

Web-Applikationsentwicklung mit OraclePortal

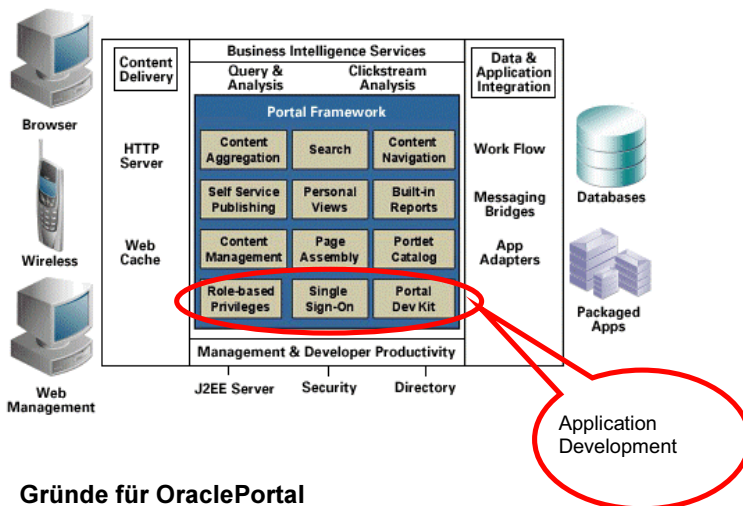
(publiziert in Fachzeitschrift Swiss Oracle User Group : SOUG-newsletter 1/2003)

OraclePortal bietet neben Content-Funktionen auch ein Modul zur *wizard-driven* Anwendungsentwicklung. **wega Informatik AG** hat mit OraclePortal eine multi-user Web-Applikation entwickelt. Unsere Erfahrungen und Beispiele von realisierten Funktionalitäten möchten wir hier gerne weitergeben.

Anforderungen konnte OraclePortal dann nur noch teilweise gerecht werden. Als Alternative kam daher die PL/SQL-Cartridge zum Einsatz, wodurch der Aufwand zwar erhöht wurde, die Benutzerwünsche aber vollumfänglich erfüllt werden konnten.

Beispiele der im Projekt verwendeten Feature und deren Implementierung:

Damit bei einem URL-Bookmarking ein unautorisierter Zugriff verhindert wird, bestand die technische Anforderung, zu jedem Zeitpunkt die genauen Privilegien des Benutzers zu kennen und diese bei jedem Seitenaufruf auch zu prüfen (*before displaying the page*-Trigger). Dies konnte mittels Sessionvariablen und Datenbankviews gelöst werden.



Gründe für OraclePortal

Die Vorteile zur Anwendung von OraclePortal sind nahe liegend: Einfache Verwendung, einfache Web-/Datenbankanbindung, und Unterstützung zum Rapid-Prototyping. Im konkreten Projekt liessen sich die Benutzeranforderungen durch ein einfaches Web-Interface umsetzen. Zusätzlich war bereits eine entsprechende Oracle-Umgebung zur Entwicklung und zum Deployment einer OraclePortal-Applikation beim Kunden vorhanden.

Applikationsfeatures

Es wurde daher ein Prototyping-Ansatz gewählt. Im weiteren Projektverlauf wurden die Anforderungen insbesondere an das Benutzer-Interface jedoch erweitert. Diesen neuen

- *Authentication via single-sign on*
Die Authentication konnte allein durch die ‚single-sign-on‘ Funktionalität von OraclePortal implementiert werden und auch in den PL/SQL-Cartridge basierenden Modulen integriert werden.

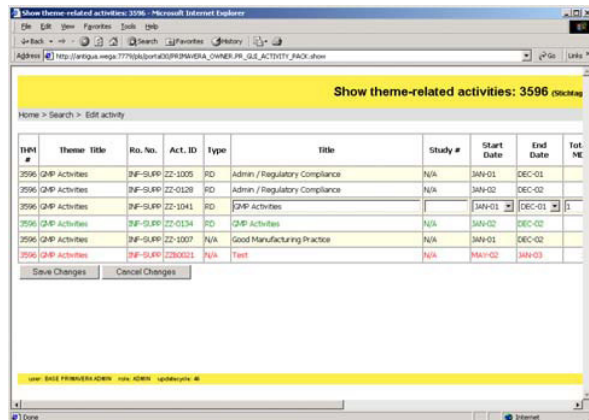
- *Autorisierung auf Funktions- und Daten-Ebene*
Die Benutzer können unterschiedliche Rollen mit unterschiedlichen Privilegien annehmen. Rollen bestimmen die Berechtigung auf der Funktionsebene, legen die Lese/Schreib-Rechte der Anwendungsdaten fest, und können die Sicht auf eine Untermenge der Daten beschränken.

- *Dateneingabe mittels row-gesteuerten Defaultwerten*
Bei der Dateneingabe war es notwendig, zahlreiche Felder mit Defaultwerten vorzubelegen. Jedoch waren diese Werte vom Benutzerkontext abhängig. Ein Implementierung via Datenbank-Trigger war nicht möglich, weil die Defaultwerte bereits vor dem Speichern dargestellt werden müssen. Die Verwendung eines PL/SQL-Event einer *Portal-Form* Komponente ermöglichte dann jedoch die Umsetzung dieser Anforderung.

- *kontextabhängige LOV's*
Es wurden 3 Typen von LOV's implementiert:
1. *einfache LOV:* Liste aller Werte einer Spalte, unabhängig vom Kontext
2. *benutzerabhängige LOV:* Liste wird bestimmt durch den Kontext des Benutzers, der sich während der gesamten Session nicht ändert.
3. *eingabeabhängige LOV:* Liste wird durch den Wert eines anderen Feldes derselben HTML-Eingabe-Form bestimmt.

Alle 3 LOV-Typen konnten mit OraclePortal implementiert werden, wobei

beim dritten Typ auf Bind-Variablen zurückgegriffen wurde.



ID#	Theme Title	Rto. No.	Act. ID	Type	Title	Study #	Start Date	End Date	Tot. NE
2596	QAP Activities	INF-SLRP 22-1000	RD	Admin / Regulatory Compliance		N/A	34N-01	DEC-01	
2596	QAP Activities	INF-SLRP 22-0128	RD	Admin / Regulatory Compliance		N/A	34N-02	DEC-02	
2596	QAP Activities	INF-SLRP 22-1041	RD	QAP Activities			34N-01	DEC-01	1
2596	QAP Activities	INF-SLRP 22-0134	RD	QAP Activities		N/A	34N-02	DEC-02	
2596	QAP Activities	INF-SLRP 22-1007	N/A	Good Manufacturing Practice		N/A	34N-01	DEC-02	
2596	QAP Activities	INF-SLRP 220002	N/A	Test		N/A	34N-02	34N-03	

- **Editiermodus ‚Einzelzeile‘**
Die Darstellung von Daten im Editiermodus ‚Einzelzeile‘ (siehe Abbildung) erlaubt dem Benutzer jeweils nur eine Zeile der Auswahlliste gleichzeitig zu ändern. Auf diese Weise kann eine sehr kontrollierte Save/Cancel-Navigation umgesetzt werden, war aber leider mit OraclePortal Mitteln allein nicht realisierbar. Hier kam die PL/SQL-Cartridge zur Anwendung.

- **Auto-Refresh nach Speichern**
Datenänderungen werden von OraclePortal standardmässig mit der Meldung ‚1 row successfully updated/inserted‘ quittiert. Dies wurde dahingehend angepasst, dass der Benutzer nach dem Speichern auf die Ursprungsseite zurückgeleitet wird und dort die neuen, erfolgreich gespeicherten Daten dargestellt bekommt. Dieses Verhalten konnte mit der in OraclePortal verfügbaren `go(<url>)`-Funktion im `redirect on successfull submission of-Trigger` realisiert werden.

Deployment und Portabilität

Das Deployment kann durch einfachen Export/Import erfolgen. Dies hat jedoch den Nachteil, dass damit auch alle Programmsourcen in die produktive Umgebung übernommen werden. Daneben weist OraclePortal jedem Objekt eine eindeutige ID zu, welche beim Export/Import ebenfalls ‚mitgenommen‘ wird. Dies kann zu potentiellen Konflikten in der Ziel-Umgebung führen.

Eingesetzt wird die Version v3.0.9.1 von OraclePortal. Bei den bereits mehrfach durchgeführten Versionsmigrationen, mussten wir leider immer wieder Portabilitätsprobleme feststellen, was einen erhöhten Aufwand zur Folge hatte. Demgegenüber läuft die installierte Umgebung sehr stabil. Dies hält wiederum den Supportaufwand gering.

Fazit

Ein Applikationsentwickler mit guten Oracle- und SQL-Kenntnissen kann mittels OraclePortal sehr schnell und einfach eine Web-Datenbank-Applikation erstellen. Steigen jedoch die Applikationsanforderungen, insbesondere die des Web-Frontends, dann sind detaillierte Kenntnisse von HTML und JavaScript unabdingbar, da vieles möglich ist, aber zur Umsetzung immer wieder (undokumentierte) Tricks und Workarounds notwendig werden.

Ist die Portal-Applikation dann erstellt, lässt sich diese relativ problemlos installieren und läuft sehr stabil. Allerdings sollte bei Versions-Migrationen unbedingt zusätzlicher Aufwand zum Testen und zur Programmanpassungen einkalkuliert werden.

Kontakt

Für weitere Fragen nehmen Sie unverbindlich Kontakt mit uns auf:

wega Informatik AG, Basel
Andre Matthijssen
www.wega-informatik.ch

E-Mail:
Andre.Matthijssen@wega-informatik.ch