich sind. Dreh- und Angelpunkt des Systems sind APEX-Anwendungen und die Oracle-Datenbank, obwohl diese Idee vom Grundsatz her auch auf andere Webanwendungen übertragbar ist.

Ein zweiter Zweck der Übung besteht darin zu zeigen, welche Vorteile, selbst in Zeiten von „Rapid Application Development“ und Plugins, die individuelle Eigenentwicklung mit jQuery und CSS in APEX-Anwendungen bieten kann. Dabei ist es immer wieder erstaunlich, mit wie wenig Code die grundlegenden Ideen verwirklicht werden können, wenn man weiß, wo und wie man ansetzen muss. In diesem Rahmen wird auch eine kurze Einführung (oder Auffrischung) in jQuery-Selektoren gegeben.

Wo werden Hilfetexte hinterlegt?

Um ein wartbares und ausbaufähiges System zu bauen, kommt nur ein dediziertes Datenmodell in Frage. Column Comments als Grundlage für Anwendungs-Hilfesysteme zu verwenden mag für einfache Data-Entry-Formulare eine große Hilfe sein; in einer wachsenden Anwendungslandschaft skaliert diese Methode jedoch nicht gut (man denke allein an Texte, die länger als 4kB sind, oder Anforderungen wie etwa die Mehrsprachigkeit). Als Primary Key dienen technische IDs.

Worauf beziehen sich die Hilfetexte?

Zunächst, um das Beispiel überschaubar zu halten, auf die APEX-Seite selbst und auf ausgewählte, immer wiederkehrende Elemente in APEX-Komponenten. Da APEX in Formularen und Reports immer wieder die selben Entwurfsmuster verwendet, können CSS-Selektoren genutzt werden, um solche Elemente allgemeingültig zu identifizieren. Was etwas abstrakt klingt, heißt konkret: Wenn es auf einer Seite beispielsweise einen Report mit statischen Spaltennamen gibt (was in 99% der Fälle so ist), dann kann der Anwender einen Hilfetext in Bezug auf die Kombination „Seite + Spaltenname“ eingeben. Gibt es einen Button mit einer eindeutigen (HTML-)ID, dann kann auch dieser mit Erläuterungen durch die Anwender versehen werden. Letztendlich alles, was sich in Form eines CSS-Selektors beschreiben lässt, eignet sich auch als „Klebeplatz“ für Hilfetexte.

Wie werden die Hilfetexte visuell dargestellt?

Um die Anwendung nicht zu überfrachten, werden die Hilfetexte auf einer Seite nur dann eingeblendet, wenn der Benutzer eine (bestimmbare) Interaktion durchführt. Dies kann zum Beispiel das Drücken einer bestimmten Taste oder einer Tastenkombination sein, oder – was natürlich intuitiver ist – das Anklicken einer Hilfe-Schaltfläche an immer derselben Position auf dem Bildschirm. Tritt dieses Ereignis ein, so sorgt ein jQuery-Skript dafür, dass alle Elemente, denen eine explizite Hilfe zugeordnet wurde, dies in Form eines kleinen Badge zu erkennen geben – so ähnlich wie bei den auffälligen kleinen Zahlen-Stickern, die an den Icons unserer Handy-Apps kleben, wenn diese etwas zu melden haben. Bei einem Klick auf ein solches Badge wird der entsprechende Hilfetext per AJAX nachgeladen und in einem eigenen Layer über der Seite angezeigt. Als Transportmittel dafür kommt JSON zum Einsatz. Das Abholen der Texte aus der Datenbank erfordert lediglich einen Application Process.

Wie können die Hilfetexte eingegeben werden?

Selbstverständlich ebenfalls mit Hilfe einer APEX-App. Der Anwender, der einen Hilfetext schreiben möchte, klickt die Hilfe-Schaltfläche an und gelangt damit wahlweise in den Eingabemodus. Durch Anklicken eines „Klebepunktes“ öffnet sich eine Eingabemaske.

Hilfetexte können ebenso editiert oder wieder gelöscht werden. Das optionale Rechtesystem ermöglicht es, dass solch ein Text nach Art einer persönlichen Nachricht nur einem bestimmten anderen Anwender angezeigt wird, sobald dieser die Seite öffnet – beispielsweise wie „bitte schau nochmal über die Zahlen und schicke mir die korrigierte Fassung bis heute Mittag“ – und das im Rahmen einer „gewöhnlichen“ APEX-Anwendung.

Zusätzliche Vorteile

Dadurch, dass die Benutzer interaktiv an der Gestaltung der Anwendung teilhaben, steigt die Identifikation mit dem Produkt. Insofern besitzt ein solches Feature neben seiner bloßen Nützlichkeit auch einen Prestige-Aspekt. Welche andere Anwendung im Unternehmen kann das sonst noch bieten? Wenn man es zudem geschickt (sprich „chic“) programmiert, wird der Spieltrieb des Anwenders genutzt, um produktive Beiträge zu generieren. Dadurch, dass die Implementierung auf Web-Standards aufsetzt, lassen sich Drittanbieter-Plugins leicht integrieren.

Ausblick auf den Vortrag

Die Präsentation zeigt, wie man ohne spürbaren Entwicklungs-Mehraufwand in APEX konsistente Bedienkonzepte und ein wirklich nützliches Hilfesystem integrieren kann, das von den Anwendern nicht als bloßes Beiwerk wahrgenommen wird, sondern wie ein zunehmend unverzichtbarer Bestandteil und als echte Bereicherung für Ihre Anwendungen. Hierzu sind lediglich ein eigenes kleines Datenmodell sowie überschaubare jQuery- und PL/SQL-Skripte erforderlich. Einer der Schlüssel zum Erfolg liegt in der fortlaufenden aktiven Beteiligung der Nutzer, ähnlich einem Social Media Framework, in dem sich die User gegenseitig Tipps und Hilfestellungen geben können. Hierfür liefert APEX alles, was das (Entwickler-)Herz begehrt.

Kontaktadresse:

Andreas Wismann
WHEN OTHERS Inh. Andreas Wismann
Hirschstr. 10
D-41564 Kaarst

Telefon: +49 (0) 2131 - 314 9966
mobil: +49 (0) 176 - 7800 3109
E-Mail: wismann@when-others.com
Internet: when-others.com