

5 Frontend und GUIs

Während früher die Gestaltung der Benutzeroberflächen einer Software als nachrangig eingestuft wurde, spielt die Leistungsfähigkeit des Frontends heutzutage eine Schlüsselrolle bei der Akzeptanz einer Applikation.

Anwendungssoftware wird heute bei Tage entwickelt. Ausgehend von der Devise WYSIWYG (*What You See Is What You Get*), legt man zu Beginn einer Entwicklung zunächst einmal fest, wie man dem Anwender die gewünschten Informationen präsentieren möchte und auf welche Weise Eingaben erfolgen.

Anwendungssoftware wird vom Frontend aus entwickelt

Es ist nicht mehr zeitgemäß, raffinierte Routinen zu schreiben und dann eine lieblose Oberfläche hinzuzufügen. Vielmehr muss die Oberfläche mit einer angemessenen eigenen Intelligenz ausgestattet sein, um typische, an Oberflächen orientierte Handlungen auch direkt ohne Rücksprache mit dem Server durchführen zu können.

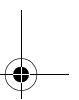
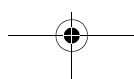
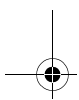
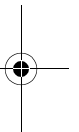
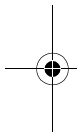
Oberfläche muss Teil der Netzwerkintelligenz übernehmen

Bisher haben die meisten Entwickler mit Vorliebe »im Dunkeln« programmiert. Lange Strecken eines Programms wurden auf gut Glück und Gottvertrauen hin entwickelt und am Ende einem mehr oder minder komplexen, aber immer unvollständigen Testszenario zugeführt.

Heute programmiert man ganzheitlich. Man beginnt bei der Oberfläche, die der Benutzer zu sehen bekommt. Am Anfang ist diese Oberfläche ganz einfach und einfach ist auch das Programm, das die ersten Daten liefert. Jedes Feature an der Oberfläche bekommt ein überschaubares Pendant in der Applikation und jede Änderung an der Logik der Applikation muss sich sichtbar an der Oberfläche widerspiegeln. Dieser neue Programmierstil hat auch einen Namen in der Entwicklergemeinde: *Extreme Programming* oder *Holistic Development*.

Holistic Development oder Extreme Programming

Leider ist die Entwicklung von Weboberflächen keineswegs so trivial, wie es zunächst den Anschein hat. Zwar kann jeder nach spätestens einer Stunde Crash-Kurs seine ersten HTML-Seiten gestalten und nach einer kurzen Spielphase mit Cascading Style Sheets gelingt es auch, die Seiten ansprechend zu gestalten und ein halbwegs einheitliches Layout über mehrere Seiten hinweg durchzuhalten. Das Problem bei der Entwicklung von Oberflächen für das Web liegt darin, dass es nur sehr wenige Tools auf dem Markt gibt, die es erlauben, gleichermaßen ansprechende wie



performante Oberflächen zu gestalten und dies noch dazu mit möglichst geringem Entwicklungsaufwand.

Klassische Webseiten sind langsam und flackern

Wenn Sie häufiger Ihre Einkäufe im Internet tätigen, ist Ihnen sicherlich aufgefallen, dass der Aufbau der Seiten bei sehr vielen Webshops lange dauert. Klickt man auf einen Link, tut sich zuerst gar nichts, dann verschwindet die Seite ganz und die neue Seite wird endlich aufgebaut. Dies geschieht auch dann, wenn sich nur ein kleines Detail in Ihrer Eingabe ändert, etwa Sie geben an, dass Ihre Adresse in den USA und nicht in Europa ist, welches ein etwas verändertes Eingabeschema verlangt. In der Praxis erweckt das den Eindruck, die Webseiten flackerten. Der Grund liegt darin, dass Webserver den Datenstrom zwischen Server und Browser nicht optimieren und auch keine Transaktionsunterstützung bieten.

Transaktionsmanagement findet auf Applikationsserver statt

Das Transaktionsmanagement wird heutzutage auf die Applikationsserver ausgelagert. Hierfür bietet Java jetzt die J2EE-Technologie, der Web AS von SAP greift auf die bewährte SAP R/3-Infrastruktur zurück und Microsoft.NET hat seinen Webserver um den Microsoft Transaction Server erweitert. Mit jeder dieser Technologien muss sich der Entwickler der Weboberflächen und auch der Weblogiken keine Gedanken mehr über die Probleme machen, die mit unvollendeten Transaktionen in Verbindung stehen.

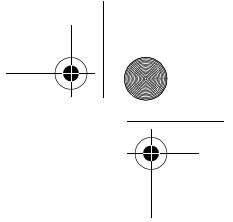
GUI-Server sind eine viel versprechende Lösung

Bei der Optimierung der Entwicklung und Ausführung von Weboberflächen sind die Werkzeuge noch sehr rar gesät. Für den Einsatz zusammen mit SAP R/3 bieten sich die folgenden drei Lösungen an:

- ▶ **Synactive Viewer**
von Synactive, der Firma des Schöpfers von ABAP, Dr. Gerhard Rodé
- ▶ **Web Dynpro**
die Webentwicklungsplattform von SAP
- ▶ **CASABAC GUI-Server**
von der Firma CASABAC

5.1 Casabac GUI-Server

Als Beispiel für eine bereits verfügbare Technologie (Web Dynpro wird erst mit dem Web AS-Release 6.30 verfügbar sein) stellen wir hier den Casabac GUI-Server vor. Der Casabac GUI-Server erweitert die Funktionalität und die Performance eines Browsers um Anwendungsgebiete, die bislang auf Grund ihrer Anforderungen an Interaktion nicht vernünftig realisiert werden konnten oder ein zusätzliches Plug-In wie Macromedia Flash auf der Browserseite erforderten.

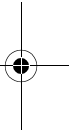
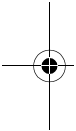


Casabac ist eine Entwicklung von Björn Müller, einem ehemaligen Software-Ingenieur von SAP, dessen Ideen zu großen Teilen die Entwicklung der Web-Dynpro-Technologie des SAP Web AS beeinflusst haben. Während Web Dynpro als Teil des Web AS (ab Release 6.30) positioniert ist, ist Casabac unabhängig von der SAP-Kerntechnologie.

5.1.1 Technologie

Genau wie Web Dynpro ist Casabac eine Entwicklungs- und Runtime-Umgebung zur Interaktion von Browseroberflächen mit Unternehmensapplikationen. Das besondere Kennzeichen von Casabac ist einerseits die flackerfreie Darstellung von Seiteninteraktionen auf Basis einer erweiterbaren reichhaltigen Control-Bibliothek. Andererseits ist die Programmierung von komplexen Oberflächen und deren serverseitige Anbindung an existierende Logik extrem einfach und offen.

■■■Bild

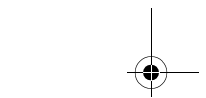


■■■Bild 04_01 fehlt!!!■■■

Abbildung 5.1 Casabac GUI-Server – Beispielseite

Technisch gesehen ist der Casabac GUI-Server eine auf dem Java-Servlet-Interface aufsetzende Infrastruktur: Seiten werden in einem WYSIWYG-Editor erstellt und als XML-Layout-Definition abgelegt. Aus dieser Layout-Definition wird Code für DHTML oder JavaScript generiert. Jede Seite hat einen Bezug auf eine serverseitige Java-Klasse, die so genannte *Adapterklasse*. Controls auf der Seite beziehen sich auf Properties und Methoden dieser Adapterklasse. Die Adapterklasse ist somit ein logisches Spiegelbild der Seite.

CASABAC setzt auf Servlets auf



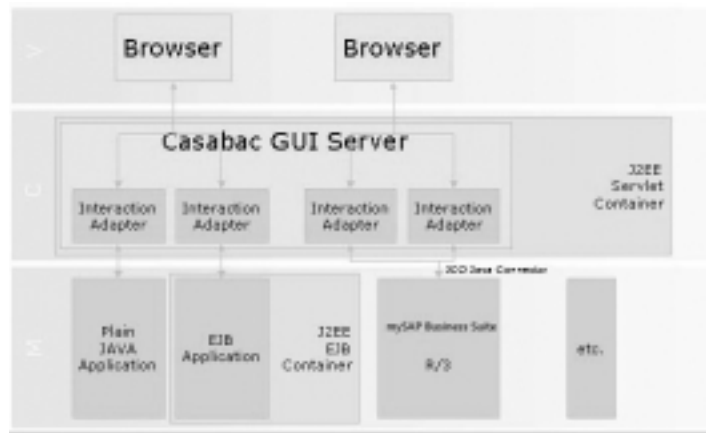


Abbildung 5.2 Casabac-Technologie

**Informationen-
saustausch über
Hidden Frame und
DHTML**

Das Besondere ist, dass die Seite nicht wie normale HTML-Seiten direkt mit Ihrer Serververarbeitung kommuniziert, sondern über einen »Hidden Frame« innerhalb der Seite. Hierdurch bleibt einerseits die Seite stabil im Browser und flackert nicht bei Serverinteraktionen. Andererseits wird zwischen Browser-Client und Server nur der logische Inhalt der Seite kommuniziert. Gegenüber normalen Webanwendungen, in denen der Server immer eine gesamte HTML-Seite zum Client schickt, bedeutet dies eine massive Reduzierung des ausgetauschten Datenvolumens.

Die Interaktivität der Browserseite entspricht somit der Interaktivität von normalen »Rich Clients«, die in C++, Java-Swing und so weiter codiert sind. Der Browser wird um Anwendungsgebiete erweitert, die bislang auf Grund ihrer Interaktionsanforderungen nur durch Erweiterung des Browsers mit Plug-Ins möglich waren. Plug-Ins bedeuten jedoch eine Manipulation des Standardsystems, was in einer sicherheitssensiblen Umgebung nicht ohne weiteres machbar ist.

**Casabac läuft
innerhalb von
J2EE**

CASABAC selbst läuft innerhalb jedes J2EE Servers – z. B. auch innerhalb des SAP Web Application Servers. Die Standardauslieferung erfolgt auf Basis von Tomcat. Features wie Mehrsprachenverwaltung, internationalisierte Controls und Adapter-Code-Generator gehören zur Standardausstattung.

Im Browser werden nur Standardtechnologien wie JavaScript und DHTML genutzt, es wird kein Plug-In oder Ähnliches verwendet.

5.1.2 Beispiel zur Kalkulation der Mehrwertsteuer

Das Beispiel aus Abbildung 5.3 ist eine Mini-Anwendung, innerhalb derer ein Mehrwertsteuerbetrag errechnet wird.



Abbildung 5.3 Beispiel einer Mini-Anwendung zur Berechnung der Mehrwertsteuer

Im Layout Painter wird zunächst die Seite designt. Der Inhalt der Seite besteht aus einzelnen Controls. Es gibt Container Controls, die wiederum selbst andere Controls enthalten können, und es gibt normale Controls, wie z. B. Eingabefelder und Buttons. Das XML-Layout der Seite sieht folgendermaßen aus:

Listing 5.1 XML-Layout einer Casabac-Seite

```
<page model="TaxCalcAdapter">
  <titlebar name="VAT Calculator"></titlebar>
  <header withdistance="false">
    <button name="Exit" method="endProcess"></button>
  </header>
  <pagebody>
    <rowarea name="VAT Input">
      <itr>
        <label name="Net Amount" width="120"/>
        <field valueprop="netAmount"
              width="200" datatype="float"/>
      </itr>
    </rowarea>
  </pagebody>
</page>
```

```

<vdist height="10"/>
<itr>
  <label name="VAT Category" width="120"/>
  <radiobutton valueprop="vatCategory"
    value="0"/>
  <label name="Without VAT" width="200"
    asplaintext="true"/>
</itr>
<itr width="120">
  <hdist width="120"/>
  <radiobutton valueprop="vatCategory"
    value="8"/>
  <label name="Reduced VAT" width="200"
    asplaintext="true"/>
</itr>
<itr width="120">
  <hdist width="120"/>
  <radiobutton valueprop="vatCategory"
    value="16"/>
  <label name="Full VAT" width="200"
    asplaintext="true"/>
</itr>
<vdist height="10"/>
<itr>
  <hdist width="120"/>
  <button name="Calculate"
    method="onCalculate"/>
</itr>
</rowarea>
<rowarea name="VAT Result">
  <itr>
    <label name="VAT Amount" width="120"/>
    <field valueprop="vatAmount" width="200"
      displayonly="true"
      datatype="float"/>
  </itr>
</rowarea>
</pagebody>
<statusbar withdistance="false"/>
</page>

```

Sie sehen, die XML-Layout-Definition entspricht genau der Darstellung der Seite. Die Seite selbst (page-Control) bezieht sich auf die Adapterklasse `TaxCalcAdapter`. Die Felder beziehen sich auf Properties der Klassen, z.B. bezieht sich ein Feld (`field-Control`) über sein `valueprop`-Attribut auf einen bestimmten Property-Namen. Der Button wiederum bezieht sich auf eine Methode `onCalculate`.

■■■Bild

■■■Bild 04_04 fehlt!!!■■■

Abbildung 5.4 Das Layout-Painter-Tool selbst läuft im Browser

Die in der Seitendefinition referenzierten Properties und Methoden werden über die Adapterklasse zur Verfügung gestellt:

Listing 5.2 Eine Minimal-Implementierung der Adapterklasse

```
import java.util.*;
import java.math.*;
import com.casabac.server.*;
import com.casabac.server.util.*;
import com.casabac.util.*;

public class TaxCalcAdapter
    extends Model
{
    // -----
    // property access
    // -----
}
```

```
BigDecimal m_netAmount = new BigDecimal(0);
public BigDecimal getNetAmount() { return
m_netAmount; }
public void setNetAmount(BigDecimal value) {
m_netAmount = value; }

BigDecimal m_vatAmount = new BigDecimal(0);
public BigDecimal getVatAmount() { return
m_vatAmount; }
public void setVatAmount(BigDecimal value) {
m_vatAmount = value; }

BigDecimal m_vatCategory = new BigDecimal(0);
public BigDecimal getVatCategory() { return
m_vatCategory; }
public void setVatCategory(BigDecimal value) {
m_vatCategory = value; }

// -----
// public adapter methods
// -----

public void onCalculate()
{
    m_vatAmount =
    m_netAmount.multiply(m_vatCategory);
    m_vatAmount = m_vatAmount.divide(new
    BigDecimal(100),2);
}
}
```

In diesem Beispiel wurde der für eine Geschäftslogik passende Datentyp `BigDecimal` verwendet. Natürlich können auch alle anderen Datentypen wie z.B. `float`, `int` etc. verwendet werden.

Mehr als die beiden gezeigten Bestandteile – Layout-Definition und Klasse – werden nicht benötigt. Der Anwendungsentwickler kommt nicht in Kontakt mit Stylesheets, JavaScript, Server-Session-Verwaltung, serverseitigem Scripting oder Ähnlichem.

In dem Beispiel wurde die Mehrwertsteuer direkt im Adapter berechnet. In einem strukturierteren Ansatz würde man alle Geschäftslogik natürlich nicht direkt in dem Adapter codieren, sondern über APIs kapseln. Vom Adapter aus können diese APIs dann angesprochen werden. Beispiele hierzu sind

- ▶ Aufruf von SAP-BAPIs über Java Connector, Web AS, Business Connector oder einen anderen HTTP-Message-Server
- ▶ Aufruf von auf Enterprise JavaBeans (EJB) basierenden Systemen
- ▶ Aufruf von beliebigen Webservices

5.2 Web Dynpro und Business Server Pages

Mit Web Dynpro versucht SAP, die klaffende Lücke zwischen Applikation und Browseroberfläche für Anwender und Entwickler gleichermaßen zu schließen. Aufbauend auf Business Server Pages, bietet Web Dynpro eine Entwicklungsumgebung und einen intelligenten GUI-Server, der das Entwickeln von Transaktionen für das Internet unterstützt.

Web Dynpro ist eine Entwicklungsumgebung zur Frontend-Entwicklung

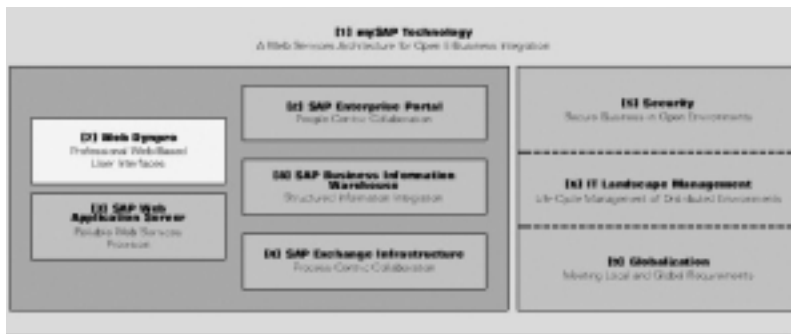


Abbildung 5.5 Web Dynpro innerhalb von mySAP Technology/SAP NetWeaver

Web Dynpro wird nach dem jetzigen Stand ab Web AS Release 6.30 zur Verfügung stehen. Es wird keine Lösungen für ein früheres Release geben. Frontends werden fortan auf einem dedizierten Web AS entwickelt, der dann über RFC auf andere SAP R/3-Instanzen zugreift.

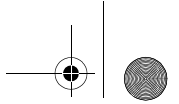
Web Dynpro ab Release 6.30

Die Grundlage von Web Dynpro ist der *Model View Controller (MVC)*. Dies ist ein Repository, das die Informationen über Aufbereitung und Transaktion abspeichert und bei Anforderung den passenden HTML-Code generiert.

Model View Controller

Eine ganze Reihe von Tools erlauben es, entweder auf vordefinierte Layouts zur Bildschirmgestaltung zurückzugreifen oder selbst eigene Layouts

Runtime-Code wird aus Metadaten erzeugt



zu entwerfen. Aus diesen Metadaten wird dann der Runtime-Code für die verschiedenen Plattformen ABAP, Java oder Microsoft.NET erzeugt.

Entwickler codiert nur noch Ereignisse in der Anwendung

Damit schließt Web Dynpro die Lücke zwischen der klassischen SAP-Dynpro-Technik und flexibler Webentwicklung. Der Controller von Web Dynpro kümmert sich um die wiederkehrenden Arbeiten auf unterer Ebene wie die Anpassung des Codes an die Möglichkeiten des Browsers. Der Entwickler kümmert sich nur um das Ausfüllen der Code-Segmente für die jeweiligen Webevents, die bei üblichen Ereignissen ausgelöst werden, zum Beispiel beim Laden einer Seite, bei einem Mausklick oder einer Eingabe.

Web Dynpro ersetzt SAP GUI

Web Dynpro zielt auch darauf, den Zugang zu SAP-Daten von besonderen Applikationen aus möglich zu machen. Man denkt da in erster Linie an mobile Eingabegeräte wie PDAs, GSM-Telefone mit Browser oder die Bereitstellung der Dienste als Webservice. Langfristig ist das Ziel, das bestehende klassische SAP GUI durch Web Dynpro zu ersetzen.

