

Erfahrungen mit UIX und JDeveloper im Projekteinsatz

Sven Tissot
pdv Technische Automation + Systeme GmbH
Hamburg

Schlüsselworte

UIX, J2EE, XML, BC4J, OC4J

Zusammenfassung

Was ist UIX? Wie kann mir ein Modell-View-Controller Framework bei der Entwicklung von Browser-basierten Anwendungen helfen? Wie einfach ist die Entwicklung von UIX-Anwendungen mit Oracle JDeveloper?

Diese Fragen soll der folgende Artikel beantworten. Anhand eines konkreten Projektes werden die Erfahrungen und Probleme – sowie deren Lösungsmöglichkeiten – dargestellt.

Was ist UIX?

UIX ist ein Framework zur Erstellung J2EE-konformer Web Anwendungen. Mit UIX stellt Oracle einen Baukasten aus Klassen-Bibliotheken, einer API, XML-Parsern, Wizards und anderen Software-Bausteinen zur Verfügung, mit dem **deklarativ** Anwendungen erstellt werden können, welche dem *Model-View-Controller Model 2* Design Pattern entsprechen:

- Die Präsentation auf dem Client (Browser, Pocket PC, ...) wird über *User Interface Komponenten* für unterschiedliche Ziel-Plattformen erzeugt.
- Der Kontrollfluss wird im *UIX-Controller* beschrieben und zur Laufzeit interpretiert.

Das Framework unterstützt verschiedene Strategien bei der Anwendungsentwicklung:

- Einbettung der Interface-Komponenten in z.B. Java Server Pages
- Verwendung der API in Java
- Deklarative Entwicklung über XML-Dateien
- Optionale Verwendung des UIX-Controllers
- Optionale Einbindung von Struts

Obwohl UIX in Java implementiert ist, können große Teile der Anwendung ohne Java realisiert werden.

Entwicklungsumgebung und Randbedingungen

Zum Projektstart wurden folgende Vorgaben gemacht:

- Entwicklung mit JDeveloper und UFX
- Möglichst wenig manuelle Java-Programmierung
- Datenhaltung in Oracle oder PostgreSQL-Datenbank
- Applikationsserver OC4J oder Tomcat
- Fortlaufende Projekt-Dokumentation in einem WikiWiki-Framework
- Einsatz von BC4J für die Generierung der Datenobjekte

Die Datenmodellierung erfolgte mit dem Tool DeZigner, welches es komfortabel ermöglicht, das Datenmodell für mehrere Ziel-Datenbanken zu erzeugen. Da sich jedoch bereits frühzeitig Probleme beim Zusammenspiel zwischen JDeveloper und PostgreSQL – insbesondere im Zusammenspiel mit BC4J – zeigten, wurde die PostgreSQL Variante gestrichen. Folgende Versionen der Entwicklungswerkzeuge wurden eingesetzt:

- Oracle Datenbank 9i Version 9.2.0.1 unter Suse Linux 7.2
- JDeveloper 9.0.3.2
- OC4J standalone 9.0.3
- CVS 1.11
- TWiki Version Dezember 2002

Bisherige Schwerpunkte und Erfahrungen in der Entwicklung

Einstieg und Lernkurve

Der Einstieg in die Entwicklung mit JDeveloper und UFX ist durch eine extrem hohe Lernkurve gekennzeichnet. Insbesondere die spärliche und teilweise inkorrekte Dokumentation in Form der JDeveloper-Hilfe (Kapitel UFX) und einer Basis-Dokumentation der API macht den Einstieg schwer:

- Erfahrener UFX Support durch Metalink ist nicht garantiert.
- Die Problemlösung erfolgt im Wesentlichen über die OTN-Foren.
- „Manisch Depressives Programmieren“: nach Phasen hoher Produktivität kämpft man tagelang mit Bugs in UFX oder im eigenen Code – wobei die Zuordnung sehr aufwendig ist.

Das macht das Entwickeln schwer

In der bisherigen Projektlaufzeit haben uns folgende Fehler teilweise erhebliche Probleme bereitet. Lösungsmöglichkeiten und Workarounds stellen wir bei der Live-Präsentation vor.

- Die Integration von JDeveloper und cvs ist nicht ideal gelöst. Insbesondere die Business Components und deren Einbindung in Applikations-Module führen zu Konflikten.
- Das Deployment in die OC4J Standalone-Version ist nicht unmittelbar möglich. Die Installation muss angepasst werden.

- Das Databinding ist nicht konsequent implementiert. Nicht alle UI-Komponenten können die Daten übernehmen.
- Tägliche Probleme und Bugs
 - Das deklaratives Datumsformat wird derzeit nicht ausgewertet.
 - Template Include files sind im handler nicht möglich.
 - Suchfilter sind nicht immer ersichtlich für den Anwender.
 - Die Range-Size bei Auswahllisten ist nicht korrekt implementiert.
 - Ein Debugger oder aussagekräftige Fehlermeldungen für UIX sind nicht implementiert.
 - Die Trennung von UIX und BC4J ist unscharf – nicht immer ist klar, was Probleme verursacht (und in welchem Forum man suchen soll).
 - Der integrierten OC4J hat Probleme beim Neustart („JVM-Bind“).

Das macht das Entwickeln leicht

Die Einbettung in ein Standard-Framework und die reichhaltigen Bausteine aus UIX und BC4J bieten erhebliche Vorteile:

- Viele Standard-Aufgaben sind implizit gelöst.
- Das Globales Layout ist durch den Einsatz von Templates sehr änderungsfreundlich.
- Eine einfache Implementierung des Kontrollflusses ist mit dem UIX-Controller möglich.
- Die Datenpräsentation, Navigation und Pflege in Tabellen wird durch Standard-Komponenten implementiert.
- Master-Detail Masken werden über Wizards generiert und dann angepasst.
- Die BC4J Komponenten bieten mächtige Funktionalität.
- Der Schwerpunkt der Entwicklung liegt auf der Interaktion und nicht in der Erstellung des Layouts. (BLAF - Browser Look and Feel, weitere Look and Feel Standards werden zukünftig bereitgestellt).

Das ist unverzichtbar

- Die fortlaufende Dokumentation aller Fehler und Workarounds in der lokalen Entwicklungsumgebung – hierfür hat sich TWiki (<http://twiki.org>) hervorragend bewährt.
- Die Schaffung einer eigenen UIX-Codebase mit Standard-Bausteinen (Auswahllisten, LOVs, Master-Detail, Pop-up Windows ...)
- Die Definition von Standards in der Namensgebung für Business Komponenten
- Der (teilweise) Einsatz von Techniken des *extreme programming*
 - *Release early, release often, release small*
 - Frühe Akzeptanztests
 - Arbeiten im Team (2 Entwickler – 1 Bildschirm)
- **Profundes** J2EE, BC4J und XML-Know-How

Fazit

UIX stellt einen viel versprechenden Ansatz dar, um im komplexen Umfeld von J2EE und MVC Modell getriebenen Entwicklungen produktiv und sicher Standard-Anwendungen zu realisieren.

Die Komplexität der Programmierung im Umfeld von J2EE Architekturen kann jedoch auch durch UIX nicht entscheidend vereinfacht werden. Neben tief greifender Kenntnis der J2EE-Architektur sind zusätzliche Beherrschung von XML und XML-Namespaces, BC4J und MVC notwendig, um mit der vom Oracle Developer gewohnten hohen Produktivität Anwendungen zu realisieren.

Die derzeitige Version (Stand JDeveloper 9.0.3.2) zeigt noch zu viele Lücken und Fehler auf, um für große Projekte empfohlen zu werden. Standard-Anwendungen im Layout des Oracle Browser Look and Feel lassen sich jedoch schnell und einfach generieren.

Kontaktadresse:

pdv Technische Automation + Systeme GmbH
Sven Tissot

Dorotheenstraße 64
D-22301 Hamburg

Telefon: +49(0)40-69213-306
Fax: +49(0) 40-69213-278
E-Mail: tissot@pdv-tas.de
Internet: <http://www.pdv-tas.de>