

Entwicklung von Informationssystemen in der Lehre — Ein Fallbeispiel der Verwendung von OSS —

Bernd Müller

Fachhochschule Braunschweig/Wolfenbüttel

Fachbereich Informatik

Am Exer 2, 38302 Wolfenbüttel

`bernd.mueller@fh-wolfenbuettel.de`

Abstract: Die Verwendung von Open-Source-Systemen in der Lehre ist gängige Praxis an deutschen Hochschulen. Wir skizzieren hier als Fallbeispiel den Einsatz von Open-Source-Systemen im Labor einer Lehrveranstaltung zu Informationssystemen, das praktisch alle relevanten technischen Aspekte von Informationssystemen abdeckt.

Das Papier versucht an diesem Fallbeispiel zu belegen, dass selbst Informationssysteme mit einem sehr großen Spektrum an technischen Anforderungen, fast ausschließlich mit Open-Source-Software zu realisieren sind.

1 Einführung

Die Lehrveranstaltung *Betriebliche Informationssysteme* im Studiengang Informatik an der Fachhochschule Braunschweig/Wolfenbüttel ist als 4 SWS Vorlesung für das 4. Semester und nach dem Praxissemester im 5. Semester als 2 SWS Labor für das 6. Semester konzipiert. Im Labor wird ansatzweise ein betriebliches Informationssystem (BIS) realisiert. Die Implementierung eines kompletten BIS ist aus offensichtlichen Gründen nicht möglich. Neben konzeptionellen Inhalten sollen exemplarisch die vielen technischen Facetten von BIS praxisnah durchgespielt werden, um die komponentenbasierte Software-Entwicklung real erfahren zu können.

Grundlage der Implementierung ist die Java Enterprise-Edition, allerdings ohne Enterprise JavaBeans. Diese sind in ihrer bisherigen Form (vor Version 3) sehr komplex und ziehen einen hohen Implementierungsaufwand nach sich, der an anderer Stelle einer knapp bemessenen Lehrveranstaltung fehlt. Um aktuelle Technologien einzusetzen, verwenden wir als GUI-Framework JavaServer-Faces und als Persistenzschicht Hibernate. Neben diesen zentralen Systemen gruppieren sich verschiedene Komponenten für die unterschiedlichen Einsatzbereiche, für die Open-Source-Implementierungen existieren. Wir stellen diese Komponenten übersichtsartig vor.

Die Lehrveranstaltung und die praktischen Übungen im Labor initiierten die Erstellung eines Lehrbuches zum Thema JavaServer-Faces. Die praktische Übungen des Buches basieren ebenfalls ausschließlich auf den noch vorzustellenden Open-Source-Systemen.

2 Die zentralen Komponenten

Der Autor betreibt ein eigenes Labor mit Linux-PCs. Teilweise findet die Veranstaltung auch in Räumen des Hochschulrechenzentrums mit Windows-PCs statt. Da die Anwendungen in Java realisiert werden, ist die Wahl des Betriebssystems unerheblich. Als Java-System wird das JDK der Java-SE-Implementierung von Sun verwendet, das sowohl für Linux als auch für Windows erhältlich ist.

Grundlage der Anwendungen ist ein Servlet-Container, der mindestens die Servlet-Spezifikation in Version 2.3 und die JSP-Spezifikation in Version 1.2 erfüllt. Wir verwenden Tomcat, den Servlet-Container der Apache-Foundation in der Version 5.5.x.

Als JavaServer-Faces-Implementierung wird MyFaces 1.1.3 verwendet. MyFaces wird von der Apache-Foundation entwickelt und hat die Tests durch den Technology Compatibility Kit (TCK) von Sun bestanden. MyFaces ist damit eine offizielle Implementierung der JavaServer-Faces-Spezifikation 1.1.

Als Datenbankmanagementsystems wird HSQLDB verwendet. HSQLDB ist eine fast vollständige Implementierung der SQL-99-Spezifikation in Java. HSQLDB verwaltet die Daten im Klartext im Dateisystem. Dadurch und durch die Verwendung von Java als Implementierungssprache eignet sich unserer Meinung nach HSQLDB sehr gut für die Lehre, da sich der Datenexport und -import zwischen dem studentischen, heimischen Arbeitsplatz und den Hochschullaboren sogar über Betriebssystemgrenzen äußerst einfach durch Kopieren von Dateien realisieren lässt. Alternative Datenbankmanagementsysteme sind etwa MySQL, Postgres, Derby, Firebird oder Interbase. Diese Auswahl ist durch die Verwendung des object-relational Mappers Hibernate vorgegeben, der die genannten Datenbankmanagementsysteme sowie alle relevanten kommerziellen Systeme unterstützt. Durch die Wahl eines OR-Mappers vereinfacht sich die Entwicklungsarbeit erheblich, da die Studierenden praktisch nur Java- und JSF-Kenntnisse und fast keine SQL-Kenntnisse benötigen.

An dieser Stelle sei zur Vollständigkeit noch erwähnt, dass JavaServer-Faces eine Servlet- und JSP-Implementierung benötigen. Diese sind in Tomcat integriert und ebenfalls Open-Source-Implementierungen der Apache-Foundation.

3 Entwicklung

Als Entwicklungsplattform verwenden wir Eclipse inklusive des Teilprojekts *Web Tools Platform (WTP)*. Dieses unterstützt das Bearbeiten von JSP-Seiten und XML-Dateien, die neben Java-Klassen den Hauptteil bei der Entwicklung von JSF-Anwendungen darstellen. Für den Build-Prozess verwenden wir Ant. Ant ist in Eclipse integriert, kann jedoch

auch stand-alone verwendet werden. Dies ist insbesondere bei Remote-Systemen ohne graphische Oberfläche von Interesse. Ebenfalls aus dem Entwicklungsprozess nicht wegzudenken ist das Logging von Systemmeldungen und -fehlern. Hier kommt *Log4J* zum Einsatz.

Seit der Version 3.0 enthält Eclipse einen eingebauten Java-Compiler, so dass für den Einsatz von Eclipse ein JRE ausreicht. Der Compiler gehört zum Java-Development-Tools-Teilprojekt (JDT) und wird daher auch JDT-Compiler genannt. Auch Tomcat benutzt seit Version 5.5 diesen Compiler für das Übersetzen von JSP-Seiten. Alternativ bietet sich Jikes als sehr schneller und nach Entwicklerrausgabe spezifikationskonformer Java-Compiler an.

Für den Aufruf des `javac`-Targets im Ant-Build-File benötigt Ant einen Java-Compiler. Hier verwenden wir allerdings nicht JDT oder Jikes, sondern den Sun-JDK-Compiler.

4 Weitere Komponenten

4.1 PDF/FOP

Informationssystem erzeugen für Druckerzeugnisse in der Regel PDF. Zur Erzeugung von PDF existieren eine Reihe von Java-Lösungen, von denen mehrere open-source sind. Wir gehen jedoch einen anderen Weg und erzeugen zunächst ein XSL-FO-Dokument (Formatting Objects), das über XSLT zu PDF gewandelt wird. XSL-FO ist eine W3C-Recommendation einer Seitenbeschreibungssprache. XSL-FO ist XML, so dass zunächst XML erzeugt werden muss. Hierfür wird *JDOM* verwendet, das ein leichtgewichtiges API für die XML-Bearbeitung darstellt. Die XSL-Transformation wird mit der JAXP-Implementierung des Sun-JDKs vorgenommen. Die JAXP-Implementierung basiert wiederum auf Xalan und Xerces der Apache-Foundation. Die Transformation nach PDF geschieht mit *FOP* (Formatting Objects Processor) von der Apache-Foundation.

4.2 Graphiken

Bei der Erzeugung von Graphiken kommen zwei Systeme zum Einsatz. Für Statistikauswertungen und -darstellungen eignet sich *JFreeChart* besonders gut, das praktisch alle Chart-Arten unterstützt. *JFreeChart* erwartet eine mehrdimensionale Datentabelle, die dann z.B. als Balken- oder Tortendiagramm dargestellt wird.

Sollen ganz allgemeine Vektorgraphiken in Web-Anwendungen dargestellt werden, so müssen diese in GIF, JPEG oder PNG überführt werden. Zwar existiert die W3C-Recommendation Scalable Vector Graphics (SVG) als Web-Standard für Vektorgraphiken, diese wird aber vom aktuellsten Firefox-Browser noch nicht vollständig implementiert und wird bei älteren Versionen oder anderen Browsern nur über ein Adobe-Plug-In unterstützt. Abhilfe schafft hier Batik, ebenfalls ein Projekt der Apache-Foundation. Batik rendert

SVG-Graphiken z.B. nach GIF, JPG oder PNG, so dass über diesen Umweg auch SVG-Graphiken problemlos in Web-Seiten eingebunden werden können.

4.3 Datenimport und -export

Informationssysteme stellen in der Regel Schnittstellen für den Datenimport und -export bereit. Standardformate sind etwa Microsofts Word und Excel. Das Apache *POI*-Projekt stellt Java-Schnittstellen für die Microsoft-Formate bereit. Im Labor verwenden wir *HSSF*, die Schnittstelle für Excel. Das Word-Äquivalent ist *HWPF*.

Abbildung 1 zeigt die verwendeten Systeme und ihre Einordnung in das Gesamtsystem anhand einer Architekturskizze.

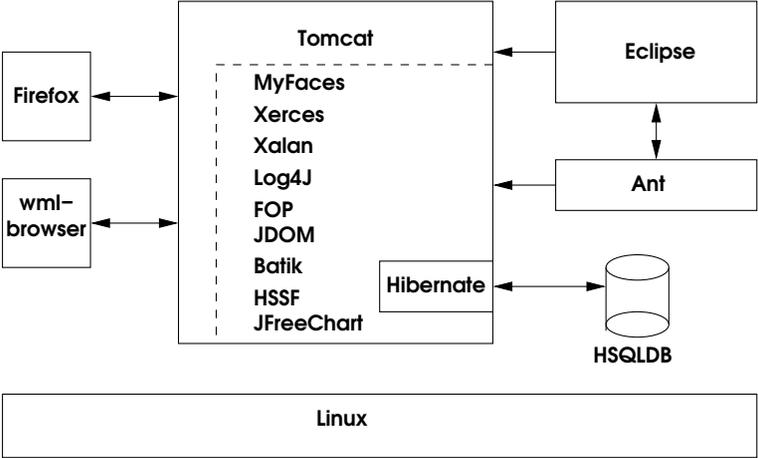


Abbildung 1: Architekturskizze der technischen Systemkomponenten

5 GUI

JavaServer-Faces rendert standardmäßig in HTML 4.01. Neben Firefox, der im Labor verwendet wird, existieren noch eine Reihe weiterer Open-Source-Browser, die z.B. im Rahmen von KDE oder Gnome entwickelt werden.

Die MyFaces-Erweiterung rendert alternativ WML (in Version 1.0.9). Auch für die Darstellung von WML existieren Open-Source-Implementierungen. Wir verwenden hier keine stand-alone Implementierung, sondern das Firefox Plug-In *wmlbrowser*.

6 Zuverlässigkeit

Die genannten Systeme zeigen im Laborbetrieb eine sehr hohe Zuverlässigkeit. Eine Ausnahme ist Tomcat, der nach wiederholtem Laden und Entfernen einer Anwendung mit dem Fehler „out of PermGen space“ abbricht. Dies ist ein bekannter Fehler, der noch nicht beseitigt werden konnte. Er tritt allerdings nur nach mehrmaligem Laden und Entfernen einer Anwendung auf und ist somit für Produktivsysteme nicht relevant. Wir möchten aber an dieser Stelle ausdrücklich betonen, dass sich die Aussage der hohen Zuverlässigkeit und damit auch des problemlosen Einsatzes auf Hochschullabore bezieht.

7 Zusammenfassung

Wir haben mit einer rein quantitativen Aufzählung von Systemen versucht deutlich zu machen, dass für viele Teilbereiche der Informationssystementwicklung Open-Source-Alternativen existieren. Die Verwendung von Open-Source-Systemen in der Lehre sollte nicht als Selbstzweck betrieben werden; wir wollen hier nicht missverstanden werden. Die Verwendung ist aber in Anbetracht der leeren Kassen der öffentlichen Hand und den damit verbundenen Einsparzwängen an den Hochschulen häufig angezeigt.

Alternative kommerzielle Systeme wurden, obwohl für alle Teilbereiche vorhanden, nicht betrachtet. Es soll hier nicht verschwiegen werden, dass viele kommerzielle Systeme für Lehrzwecke von den Herstellern unentgeltlich zur Verfügung gestellt werden und somit der oben genannte finanzielle Aspekt entfällt. Zudem verschwimmt die Grenze zwischen open-source und kommerziell. Zum einen enthält das Sun-SDK Xerxes und Xalan von Apache, IBMs WebSphere-Systeme enthalten den Apache-Web-Server und Eclipse. Zum anderen sind Eclipse und Derby eine Schenkung der IBM, ADF-Faces eine Schenkung von Oracle und Mozilla/Firefox eine Schenkung von Netscape. Der Autor besitzt jedoch eine gewisse Affinität zu Open-Source-Systemen, die das Fallbeispiel motiviert hat.

Unsere Erfahrung ist, dass die verwendeten Systeme durchweg von sehr hoher Qualität sind. Ein Großteil der Systeme wird von der Apache- bzw. Eclipse-Foundation entwickelt, die für eine sehr hohe Software-Qualität bekannt sind. Zumindest was den Bereich der Informationssystementwicklung angeht, kann aus technologischer Sicht die Aussage getroffen werden, dass große Teile des Gesamtspektrums mit Open-Source-Systemen abgedeckt werden können.

Der Umfang der beschriebenen Lehrveranstaltung lässt weitere Implementierungsarbeiten leider nicht zu. Es existieren aber auch in den nicht angesprochenen Teilbereichen der Informationssystementwicklung, wie etwa Namensdienste, Lastverteilung, Ausfallsicherheit und Kryptographie eine ganze Reihe von Open-Source-Implementierungen, deren Einsatz in der Lehre noch zu evaluieren ist.

8 Verwendete Systeme

Tabelle 1 gibt einen Überblick über die verwendeten Open-Source-Systeme. Basissysteme wie Linux, Firefox und das (nicht open-source) JDK von Sun sind nicht aufgeführt. Insgesamt finden 17 Open-Source-Systeme Verwendung.

System	Beschreibung	URL
Tomcat	Servlet-Container	http://tomcat.apache.org
MyFaces	JSF-Implementierung	http://myfaces.apache.org
HSQldb	Pure-Java DBMS	http://www.hsqldb.org
Hibernate	OR-Mapper	http://www.hibernate.org
Xerces	XML-Parser	http://xerces.apache.org/xerces2-j
Xalan	XSL-Transformator	http://xml.apache.org/xalan-j
Eclipse	Entwicklungsplattform	http://www.eclipse.org
Eclipse-WTP	Eclipse für JSP/Web/XML	http://www.eclipse.org/webtools
Ant	Build-Werkzeug	http://ant.apache.org
Log4J	Logging-System	http://logging.apache.org/log4j
FOP	XSL-FO Verarbeitung	http://xmlgraphics.apache.org/fop
JFreeChart	Chart-Bibliothek	http://www.jfree.org/jfreechart
JDOM	XML-API	http://www.jdom.org
Batik	SVG-Verarbeitung	http://xmlgraphics.apache.org/batik
HSSF	Excel-API	http://jakarta.apache.org/poi/hssf
Firefox	Web-Browser	http://www.mozilla.com
wmlbrowser	Firefox-Plug-In für WML	http://wmlbrowser.mozdev.org

Tabelle 1: Die verwendeten Open-Source-Systeme