

# KOMPLEXNÍ PREZENTAČNÍ VRSTVA PORTÁLOVÝCH ŘEŠENÍ NA BÁZI TECHNOLOGIE PORTLETŮ A SPECIALIZOVANÝ NÁVRHOVÝ VZOR PRO JEJÍ IMPLEMENTACI

**Ivo Martiník**

Ekonomická fakulta VŠB-TU Ostrava, Sokolská třída 33, 701 21 Ostrava 1, ČR,  
ivo.martinik@vsb.cz

## **ABSTRAKT:**

Príspevek se zabývá implementačními aspekty specializovaného návrhového vzoru Brick-Box Model-View-Controller a jeho použitím při realizaci prezentační vrstvy portálových řešení na bázi technologie Java portletů. Uvedený návrhový vzor byl prakticky využit při realizaci portálového řešení EkF VŠB-TUO s cílem sjednocení návrhu a implementace prezentační vrstvy v prostředí tenkého a tlustého klienta.

## **KLÍČOVÁ SLOVA:**

Portál, portlet, Model-View-Controller, prezentační vrstva, tenký klient, fragment

## **1. ÚVOD**

Java EE technologie (viz. [4]) se stále více uplatňují i v oblasti realizace komplexních portálových řešení. Portál lze v této souvislosti chápat jako soubor komponent a technologií, které umožňují zejména:

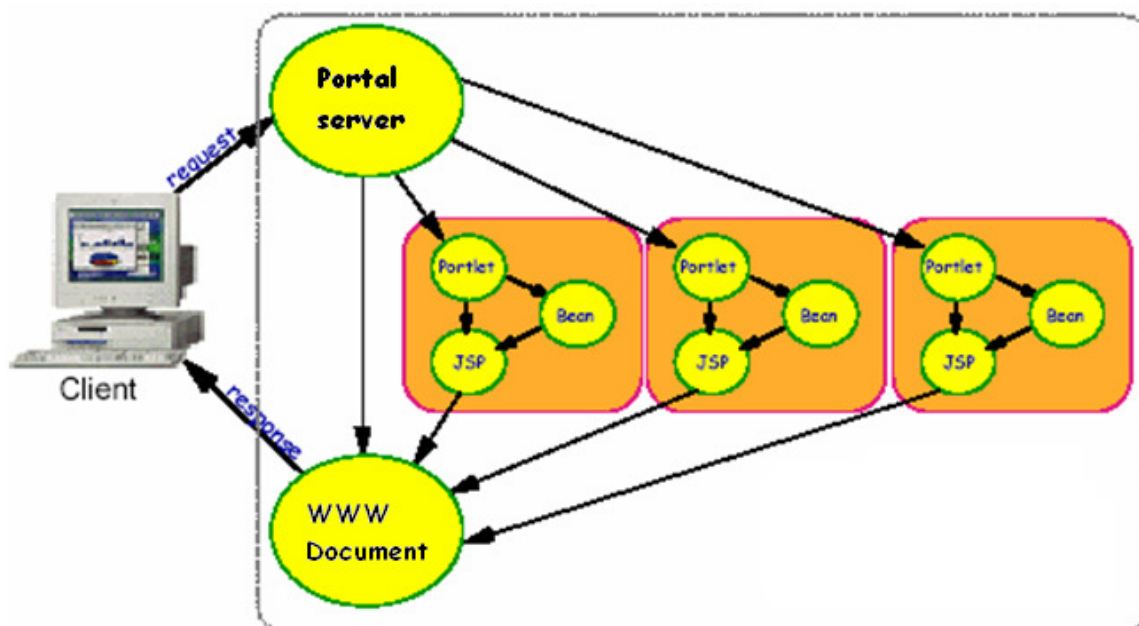
- Presentaci statických i dynamických informací prostřednictvím WWW rozhraní s využitím technologie portletů (viz. [1], [2])
- Personalizovaný přístup uživatelů k informacím
- Integraci aplikací
- Autentizaci a autorizaci včetně delegování administrace
- Propojení s externími informačními zdroji (databáze, adresářové služby)
- Řízenou publikaci informačního obsahu a jeho udržování v aktuální podobě

Portletem rozumíme web komponentu vytvořenou v programovacím jazyce Java, která není samostatně spustitelná a je po celou dobu svého životního cyklu umístěna v prostředí portletového kontejneru (*portlet container*), přijímá požadavky ze strany klienta a na jejich základě dynamicky generuje fragment WWW dokumentu. Fragmentem rozumíme část WWW dokumentu splňující jistá pravidla, jež spolu s ostatními fragmenty vytváří kompletní WWW dokument. Tento fragment vygenerovaný příslušným portletem je obvykle na úrovni portletového kontejneru (příp. portálu) sloučen podle přesně stanovených pravidel s fragmenty vygenerovanými dalšími portlety příslušnými danému WWW dokumentu. Portlety jsou v rámci portálového řešení tedy typicky využívány k vytváření prezentační vrstvy informačních systémů.

Jedním z významných návrhových vzorů, jež se již mnoho let výrazně uplatňují při návrhu a implementaci nejen informačních systémů, představuje *Model-View-Controller*

(MVC). Koncept MVC je od jeho samotných počátků svázán s paradigmatem objektově-orientovaného programování, technologie bylo poprvé využito v knihovnách tříd programovacího jazyka Smalltalk (viz. např. [5]), její výskyt je typický při tvorbě grafického uživatelského rozhraní a značnou popularitu si rovněž získala v knihovnách tříd Swing programovacího jazyka Java.

Návrhový vzor *Model-View-Controller* lze samozřejmě uplatnit i v souvislosti s vícevrstevnými distribuovanými programovými systémy s využitím konceptů Java EE architektury, kde portlet tradičně vystupuje v roli *controlleru*, komponenta *view* je typicky implementována technologií Java Server Pages (JSP – viz. [6]) a v úloze komponenty *model* nalezneme instance tříd Java Bean, příp. Enterprise Java Bean – viz. obr. 1.



Obr. 1: Model-View-Controller v prostředí portálových řešení

## 2. FAKULTNÍ PORTÁL JAKO ORGANICKÁ SOUČÁST UNIVERZITNÍHO PORTÁLOVÉHO ŘEŠENÍ

V době vzniku příspěvku je na pracovišti jeho autora realizován projekt nové verze veřejné i neveřejné části fakultního portálu, který:

- je plně v souladu se schválenými zásadami pro tvorbu veřejné i neveřejné části portálu univerzity,
- důsledně využívá jednotné univerzitní báze dat,
- je implementován s využitím Java EE technologií s důrazem na maximální stabilitu a škálovatelnost řešení vzhledem k vysokému počtu uživatelů,
- je provozován v distribuovaném prostředí centrálních systémů fakulty i univerzity s využitím služeb jednotného databázového prostředí univerzity,

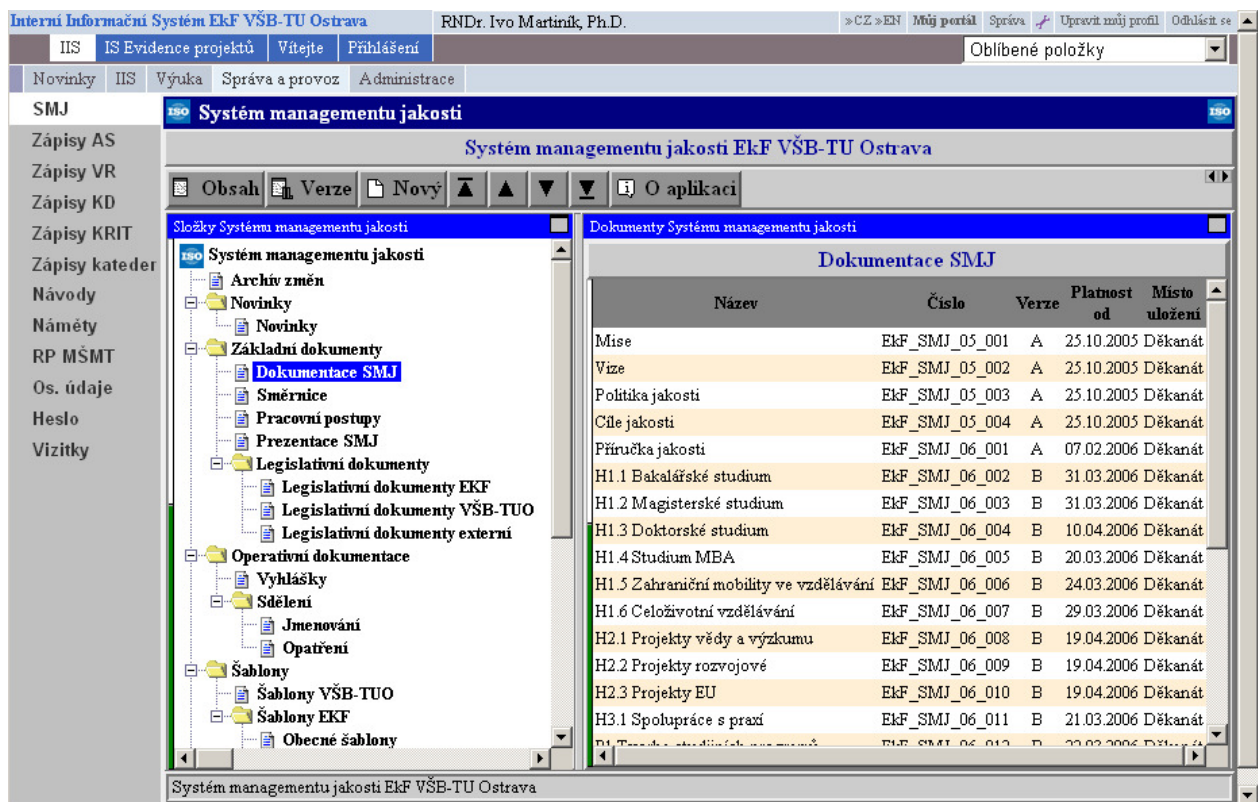
- využívá jednotného systému autentizace a autorizace portálového prostředí a přístup k jednotlivým jeho modulům na základě příslušnosti uživatele k definovaným uživatelským rolím.

Pro realizaci neveřejné části portálu byla stanoveny následující požadavky:

- Implementace bude nezávislá na hardwarové a softwarové platformě strany klienta i serveru.
- Veškeré uživatelské rozhraní bude dostupné v prostředí WWW prohlížeče bez nutnosti instalace přídatných plug-in komponent (zejména Java Virtual Machine, Flash apod.) s dosažením maximálního komfortu.
- Uživatelské rozhraní na straně tenkého klienta bude navrženo v duchu designu známého z grafického rozhraní standardních operačních systémů (Windows XP, Linux), kdy uživatelé pracují s jednotlivými moduly informačního systému podobným způsobem, jako s aplikacemi nad operačním systémem umístěnými v oknech grafického uživatelského rozhraní. (viz. obr. 2 a 3).
- Prostředí informačního systému bude integrováno s jinými systémy využívanými pedagogy a studenty fakulty.
- Provoz neveřejné části bude realizován formou virtuálního portálu v rámci jednotného univerzitního portálového prostředí. Uživatelé budou moci vstupovat do prostředí dalších provozovaných virtuálních portálů podle přidělených přístupových práv bez nutnosti opakování procesu autentizace a autorizace.
- Implementace bude respektovat specifikaci JSR-168 pro tvorbu Java portletů a bude realizována s využitím technologií Java SE (viz. [3]) a Java EE (viz. [4]).

Pro vlastní implementaci neveřejné části portálu jsou využity následující technologie:

- *Prezentační vrstva* řešení je implementována s využitím technologií XHTML, JavaScript, Document Object Model (DOM – viz. [7]) a Cascading Style Sheets (CSS – viz. [8]). Vzhledem k rozdílným implementacím těchto technologií v prostředí jednotlivých WWW prohlížečů (např. MS Internet Explorer v. 6 implementuje CSS Level 1 a DOM Level 1, Mozilla Firefox v. 1.0.7 implementuje CSS Level 2 a DOM Level 2, apod.) se autoři rozhodli pro implementaci prostředí, jež bude provozovatelná ve WWW prohlížečích verzí MS Internet Explorer v. 6 a Mozilla Firefox v. 1.0.7 a vyšších verzích (jež jsou dostupné na běžně užívaných platformách) s cílem dosažení maximálního uživatelského komfortu připomínajícího práci v grafickém prostředí operačního systému podporujícího současný běh několika úloh – viz. obr. 2. a 3.
- *Web a aplikační vrstva* implementace je realizována v prostředí portálového serveru IBM WebSphere Portal Server (viz. [9]) s podporou technologie Java EE v. 1.4 a portletů (IBM WebSphere v. 6 je certifikovaný aplikační server pro uvedené standardy – viz. [10]). Primárně jsou pak využity technologie Java servletů a Java portletů bez nutnosti nasazení standardu Java Server Pages.



Obr. 2: Prezentační vrstva neveřejné část portálu

- **Databázová vrstva** je tvořena databázovým systémem IBM DB2 (v našem případě tradičně provozovaná na platformě midrange centrálních systémů řady IBM iSeries – viz. [11]) s využitím produktu IBM DB2 Content Manager (viz. [12]), který plní roli univerzitního systému pro správu dokumentů a ostatních zdrojů multimediálního charakteru a mezi jehož základní funkcionality řadíme zejména:
  - Ukládání (*storage*) – existence rozhraní (programy, knihovny API) umožňujícího ukládání dokumentů obecně multimediálního charakteru spolu s množinou metadat popisujících obsah a charakteristiky daného dokumentu. Optimálně je požadována i možnost verzování dokumentů.
  - Podpora pracovního toku dokumentů (*workflow*) – automatická dostupnost dokumentů přesně definovaným uživatelům nebo skupinám uživatelů v jednotlivých fázích jeho životního cyklu.
  - Vyhledávání (*search*) – možnost vyhledávání v metadatech dokumentů i full-textové prohledávání obsahů uložených dokumentů.
  - Výběr (*retrieval*) – možnost přístupu k obsahu dokumentů, příp. provádění převodu dokumentů do jiných formátů.
  - Distribuce (*distribute*) – podpora práce v distribuovaném výpočetním prostředí a umožnění efektivního přístupu ke službám systému správy dokumentů v jednotlivých geografických lokalitách.

- Kontrola přístupu (*access control*) – kontrola zda příslušný uživatel systému (autentikovaný nebo neautentikovaný) má právo provádět dané akce nad úložištěm multimediálních dokumentů.
- Indexování (*index*) – možnost tvorby indexů nad metadaty i plnými texty dokumentů pro urychlení procesu vyhledávání.

### 3. SPECIALIZOVANÝ NÁVRHOVÝ VZOR BRICK-BOX MODEL-VIEW-CONTROLLER PRO IMPLEMENTACI PREZENTAČNÍ VRSTVY NEVEŘEJNÉ ČÁSTI FAKULTNÍHO PORTÁLU

Pro vlastní implementaci prezentační vrstvy neveřejné části portálu je využito speciálně pro tento účel vyvinutého návrhového vzoru *Brick-Box Model-View-Controller*, v němž v roli komponenty *controller* vystupuje instance Java portletu, komponenta *model* je reprezentována instancí třídy Enterprise Java Bean (příp. Java Bean) provádějící interakci s databázovým systémem a komponenta *model* není v tomto případě reprezentována tradičním způsobem instancí stránky JSP, ale objektem splňujícím specifikace standardu pro Java Beans.

|                              | 0  | 1 | 2   | 3 | 4   | 5 | 6   | 7                                  | 8  | 9 | 10  | 11                                 | 12                                 |                                    |                                    |
|------------------------------|--|---|---|---|---|---|---|------------------------------------|--|---|---|------------------------------------|------------------------------------|------------------------------------|------------------------------------|
| <b>Pondělí</b><br>30.10.2006 |  |   | 157300 -<br>Operační<br>výzkum A<br>(Cvičení)<br>A002 9 <sup>00</sup> -10 <sup>30</sup> |   | 157328 - Databázové systémy<br>(Cvičení)<br>Doc. Kalužová<br>EM4F1<br>A002 10 <sup>45</sup> -11 <sup>30</sup> |   | 12 <sup>30</sup> -13 <sup>15</sup>  | 13 <sup>15</sup> -14 <sup>00</sup> |  |   | 14 <sup>15</sup> -15 <sup>00</sup>  | 15 <sup>00</sup> -15 <sup>45</sup> | 16 <sup>00</sup> -16 <sup>45</sup> | 16 <sup>45</sup> -17 <sup>30</sup> | 17 <sup>45</sup> -18 <sup>30</sup> |
| <b>Úterý</b><br>31.10.2006   | 157300 -<br>Operační<br>výzkum A<br>(Cvičení)<br>A002 7 <sup>15</sup> -8 <sup>45</sup> |   |   |   | 157300 -<br>Operační výzkum<br>A (Cvičení)<br>A002 10 <sup>45</sup> -12 <sup>15</sup>                         |   | 157300 -<br>Operační výzkum<br>A (Cvičení)<br>A002 12 <sup>30</sup> -14 <sup>00</sup> |                                    | 157300 -<br>Operační výzkum<br>A (Cvičení)<br>A002 14 <sup>15</sup> -15 <sup>45</sup>  |   |   |                                    |                                    |                                    |                                    |
| <b>Středa</b><br>1.11.2006   | 157300 -<br>Operační<br>výzkum A<br>(Cvičení)<br>A002 7 <sup>15</sup> -8 <sup>45</sup> |   | Blokování místnosti<br>A002 9 <sup>00</sup> -12 <sup>15</sup>                           |   |   |   | 155300 -<br>Informatika A<br>(Cvičení)<br>A002 10 <sup>45</sup> -11 <sup>30</sup>     |                                    | Náhrada cvičení<br>151334 -<br>Statistika B<br>A002 12 <sup>30</sup> -14 <sup>00</sup> |   | 157300 -<br>Operační výzkum<br>A (Cvičení)<br>A002 14 <sup>15</sup> -15 <sup>45</sup> |                                    |                                    |                                    |                                    |

Obr. 3: Prezentační vrstva neveřejné část portálu

Implementace kvalitní prezentační vrstvy na straně tenkého klienta je pak jedním z klíčových problémů realizace neveřejné části portálu. Jak již bylo řečeno, stanoveným cílem bylo dosažení maximálně kvalitního uživatelského interface (včetně grafických komponent typu *dialog box*, *pop-up menu*, *tree* apod., jež nejsou standardně v rámci technologie XHTML dostupné) na straně uživatele bez možnosti využití instalovaných plug-in komponent WWW

prohlížeče (zejména technologie Java Applets, Flash apod.). Autory projektu byla tedy navržena a implementována poměrně rozsáhlá knihovna objektových komponent v programovacích jazycích Java na straně aplikačního serveru a JavaScript na straně tenkého klienta k realizaci uživatelského rozhraní na straně tenkého klienta s následujícími vlastnostmi:

- Filozofie tvorby grafického uživatelského rozhraní strany tenkého klienta je obdobná filozofii tvorby grafického uživatelského rozhraní tlustého klienta (známé např. z technologií Java AWT a Java Swing). Za tím účelem byla implementována knihovna tříd Java Bean, pomocí nichž je prováděna tvorba komponent *view* strany portálového serveru.
- Při generování grafického uživatelského rozhraní je kladen maximální důraz na rychlost vytváření konkrétního fragmentu WWW stránky, který typicky obsahuje značné množství grafických uživatelských komponent. Z tohoto důvodu bylo upuštěno od nasazení technologie Java Server Pages v roli komponent *view* návrhového vzoru, u nichž např. probíhá při prvním odkazu ze strany uživatele portálu kompilace zdrojového kódu JSP stránky za účelem vytvoření instance Java servletu, což značně zpomaluje generování fragmentu výsledné WWW stránky v prostředí portálových technologií s využitím Java portletů.
- Z obdobných důvodů rovněž není využito technologie Java Server Faces (JSF – viz. [13]), i když způsob generování prezentační vrstvy portálu respektuje mnohé ze zde známých a zavedených principů (např. práce s událostmi, využití komponent Java Beans apod.).

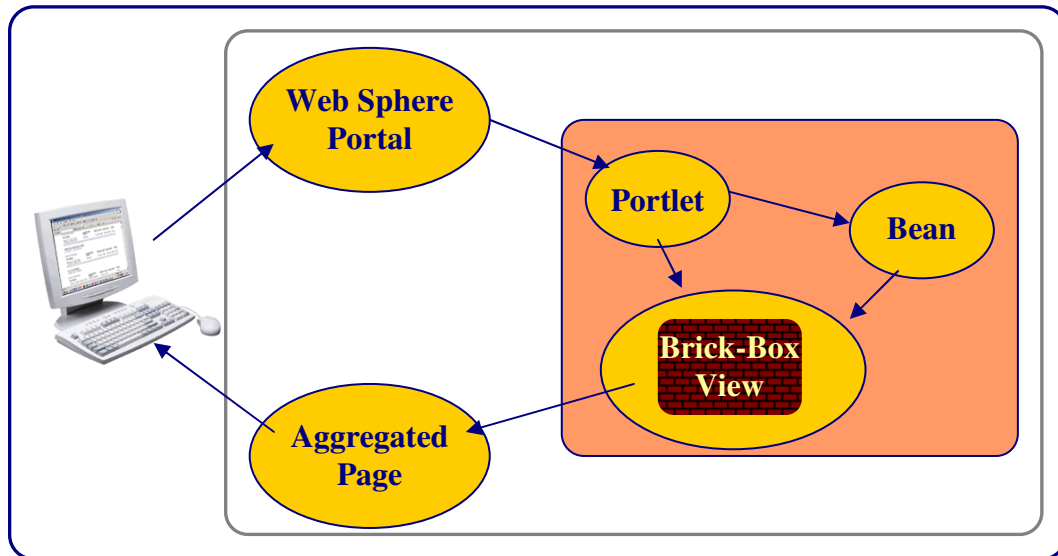
Specializovaný návrhový vzor *Brick-Box Model-View-Controller* a jeho funkcionality vyplynuly především z požadavku na rychlost generování fragmentu WWW stránky portálového prostředí. V důsledku komplikovanosti a rozsahu XHTML kódu výsledného fragmentu se ukázalo velice účelným ustavení programové cache paměti, která uchovává již vygenerované části jednotlivých fragmentů stránek s *konstantním obsahem* reprezentujících jednotlivé grafické komponenty (např. tlačítka, panely, menu apod.) příp. jejich části (záhlaví tabulek, tabulátory apod.). Tyto předem vygenerované elementy výsledného fragmentu WWW stránky s konstantním obsahem v rámci obecně dynamicky generované WWW stránky si lze představit jako cihly v budované zdi, které budou při její stavbě slepeny jistým pojivem. Tímto pojivem jsou *dynamicky generované části* fragmentu WWW stránky (např. obsah tabulek, rolovacích menu apod.), jejichž obsah produkován komponentou *model* návrhového vzoru musí být přesně uzpůsoben již vygenerovaným konstantním částem fragmentu stránky (viz. obr. 4).

Praktická realizace fakultního portálového řešení s využitím návrhového vzoru *Brick-Box Model-View-Controller* prokázala jeho funkčnost při významné podpoře rychlosti generování fragmentů WWW stránek obsahující velice komplexní uživatelské rozhraní.

#### 4. ZÁVĚR

Závěrem příspěvku lze tedy konstatovat vysokou aktuálnost více než 20 let starého návrhového vzoru *Model-View-Controller* při návrhu a implementaci portálových řešení s vícevrstvou architekturou. Zásadním cílem dalších generací návrhových vzorů pro tvorbu prezentační vrstvy portálových řešení by pak mělo být sjednocení filozofií a technologií při

vytváření uživatelského rozhraní pro prostředí tenkého a tlustého klienta, tj. zejména vybudování univerzální knihovny tříd, při jejichž použití by bylo možné uživatelské rozhraní grafické aplikace určené pro stranu tlustého klienta bez větších a podstatnějších změn přenést i do prostředí portálových aplikací a tenkého klienta.



Obr. 4: Návrhový vzor Brick-Box Model-View-Controller

## LITERATURA

1. ABDELNUR A., HEPPER S.: Java Portlet Specification – Version 1.0 (JSR 168), Sun Microsystems, 2003
2. LINWOOD J., MINTER D: Building Portals with the Java Portlet API, Apress, 2004, ISBN 1-59059-284-0
3. <http://java.sun.com/javase/>
4. <http://java.sun.com/javaee/>
5. BURBECK S.: Applications Programming in Smalltalk-80: How To Use Model-View-Controller, 1992, ParcPlace, <http://st-www.cs.uiuc.edu/users/smarch/st-docs/mvc.html>
6. HALL M.: More Servlets and Java Server Pages, Sun Microsystems Press, 2002, ISBN 0-13-067614-4
7. <http://www.w3.org/DOM/>
8. <http://www.w3.org/Style/CSS/>
9. <http://www-306.ibm.com/software/webservers/appserv/was/>
10. <http://java.sun.com/javaee/overview/compatibility.jsp>
11. <http://www-03.ibm.com/systems/i/>
12. <http://www-306.ibm.com/software/data/cm/cmgr/mp/edition-enterprise.html>
13. <http://java.sun.com/javaee/jaserverfaces/>