

Praxissemesterbericht 2 Computer Networking Orientation in Objects - Mannheim

Benjamin Bratkus, Matr.Nr. 215180, benscho@benscho.org
FH Furtwangen
Sommersemester 2005

Powered by L^AT_EX

11. Oktober 2005

Inhaltsverzeichnis

1 Die Firma Orientation in Objects	4
1.1 Übersicht	4
1.2 Aufgabengebiet	4
1.3 Arbeitsumfeld	4
2 Portallösungen mit Open Source Portalen und Portletentwicklung nach JSR 168	5
2.1 Einleitung	5
2.2 Portale	5
2.3 Portlet Entwicklung mit dem JSR 168	6
2.4 Portlet Container Pluto	6
2.5 JBoss Portal 2.0	6
2.6 Jetspeed2	7
2.7 Abschluss	8
3 JPdfUnit	9
3.1 Einleitung	9
3.2 JUnit	9
3.3 PDFBox	10
3.4 Log4j Project	10
3.5 JPdfUnit - Make your pdf green!	10
3.6 Abschluss	12
4 Projektarbeit	13
4.1 FrontEnd Entwicklung mit Struts und Common Control	13
4.2 Freudenberg IT	13
5 Fazit	14

Vorwort

Dieses Dokument entstand im Rahmen des zweiten Pflichtpraktikum des Studiengangs Computer Networking der Fachhochschule Furtwangen im Sommersemester 2005. In den nachfolgenden Kapiteln wird auf die Aufgabengebiete innerhalb der Firma OIO¹ in Mannheim eingegangen. Unternehmensspezifische Informationen werden nicht veröffentlicht.

¹Orientation in Objects

1 Die Firma Orientation in Objects

1.1 Übersicht

Die Firma Orientation in Objects GmbH ist seit einigen Jahren in der Software Entwicklung, speziell im Open Source Bereich, sowie der Entwicklung von J2EE-Anwendungen tätig. Neben der Entwicklung kundenspezifischer Softwarelösungen bietet das kleine mittelständische Unternehmen ein ebenso umfangreiches Angebot an Schulungen, die sowie intern als auch extern gehalten werden. Hinzukommen individuelle Coachings, die entweder durch Projekte oder direkte Anfragen von Kunden entstehen.

1.2 Aufgabengebiet

Am Anfang des Praxissemesters stand die Entwicklung mit J2EE Komponenten auf Open Source Portalen unter Einsatz von Servern und Containern, wie dem Apache Tomcat. Anschliessend kam die Entwicklung eines Java Frameworks für automatische Dokumententests. Die ständige Begleitung und den Abschluss bildete die Mitarbeit in verschiedenen externen Projekten, bei denen sowohl J2EE Komponenten, sowie natives Java eingesetzt wurden.

1.3 Arbeitsumfeld

Die Ausstattung des Arbeitsplatzes war zufriedenstellend. Ebenso war ein persönlicher Kontakt zu den Kollegen möglich, was den Einstieg in die Firma sehr erleichterte, auch war das Arbeitsklima stets korrekt und gut. Neben den festen Aufgabengebieten war es möglich das breite Schulungsangebot der Firma zu nutzen und so zusätzliches Wissen aufzubauen, was sehr positiv gewertet werden kann. Es konnten die nachfolgenden Schulungen besucht werden:

- „Das Buildtool Apache Ant“
- „Das Buildtool Apache Maven“
- „Effective Java“
- „Java Management Extensions JMX“
- „Servlets und JavaServer Pages JSP“
- „Tomcat Administration und Entwicklung“
- „Webanwendungen mit Apache Struts“

2 Portallösungen mit Open Source Portalen und Portletentwicklung nach JSR 168

„Ein Portal ist eine Applikation welche basierend auf Webtechnologien einen zentralen Zugriff auf personalisierte Inhalte sowie bedarfsgerecht auf Prozesse bereitstellt. Charakterisierend für Portale ist die Verknüpfung und der Datenaustausch zwischen heterogenen Anwendungen über eine Portalplattform. Eine manuelle Anmeldung an den in das Portal integrierten Anwendungen ist durch Single Sign On nicht mehr notwendig, es gibt einen zentralen Zugriff über eine homogene Benutzungsoberfläche. Portale bieten die Möglichkeit, Prozesse und Zusammenarbeit innerhalb heterogener Gruppen zu unterstützen.“ [1, Fraunhofer IAO]

2.1 Einleitung

Das erste Hauptaufgabengebiet bestand in der Auseinandersetzung mit Open Source Portalen und dem Standard für Portletentwicklung. Hierzu wurde das neue Open Source Portal der Apache Group - Jetspeed2[2], sowie das JBoss Portal 2.0[3] der JBoss Community durch die Entwicklung von Portlets auf Ihre produktive Einsatzfähigkeit getestet. Desweiteren sollte durch das angeeignete Wissen anschliessend ein offenes Seminar entstehen.

2.2 Portale

Durch die Veröffentlichung des JSR 168 dem Standard für die Portlet Entwicklung, war es unabdingbar, Open Source Portale auf ihre Einsatzfähigkeit mit dem Standard zu testen. Durch das firmeninterne Knowhow lag die Wahl der Open Source Portale, die in Frage kamen, bei einer Portallösung der JBoss Community, sowie der Apache Group. Auch die Arbeit mit den gruppenspezifischen Servertechnologien, beispielsweise dem Tomcat¹ und JBoss², war unabdingbar und erwies durch firmeninternes Wissen als unproblematisch. Bei beiden Portal-Projekten wurde an der Benutzer - und Entwicklergemeinde teilgenommen, um so schnell wie möglich Wissen um die Technologie aufzubauen und einsetzen zu können. Hierbei konnte sehr gut beobachtet werden, dass sich die Portallösungen generell immer durch Ihre Deskriptoren und der Struktur der Server auf Dateiebene unterscheiden. Diese Deskriptoren sind Metainformationen in Form von XML³-Dateien, die ein Portal zur Beschreibung einer Applikation benötigt. Desweiteren konnte ein Teil der Benutzer Dokumentation zu Jetspeed2 verfasst werden, die dann den anderen Benutzern über ein Knowledge Management System, in diesem Fall ein Wiki, zugänglich gemacht werden konnte.

¹J2EE Kontainer

²Anwendungsserver

³eXtensible Markup Language

2.3 Portlet Entwicklung mit dem JSR 168

Die Entwicklung von Portlets basiert auf der allseits bekannten Webtechnologie Servlets, nur ist diese auf Portale ausgelegt. Somit war die Notwendigkeit gegeben, einen Standard zu definieren, der die Kommunikation und die Interaktion der Komponenten, hier den Portlets und den umgebenden Kontainern beschreibt. Hierzu wurde der JSR 168 [4] definiert. Anschliessend war es möglich selbst einige Portlets zu entwickeln und diese Anwendungen in die Portale zu integrieren.

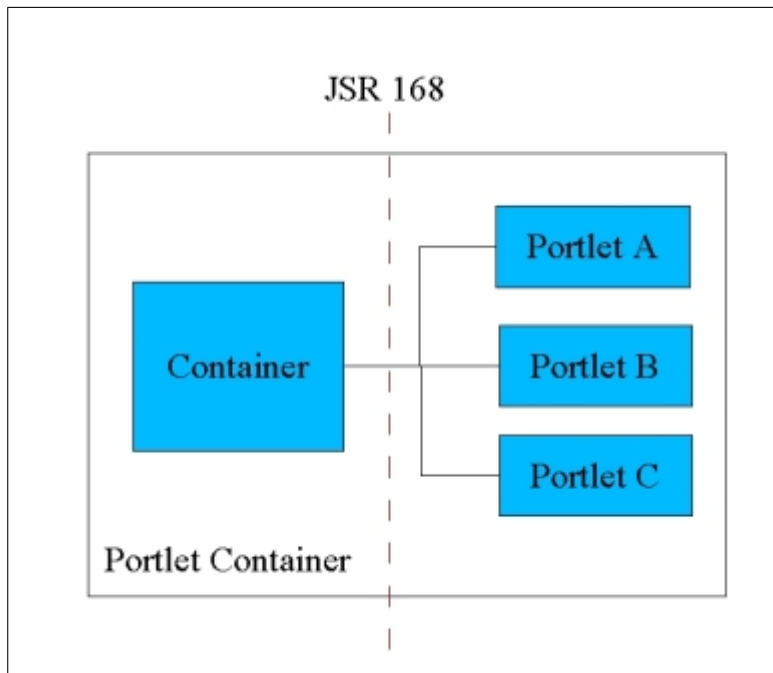


Abbildung 1: Kommunikation Portlet Kontainer und Portlets

2.4 Portlet Container Pluto

Für eine effektive Arbeit mit Portalen und einer standardisierten Entwicklung von Portlets ist es nahezu notwendig, dass sich die Portale an der Referenzimplementierung eines Portlet Containers halten. Die Referenzimplementierung eines Portlet Containers ist PLUTO[5] der Apache Group genommen. Die entwickelten Portlets wurden vor Ihrem Einsatz in den Zielportalen mit Hilfe dieses Containers entwickelt und anschliessend getestet. Erst nach einem erfolgreichem Testlauf in die Portale integriert.

2.5 JBoss Portal 2.0

Zeitlich etwas verzögert startete neben der Apache Group ebenso JBoss an der Entwicklung eines Portals, das den neuen Standard für Portlets unterstützen sollte. Wie auch bei Jetspeed2 ist es nahezu ohne grossen Aufwand und Anpassungen möglich die standardkonformen Portlets in dem JBoss Portal 2.0 einzu-

setzen. Es werden hierzu ebenso weitere Deskriptoren in XML benötigt um die nach Standard entwickelten Portlets einzusetzen. Auch in den administrativen Möglichkeiten bietet JBoss Portal eine Vielzahl an Möglichkeiten, beispielsweise zur Benutzerverwaltung.

2.6 Jetspeed2

Die erste Technologie mit der gearbeitet wurde, war das Portal Jetspeed2 der Apache Group, welches ebenso den Portlet Container Pluto integriert und somit ein referrenznahes Portal darstellt. Somit beinhaltet das Portal die Funktionalitäten, die laut Spezifikation an ein Portal gestellt werden können. Es bietet Möglichkeiten zu der Benutzer-, Gruppen-, und Rollenverwaltung die auf jeden einzelnen Benutzer angepasst werden können. Ebenso ist die Funktionalität zur Verwaltung der portalinternen Applikationen gegeben, welche stark an die Art und Weise eines Web-oder Applikationsservers erinnert. Auch die ersten Erfolge in der Entwicklung eigener Portlets konnten auf diesem Portal verzeichnet werden. Die Datenhaltung des Portals kann auf die unterschiedlichsten Möglichkeiten realisiert werden, jedoch kommt Jetspeed2 standardmässig mit einer Hypersonic DB Datenbank. Im Umfang der Arbeit kann gesagt werden, dass Jetspeed2 den Schwerpunkt bildete und sogar soweit modifiziert wurde, dass es als Firmenportal genutzt werden könnte.

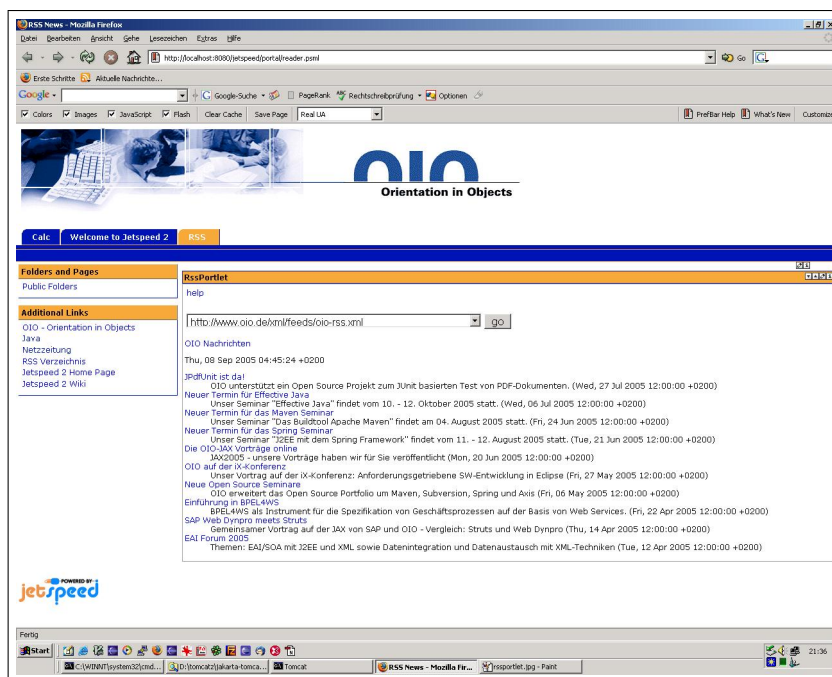


Abbildung 2: Screenshot: Rss Portlet in Jetspeed2

2.7 Abschluss

Man kann sagen, dass auf allen Bereichen Erfolge sichtbar erreicht waren. Additional zu dem gewonnenen Wissen ist eine Arbeit im Jetspeed2 Dokumentationsprojekt[16] sowie ein Austausch eigens entwickelter Portlets mit den Projekt Mitgliedern der JBoss Portal Entwicklung einhergegangen. Desweiteren wurde eine Seminarbeschreibung für die Firma OIO verfasst, auf deren Basis beginnend ab dem Herbst 2005 eine Schulung im Bereich Portletentwicklung und Portaltechnologien stattfinden kann.

3 JPdfUnit

3.1 Einleitung

Einleitend kann gesagt werden, dass die Vielfalt an Applikationen, ein ebenso grosses Mass an Nachfolgeprodukten mit sich bringt. Durch eine Java Webapplikation, die sowohl für den Benutzer als auch den Administrator PDF Dokumente als Output generieren kann, entstand die Anforderung, dass es möglich sein muss, diese automatisch erzeugten PDF Dokumente ohne die Einsicht eines Menschen auf Ihre Richtigkeit bezüglich der Form und dem Inhalt zu testen. Hierzu entstand die Idee, das Testframework JUnit mit einer PDF Bibliothek zu einem Framework zu kombinieren und so die automatisch erzeugten Dokumente zu testen.

3.2 JUnit



Abbildung 3: Logo: JUnit

Das JUnit[7] Framework ist eine Produkt verschiedener Größen der Java Entwicklung. JUnit ermöglicht dem Benutzer Objekte in sogenannten Tests zu erzeugen und mit diesen Objekten den eigenen Source Code zu testen. Durch diese Tests wird der Entwickler schon vor der Auslieferung seiner Software auf mögliche Fehlerquellen in seinem Programmfluss aufmerksam. Ebenso wird die Stabilität der Software durch worst case Testfälle auf die Probe gestellt. Die nächste Stufe der Softwareentwicklung mit JUnit ist das testdriven development, was bedeutet, dass der Entwickler vor der eigentlichen Implementierung seiner Software die Tests für, im Falle von Java, Methoden und Klassen schreibt um ein höchstes Mass an Sicherheit und Stabilität zu gewährleisten.

3.3 PDFBox



Abbildung 4: Logo: PDFBox

Für den eleganten Zugriff auf ein PDF Dokument mit Hilfe von Java Klassen wurde die PDF Bibliothek PDFBox[8] ausgewählt. Neben den anderen möglichen Kandidaten erwies sich PDFBox im Umgang mit bestehenden PDF Dokumenten als der Topkandidat. PDFBox ermöglicht dem Benutzer dieser Bibliothek eine grosse Menge an Informationen zu dem Dokument ansich, sowie dem Inhalt zu erhalten. Als Entscheidungskriterika für eine Bibliothek konnten folgende Punkte gefunden werden:

- Das Lizenzrecht
- Die Aktivität der Gemeinschaft
- Die Häufigkeit und das Erscheinen von neuen Versionen
- Die das Erfüllen funktionaler Anforderungen

3.4 Log4j Project

Das Log4j project[9] wurde durch eine Abhängigkeit der Pdfbox in das Project integriert. Es bietet komfortable Möglichkeiten das Logging einer Software oder einer API zu managen. JPdfUnit selbst benutzt das Log4j Projekt selbst nicht, das die Fehlerhandhabung primär durch RuntimeExceptions regelt und somit dem Benutzer zur Laufzeit über Fehler und Fehlerquellen informiert. Diese projektspezifische Abhängigkeit konnte jedoch nach dem zweiten Release zurückgerufen werden.

3.5 JPdfUnit - Make your pdf green!



Abbildung 5: Logo: JPdfUnit

JPdfUnit[12] kann als erfolgreich abgeschlossenenens Projekt betrachtet werden. Eine Recherche, Analyse, Implementierung, sowie Tests und Wartung mit anschliessender Vermarktung wurden durchgeführt, dabei wurde der vorgegebene Zeitplan eingehalten.

- Problemanalyse Zu Beginn des Projekts wurde eine Anforderungsanalyse betrieben. Es wurde die gewünschten Funktionalitäten der Projektbeteiligten gesammelt und erfasst. Anschliessend wurde ausreichend Zeit in die Arbeit mit JUnit, somit der Entwicklung von Testfällen und Test,

und der Auswahl der PDF Bibliothek investiert, um sicherzugehen, dass die gewünschte Funktionalität auch angeboten werden kann. Ebenso wurde auch die Version des JSK beraten und diskutiert, wobei durch die Abhängigkeiten zu den anderen Bibliotheken die Wahl auf die Version 1.4 fiel.

- Entwurf Anschliessend an die Analyse wurde beraten und entschieden, wie die Funktionalität angeboten und implementiert werden kann. Dazu wurden Vergleiche zu bestehenden JUnit Erweiterungen gezogen und versucht ein höchstes Mass an Flexibilität und Funktionalität anzubieten. Letztendlich war es durch die intensive Design Phase möglich, die selben Funktionalitäten bezogen auf das Themengebiet anzubieten und sogar andere Erweiterungen zu übertreffen. Diese Phase der Entwicklung stellte den grössten Aufwand an Zeit und Ressourcen dar, da hierzu ein hohes Mass an Recherche, wie Kommunikation notwendig war.
- Implementierung Die Implementierung des Projekts konnte aufgrund von intensiven Versuchen im Vorfeld und der Analyse schnell beendet werden. Ein grosser Teil der Funktionalität wurde bereits im Vorfeld getestet, um den Projektanforderungen zu genügen und eine Abschätzung zu ermöglichen und zu tätigen, ob ein solches Framework generell realisierbar ist. Zu der Funktionalität der API kam ein für JUnit Erweiterungen typisches intensives Testen der eigenen Implementierung hinzu. Diese Tests wurde zu der eigentlichen Implementierung hinzugefügt, um dem Anwender ein grosses Spektrum an Beispielen zu bieten. Für eine effektive und saubere Arbeit während der Entwicklung wurde Eclipse verwendet und die Erweiterungen PMD[11] und Checkstyle[12] in die Entwicklungsumgebung integriert. Für das Buildmanagement wurde Apache Ant¹ verwendet.
- Test Anknüpfend an die Implementierung des Frameworks wurden ausgiebige Tests entwickelt. Die, für ein Testingframework, natürlich ein wichtiger Bestandteil sind. Hierzu wurde die Funktionalitäten der einzelnen Klassen durch verschiedene generative Dokumente getestet.
- Wartung und Veröffentlichung Nach der Erstellung der initialen Version wurde versucht, das Projekt im WWW², sowie bei verschiedenen Gemeinschaften bekannt zu machen. Hierzu wurde eigenes auf Sourceforge.net ein Open Source Projekt angelegt, welches voll ausgestattet als Basis dienen soll. Eine Homepage, Foren, Emailverteiler sowie eine Versionsverwaltung, um einige Teilkomponenten zu nennen, wurden erstellt und zur Benutzung freigegeben. Die Homepage des Projektes umfasst eine ausführliche Dokumentation, eine Architekturübersicht, sowie Kontaktmöglichkeiten zu den Projektmitgliedern, beispielsweise den Foren. Anschliessend wurde Kontakt zu den Gemeinschaften der in Relation stehenden Projekten hergestellt, um das Projekt bekannt zu machen. Schon zwei Tage nach dem initialen Start des Projekts kamen die ersten Wünsche von Benutzern, die auch schon teilweise realisiert werden konnten, was aber als optional eingestuft wurde.

¹build tool - ant.apache.org

²World Wide Web

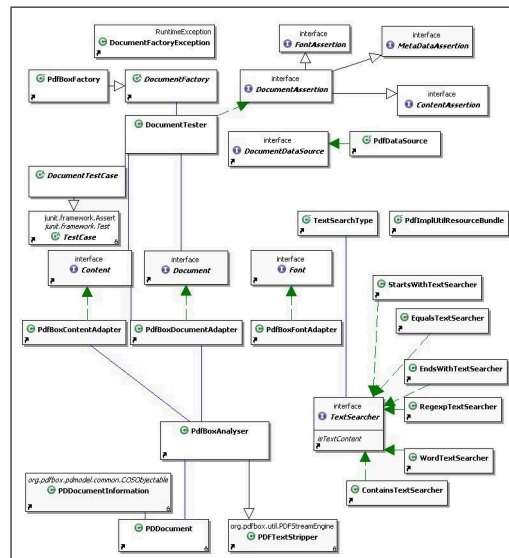


Abbildung 6: Architekturübersicht: JPdfUnit

3.6 Abschluss

Durch die termingerechte Fertigstellung des Frameworks sowie die rechtzeitige Veröffentlichung der Homepage auf SourceForge lässt sich sagen, dass die initiale Version von JPdfUnit erfolgreich gestartet ist. Die Arbeit war sehr abwechslungsreich und hatte einen sehr hohen pädagogischen Wert. Die anfängliche Resonanz war sehr positiv. Ein wichtiger Aspekt bei der Veröffentlichung des Projektes war die Bekannmachung desselben durch die Vorstellung von JPdfUnit bei anderen Testanwendungen, sowie bei den Abhängigkeiten (PdfBox, JUnit, Log4j). Nach lediglich zwei Tagen wurden bereits Erweiterungswünsche durch Benutzer geäußert und auch implementiert. Natürlich soll das Projekt sofern es die Zeit zulässt erweitert und verbessert werden. Final konnte ausserdem ein Eclipse Plugin, sowie zwei neue Releases veröffentlicht werden.

4 Projektarbeit

Neben den zugeteilten Aufgaben war es ausserdem möglich in verschiedenen Projekten der Firma OIO für Kunden mitzuwirken.

4.1 FrontEnd Entwicklung mit Struts und Common Control

Enleitend für die FrontEnd Entwicklung mit Struts[18] und Common Control[17] konnte das Seminar - Struts - der Firma OIO besucht werden, wodurch das essentiell wichtige Knowhow erworben werden konnte. Anschliessend war es möglich nach einer Einführung in Common Control an der FrontEnd Entwicklung mitzuwirken. Im Rahmen des Projekts wurden klassische Komponenten einer Webanwendung, wie Listen ,Suchfunktionalität, das Anlegen oder auch das Editieren von Objekten der Anwendung sowie Zugriffsreglement auf die verschiedenen Seiten realisiert.

4.2 Freudenberg IT

Durch einen Dienstleistungsauftrag der Freudenberg IT KG war es möglich an diesem Projekt grundlegend mitzuwirken. Es handelte sich hierbei um eine Drei-Schichten-Applikation, die über Torque[13] auf eine MSSQL-Datenbank zugreift, eine Geschäftslogikimplementierung besitzt und der Benutzer über WebDynpro[14] interagiert. Die Aufgabe bestand hierbei die WebDynpro Benutzeroberfläche mit den benötigten Daten zu versorgen und für eine transaktionssichere Speicherung Sorge zu tragen. Die Problematik lag hierbei bei der existierenden Datenbank, die durch Zeit und die damit verbundenen neuen Anforderungen unkontrolliert gewachsen war. Dennoch kann gesagt werden, dass trotz der schwierigen Voraussetzungen auch hier Erfolge verbucht werden konnten.

5 Fazit

Zu der Arbeit in diesem Praxissemester lässt sich sagen, dass dieses Sommersemester einen grossen Wissensaufbau ermöglicht hat. Es war eine gute Einführung in das ingenieurstypische Arbeiten. Die Zusammenarbeit mit den Ingenieuren der Firma Orientation in Objects war hervorragend. Das Aufgabenfeld war abwechslungsreich und interessant, so dass die internen und auch die externen Projekte erfolgreich bearbeitet und beendet werden konnten.

Literatur

- [1] Fraunhofer-Institut für Arbeitswissenschaften und Organisation, IAO (Hrsg.): [http://de.wikipedia.org/wiki/Portal_\(Informatik\)](http://de.wikipedia.org/wiki/Portal_(Informatik))
- [2] Jetspeed 2 :
<http://portals.apache.org/jetspeed-2/>
- [3] JBoss Portal 2.0 :
<http://www.jboss.com/products/jbossportal>
- [4] JSR 168 :
<http://jcp.org/aboutJava/communityprocess/final/jsr168/index.html>
- [5] Portlet Container Pluto :
<http://portals.apache.org/pluto/>
- [6] Linwood J., Minter D. :
Building Portals with the Java Portlet API, apress, 2004
- [7] JUnit Framework :
<http://junit.org> <http://junit.sourceforge.net/doc/testinfected/testing.htm>
- [8] PDFBox :
<http://pdfbox.org>
- [9] log4j :
<http://logging.apache.org/log4j/docs/>
- [10] PMD :
<http://pmd.sourceforge.net>
- [11] Checkstyle :
<http://checkstyle.sourceforge.net>
- [12] JPdfUnit :
<http://jpdfunit.sourceforge.net>
- [13] Torque :
<http://db.apache.org>
- [14] WebDynpro
<http://www.oio.de/sap-was-architecture.htm> <http://www.oio.de/web-dynpro-the-new-sap-face.htm> <http://www.oio.de/sap-netweaver-components.htm>
- [15] Tomcat
<http://jakarta.apache.org/tomcat/>
- [16] Jetspeed2 Dokumentation <http://wiki.apache.org/portals/Jetspeed2>
- [17] Common Control
<http://www.common-controls.com>

[18] Struts
<http://struts.apache.org>