

Trotz immer moderneren Web-Frameworks und aufwändigen AJAX-Features müssen Endanwender beim Umstieg von Forms-Anwendungen in Richtung Web Abstriche beim Bedienkomfort und der Usability machen. Rich Internet Applications schließen diese Lücke und ermöglichen es, zeitgleich die gesamte Software-Architektur der Forms-Lösung schrittweise für zukünftige Anforderungen fit zu machen. Der Artikel zeigt ein exemplarisches Vorgehen mit Adobe Flex und geht auf die Herausforderungen bei der Forms-Modernisierung ein.

Desktop-Feeling auch im Web – Forms-Modernisierung mit Rich Internet Applications

Robert Szilinski, esentri consulting GmbH

Viele Unternehmen stehen vor der Frage, ob und wie sie ihre bewährten Forms-Anwendungen durch moderne Technologien ablösen können. Nicht selten wurden viele Hundert Personennjahre in die Entwicklung der Geschäftslogik investiert und es ist kaum vorstellbar, von heute auf morgen mit einem „Big Bang“ auf eine neue Lösung umzustellen. Zu groß sind die Herausforderungen, die sowohl auf die IT-Verantwortlichen als auch auf die Endanwender warten.

Web-Anwendungen oder Fat-Client?

State of the art ist, dass man heute „im Web“ arbeitet und reine Desktop-Lösungen als nicht mehr zeitgemäß gelten. Zwar bieten diese nach wie vor den größten Bedienkomfort und haben durchaus ihre Vorteile; sollte jedoch zu einem späteren Zeitpunkt auch eine Web-Lösung gefordert werden, heißt es in der Praxis häufig, die Anwendung nochmals komplett für das Web zu entwickeln – Code muss also doppelt gepflegt werden und das produziert unnötig hohe Kosten.

Auf der anderen Seite haben klassische Web-Lösungen ebenfalls ihre Schwächen. Zunächst einmal müssen sich die Anwender auf das neue Arbeiten im Web einstellen. Web-Anwendungen werden im Wesentlichen durch das Request-Response-Prinzip geprägt und das Springen zwischen den Seiten sowie die vorwiegende Steuerung mit der Maus anstatt der Tastatur ist für vie-

le eine große Umstellung. Gleichzeitig liegt die Transaktionskontrolle bei modernen Web-Frameworks typischerweise nicht mehr beim Endanwender sondern im Application Server – nicht für alle Anwendungsfälle ist dies ein gewünschtes Verhalten. Das Problem verschärft sich noch, wenn beispielsweise die alte Forms-Anwendung (stateful)

PL/SQL-Geschäftslogik aufruft und die Logik auch bei der Umstellung auf eine neue Technologie weiterverwendet werden soll. Insbesondere der Einsatz globaler PL/SQL-Variablen, die in der Datenbank-Session gehalten werden, erschwert die Weiterverwendung der Logik, denn die meisten Web-Frameworks kommen durch den Einsatz des

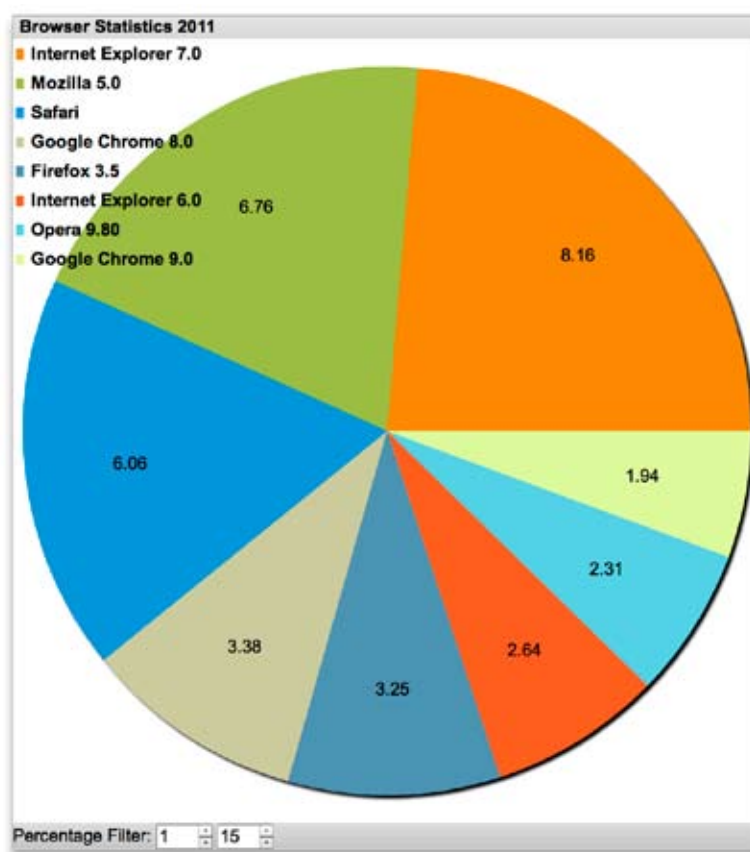


Abbildung 1: Beispiel eines einfachen Flex-Dialogs mit Pie-Chart

```

<?xml version="1.0"?>
<s:Application
  xmlns:fx="http://ns.adobe.com/mxml/2009"
  xmlns:mx="library://ns.adobe.com/flex/mx"
  xmlns:s="library://ns.adobe.com/flex/spark"
  keyDown="application1_keyDownHandler(event)"
  creationComplete="onCreationComplete(event)">

  <fx:Script>
    <![CDATA[
      import mx.collections.ArrayCollection;
      import mx.controls.Alert;
      import mx.events.FlexEvent;

      [Bindable] public var chartData:ArrayCollection = new ArrayCollection([
        {browser:"Firefox 3.6", perc:36.82},
        {browser:"Internet Explorer 8.0 ..", perc:24.58},
        {browser:"Internet Explorer 7.0 ..", perc:8.16},
        {browser:"Mozilla 5.0 ..", perc:6.76},
        {browser:"Safari", perc:6.06},
        {browser:"Google Chrome 8.0", perc:3.38},
        {browser:"Firefox 3.5 ..", perc:3.25},
        {browser:"Internet Explorer 6.0", perc:2.64},
        {browser:"Opera 9.80", perc:2.31},
        {browser:"Google Chrome 9.0", perc:1.94}
      ]);

      protected function onCreationComplete(event:FlexEvent):void {
        chartData.filterFunction = filterFunc;
        chartData.refresh();
      }

      private function filterFunc( item:Object ):Boolean {
        return ( item.perc >= minPerc.value && item.perc <= maxPerc.value );
      }

      protected function application1_keyDownHandler(event:KeyboardEvent):void
      {
        if( event.keyCode == Keyboard.F1 ) {
          Alert.show(„F1“);
        }
      }

    ]]>
  </fx:Script>

  <s:Panel verticalCenter="0" horizontalCenter="0" title="Browser Statistics 2011">
    <mx:PieChart id="pieChart" dataProvider="{ chartData }">
      <mx:series>
        <mx:PieSeries field="perc" nameField="browser" labelPosition="inside" />
      </mx:series>
    </mx:PieChart>
    <mx:Legend dataProvider="{ pieChart }"/>
    <s:controlBarLayout>
      <s:HorizontalLayout verticalAlign="middle"/>
    </s:controlBarLayout>
    <s:controlBarContent>
      <s:Label text="Percentage Filter: .."/>
      <s:NumericStepper id="minPerc" valueCommit="{ chartData.refresh() }" maximum="40" value="0" />
      <s:NumericStepper id="maxPerc" valueCommit="{ chartData.refresh() }" maximum="40" value="40" />
    </s:controlBarContent>
  </s:Panel>
</s:Application>

```

Listing 1: Der gesamte MXML-Code für Abbildung 1 mit Actionscript

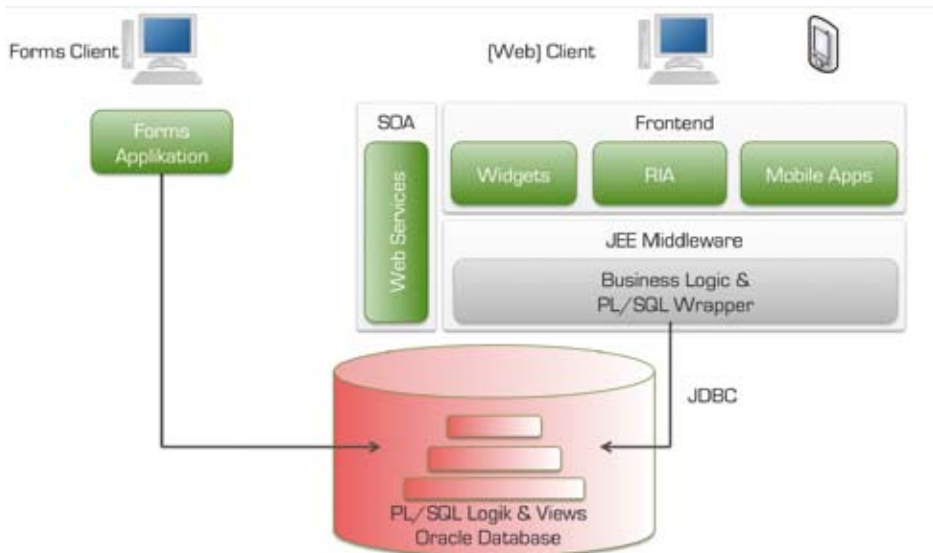


Abbildung 2: Forms-Modernisierung mit Java Middleware und Frontend-Alternativen

Connection-Pooling-Konzepts nicht ohne Weiteres damit zurecht. Ein Parallelbetrieb zwischen „alter“ und „neuer“ Welt wäre so nicht ohne Weiteres möglich.

Es stellt sich also die zentrale Frage, welche Technologie am besten für die Modernisierung der eigenen Forms-Anwendungen geeignet ist. Einen Königsweg kann es aufgrund der unterschiedlichen Anforderungen in dem jeweiligen Unternehmenskontext nicht geben. Im Folgenden wird aber ein vielversprechender Ansatz vorgestellt, bei dem kein Endanwender im Frontend auf liebgewonnene Funktionen verzichten muss und die IT-Architektur im Backend dennoch enormes Potential für zukünftige Entwicklung bietet. Dabei wird der Oracle-Technologie-Stack nur im Frontend zugunsten des Open-Source-Frameworks Adobe Flex verlassen.

Rich Internet Applications – das Beste aus beiden Welten

Gemessen an heutigen Maßstäben werden die Benutzeroberflächen von Forms-Anwendungen heute eher als alt und „unsexy“ wahrgenommen. Große „Wow-Effekte“ erzeugen dagegen vor allem die Rich Internet Applications, allen voran die Adobe-Flash-Technologie, die häufig zum Visualisieren von Charts oder aufwändigen Komponenten

in Web-Anwendungen verwendet wird (siehe Abbildung 1).

Dabei bietet Adobe mit dem Flex-Framework noch weitaus mehr Möglichkeiten, auch eigene Benutzeroberflächen mittels Action Script und MXML zu programmieren (siehe Listing 1), die dann ebenso im Flash Player des Browsers ausgeführt werden können, jedoch keine HTML-Grundlage mehr haben. Aktuelle Studien zufolge ist trotz der benötigten Laufzeitumgebung keine zusätzliche Installation auf dem Client notwendig, da über 99 Prozent der Browser das Flash-Plug-in bereits mitbringen.

Das Erstellen einfacher Pflegemasken und Dialoge geht dabei recht schnell von der Hand – schließlich stehen sehr viele ausgereifte GUI-Komponenten zur Auswahl – und orientiert sich vollständig am MVC-Paradigma. Die Entwicklung kann daher hervorragend auf verschiedene Personen verteilt werden und auch eine agile Vorgehensweise, bei der beispielsweise die neuen Benutzeroberflächen schnell und zeitnah mit den Fachverantwortlichen abgestimmt werden können, wird gefördert.

Neben der guten IDE-Unterstützung ist es zur weiteren Beschleunigung der Entwicklung natürlich auch möglich, die Forms-Anwendung nach XML zu exportieren (es gab zu diesem Thema bereits einige Beiträge in den DOAG News und Vorträge auf Veranstaltungen) und



Oracle Projekte agil durchführen!

Nutzen Sie die Vorteile agiler Methoden und führen Sie Ihre IT-Projekte zum Erfolg.

Wir zeigen Ihnen wie!

Mit der richtigen Strategie

Die Zukunft aktiv gestalten durch Geschäftsprozessoptimierung und konsequentes IT-Business-Alignment. Nutzen Sie das Potential Ihrer IT, um wettbewerbsfähig zu bleiben und Innovationen zu realisieren.

Mit der richtigen Technologie

Das Beste aus beiden Welten mit Oracle Technologien und Open Source. Richtungsweisende Web-Applications mit Java Frameworks, ADF, APEX, Adobe Flex und Oracle WebCenter. Modernisieren Sie bewährte Forms Anwendungen durch Rich Internet Applications und die SOA Suite.

Mit der richtigen Methode

esentri verbindet die Konzepte sozialer Netzwerke mit bewährten Projektmanagementmethoden und agilen Vorgehensweisen. Mit unserer innovativen Social PM Methode wird jedes Projekt als soziales Netzwerk verstanden und die Kommunikation optimiert. Mehr Infos unter: <http://www.social-pm.com>

Mit dem richtigen Partner

esentri
consulting GmbH

Pforzheimer Straße 132
76275 Ettlingen
07243 / 354 90 0
www.esentri.com
info@esentri.com
twitter.com/esentri

anschließend in MXML (der Auszeichnungssprache von Flex) zu transformieren, um eine gute Ausgangslage für eine schnelle Flex-Entwicklung zu schaffen. Je nach Maske empfiehlt es sich aber, nicht einfach exakt die alte Funktionalität in eine neue Flex-Maske zu übernehmen, sondern in diesem Zug auch die Gestaltung der Benutzeroberflächen zu überdenken und ggf. die GUI-Prozesse zu optimieren.

RIA ist mehr als nur schöne Masken

Neben sehr ansprechenden Masken bieten Rich Internet Applications aber noch weitaus mehr interessante Features, die im Rahmen der Technologie-Auswahl berücksichtigt werden sollten. Gerade für Power-User sind beispielsweise die vielfältigen Möglichkeiten der Tastaturbenutzung sehr wichtig. So kann Flex sämtliche Tastatureingaben verarbeiten und auch die beliebten Funktionstasten können weiter verwendet werden.

Für das Arbeiten mit der Maus bietet das Flex-Komponentenmodell die Möglichkeit, echtes „Drag & Drop“ zu implementieren, und somit auch die Chance, die in die Jahre gekommenen Forms-Oberflächen auch bezüglich Usability zu modernisieren. Generell findet bei RIA übrigens zu keinem Zeitpunkt ein spürbares Nachladen der Anwendung statt.

Besonders spannend ist auch die Möglichkeit, Echtzeit-Features in der Anwendung zu realisieren (Server Push sowie Publish & Subscribe). So können Daten über mehrere Clients hinweg zeitgleich synchronisiert und Konflikte direkt aufgelöst werden. Interessant ist diese Möglichkeit vor allem beim zeitgleichen Bearbeiten von Kundendaten, da Datensätze nicht mehr exklusiv gesperrt werden müssen, sondern Änderungen direkt für alle Endanwender zeitgleich sichtbar sind. Die Transaktionskontrolle liegt dabei im Gegensatz zu klassischen Web-Anwendungen voll beim Benutzer.

Ein weiteres Highlight ist die Möglichkeit, eine Flex-Anwendung mit einer Offline-Fähigkeit auszustatten. Ein typisches Beispiel für diesen Bedarf ist ein Vertriebsmitarbeiter, der zeitwei-

se offline beim Kunden ist oder unterwegs noch seine Notizen zum letzten Kundengespräch machen möchte. Diese kann er fortan ohne bestehende Internetverbindung machen, da der Flex-Client die Möglichkeit besitzt, Daten in einer lokalen Zwischendatenbank (wie SQLite) zu speichern. Diese werden dann analog den Echtzeitdaten synchronisiert, sobald der Client wieder online ist.

Technisch findet die Kommunikation zwischen dem Browser und dem Backend übrigens über ein Binärprotokoll (AMF) statt, das im Gegensatz zu HTML über HTTP sehr schnell und komprimiert auch große Datenmengen an den Client übertragen kann.

Java Middleware als Grundlage für RIA und SOA

Damit Flex-Applikationen mit der Datenbank kommunizieren können, wird der Aufbau einer Zwischenschicht (Middleware) empfohlen. Diese kann vollständig mit dem Oracle-JEE-Stack implementiert werden und ist damit 100-prozentig kompatibel zu den gängigen Standards. Die Brücke zwischen Java und Flex ist übrigens nicht proprietär, sondern lässt sich mit den Produkten BlazeDS (Adobe) und GraniteDS komplett mit Open-Source-Software realisieren.

Je nach Bedarf ist es in der Middleware auch möglich, bestehende PL/SQL-Geschäftslogik durch Java-Wrapper-Klassen weiter zu nutzen und somit die wertvolle Investition zu schützen. Dabei ist auch der Parallelbetrieb zwischen alter und neuer Technologie möglich, indem die Middleware sich wie ein Forms-Client verhält (siehe Abbildung 2). Zusätzlich besteht die Möglichkeit, dass diese Wrapper-Klassen auch in Form von Web-Services veröffentlicht werden und ein strategischer Ausbau in Richtung Service-orientierter Architekturen forciert werden kann. Diese Überlegungen sollten elementar bei der Modernisierung von Forms-Anwendungen sein, denn nur der Austausch der Benutzeroberflächen bietet langfristig keinen wirklichen Mehrwert. Der Nutzen wird aber durch eine Öffnung gegenüber ande-

ren Applikationen und Geschäftsprozessen mit BPEL erreicht, insbesondere um Medienbrüche abzubauen und Mehrfacheingaben zu vermeiden. Das Vorgehen bietet außerdem die Flexibilität, weitere Frontend-Anwendungen wie Widgets oder mobile Applikationen auf Basis der gleichen Geschäftslogik zu entwickeln (siehe Abbildung 2).

Fazit

Rich Internet Applications vereinen das Beste aus Web-Anwendungen und Fat-Clients. Sie sind damit eine interessante Alternative für Firmen, die eine Web-Strategie verfolgen, ihren Anwendern aber den Komfort einer Desktop-Anwendung bieten möchten. Zusätzliche spezielle Features wie Echtzeit- oder Offline-Fähigkeit sind weitere Alleinstellungsmerkmale. Die Auswahl der Benutzeroberflächen ist dennoch nur ein Teil der Lösung und es gibt keinen Königsweg für die Forms-Modernisierung – anstatt Adobe Flex könnte man je nach konkreten Anforderungen alternativ auch klassische Web-Frameworks mit AJAX oder zukünftig JavaFX einsetzen.

Wichtig ist, dass die individuellen Anforderungen genau analysiert werden und die Forms-Modernisierung auch als Chance verstanden wird, Prozesse zu optimieren und die IT-Strategie zu überarbeiten. Ansonsten besteht die Gefahr, am Ende für viel Geld nur alten Wein in neue Schläuche gegossen zu haben.

Mit Flex wurde in diesem Artikel eine spannende Alternative vorgestellt, die sich sehr gut mit einer Java-Middleware- beziehungsweise SOA-Strategie kombinieren lässt und für bestimmte Anwendungsfälle einen enormen Mehrwert liefert. Diese könnten im Rahmen einer schrittweisen Vorgehensweise als Rich Internet Applications umgesetzt werden und auch bestehende Forms-Anwendungen jederzeit sinnvoll ergänzen.

Kontakt:

Robert Szilinski
robert.szilinski@esentri.com