

PORT@L

implementace portálového řešení IBM WebSphere Portal v českém univerzitním prostředí

*technická a závěrečná zpráva projektu
FR CESNET č. 043R1/2003*

Jan Valdman, Maxmilán Otta, Jiří Šimonek, Kateřina
Rašková a kol.

Plzeň, únor 2005

©2005 Západočeská univerzita v Plzni

Názvy produktů a společností uvedené v této zprávě mohou být ochrannými známkami nebo registrovanými ochrannými známkami svých vlastníků.

V textu jsou použity obrazové materiály pocházející z prezentací a dokumentů společnosti IBM.

Abstract

In this paper we describe our efforts to implement IBM WebSphere Portal version 5 in academic environment of three Czech universities. The report covers the following areas: product instalation, configuration for local IT infrastructure, basic portal administration, content management and custom portlet development.

The second part of the paper deals moreorless with grant issues — project milestones, accounting, achieved goals and some more. Additional valuable information is included in rich attachments.

Keywords

IBM WebSphere Portal, IBM Application Server, WAS, WPS, WPF, portal, portlet development

Obsah

1 Úvod	1
1.1. Co bylo na začátku	1
1.2. Portály a vysoké školy v roce 2003	2
1.3. Výběr produktu, pokusy s WSP4.2	3
I Budování portálu	5
2 Instalace	7
2.1. Volba OS	8
2.2. WebSphere Portal 5.0 a 5.02	8
2.3. Instalace na UPOL	10
2.4. Instalace WPS 5.0.2 na OSU	12
2.4.1. Instalace na RedHat 8.0	12
2.4.2. Instalace na SuSE Linux – SLES8	14
2.4.3. Nastavení IBM HTTP serveru (IHS)	14
2.5. Instalace na ZČU	14
2.6. Fixpacky a patche	15
2.7. Poznámky	17
3 Administrace WAS	19
3.1. Administrační konzole WAS	19
3.2. Systém konfiguračních souborů	22
3.2.1. Hierarchie adresářů	22
3.2.2. Konfigurační soubory	23
3.3. Uživatelské registry portálu	23
3.3.1. Vztah registrů WAS a WPS	25
3.3.2. WebSphere Member Management (WMM)	25

3.3.3. Jak probíhá autorizace	26
3.4. Instalace JDBC ovladačů	26
3.5. Instalace JCO	27
3.5.1. Postup instalace	27
3.5.2. Nastavení v SAP	27
4 Napojení na infrastrukturu	29
4.1. Migrace do databázového systému Oracle	30
4.1.1. Příprava databáze – ZČU	30
4.1.2. Migrace – ZČU	32
4.1.3. Nastavení a Migrace do Oracle – OSU	36
4.2. Nastavení LDAP	38
4.2.1. OpenLDAP na ZČU	38
4.2.2. Poznámky	39
4.2.3. Novell eDirectory na OSU	39
4.3. Generování SSL klíčů a IKEYMAN	39
4.4. Zapnutí Global Security	40
4.5. Zapnutí HTTPS browser–IHS (ZČU)	40
4.5.1. Nastavení IHS	40
4.5.2. WAS web admin	41
4.5.3. httpd.conf	41
4.5.4. Přihlašování do WPS	41
4.5.5. Finalizace	41
4.6. Zapnutí HTTPS browser–IHS (OSU)	42
4.6.1. Nastavení IHS pomocí konfiguračního souboru	42
4.6.2. Nastavení IHS pomocí konzole (doporučeno)	43
4.7. Zapnutí SSL WAS–LDAP a WAS–IHS (ZČU)	43
4.7.1. Registrace CA chain	44
4.7.2. Úprava konfiguračních souborů	44
4.8. Zapnutí SSL WAS–LDAP a WAS–IHS (OSU)	44
4.8.1. Úprava Web server plug in souboru	45
4.8.2. Nastavení portálu pro LDAP SSL	46
4.8.3. Nastavení portálu pro SSL provoz	47
4.9. Některé problémy	47
4.9.1. Novell a WPS5.0	47
4.9.1.1. Problém uniqueID	48
4.9.1.2. Problém vyhledávacích filtrů	48
4.9.1.3. Problém nastavení WMM	48

4.9.1.4. Výkonový problém s OpenLDAP	48
4.9.2. Problém komunikace SSL	49
4.9.2.1. Zapnutí SSL pro LDAP	49
4.9.2.2. Portal Document Manager	49
4.9.3. Problémy s Oracle databází	49
4.9.3.1. Nastavení locale	49
4.9.3.2. Uživatel db2admin	49
5 Správa WebSphere Portal	51
5.1. Administrační rozhraní WPS	52
5.2. Konfigurační soubory WPS	52
5.3. Autorizace	53
5.3.1. Prostředky chráněné WPS	53
5.3.2. Role Based Access Control	54
5.3.3. Dědění oprávnění	57
5.3.4. Jak nastavovat přístupová práva	57
5.3.5. Nastavení práv – příklad ZČU	58
5.3.6. Jak delegovat řízení přístupových práv	58
5.3.7. Doporučení	58
5.4. Branding portálu	59
5.4.1. Jak na to	59
5.4.2. Tipy pro zrychlení práce	60
5.5. Vypnutí samoobslužných mechanismů	61
5.6. Mapování URL	62
5.7. Konfigurační rozhraní XML	62
6 Správa obsahu a další komponenty	65
6.1. Možnosti správy obsahu	66
6.2. Publikujeme statický obsah	66
6.2.1. Fileserver portlet	66
6.2.2. Proužková reklama	66
6.2.3. iFrame portlet	67
6.2.4. StaticText portlet	67
6.2.5. HTMLcontent portlet	68
6.2.6. CSV portlet	68
6.3. Publikujeme dynamický obsah	69
6.3.1. JSP Server portlet	70
6.3.2. RSS portlet	70
6.4. Správa dokumentů	70

6.4.1. Portal Document Manager (PDM)	71
6.4.2. Prohlížeče souborů	72
6.4.3. Cleverbee FileExplorer portlet	73
6.5. Personalization	74
6.6. Customization	75
6.7. Single-Sign-On	75
6.8. Credential Vault	76
6.9. Hledání	77
7 Vývoj a integrace aplikací	79
7.1. J2EE standardy a API	80
7.2. Zdroje informací aneb J2EE není legrace	81
7.3. Metodika integrace aplikací do portálu	81
7.3.1. Možnosti přístupu k IS v portálu	82
7.3.2. Integrace na databázové vrstvě	82
7.3.2.1. Portlet Sestavy dotazů	83
7.3.2.2. JDBC Portlet Builder	84
7.3.3. Integrace přes nativní aplikační rozhraní	84
7.3.4. Integrace na prezentační vrstvě	85
7.4. Integrace SAP do WebSphere Portal	87
7.4.1. SAP HTML GUI Integration	88
7.4.2. SAP Business Warehouse Integration portlet	88
7.4.3. SAP Enterprise Integration Portlets	88
7.4.4. IBM SAP MSS My Staff Portlets	88
7.5. WebSphere Portal Application Integration	88
7.5.1. JDBC Portlet Builder	89
7.5.1.1. Konfigurace JDBC portlet builderu	89
7.5.1.2. Vytvoření portletu pomocí JDBC portlet builderu	90
7.5.2. SAP Portlet Builder	92
7.5.2.1. Konfigurace SAP Portlet Builderu	92
7.5.2.2. Konfigurace portletu pomocí SAP Portlet Builderu	92
7.5.2.3. Konfigurace SAP Business Object Editoru	93
7.5.2.4. Vytvoření nového objektu v SAP Business Object Editoru	93
7.5.3. Instalace a konfigurace portletů	95
7.6. Vývoj vícevrstvých aplikací pro portál	95
7.6.1. Úvod do EJB	96
7.6.2. Vývoj EJB pomocí IBM WebSphere Studio	97

7.6.3. Metodiky použité při vývoji EJB	97
7.6.4. Přehled vytvořených EJB	99
7.6.5. Řešení autentizace ve vícevrstvé architektuře	99
7.6.6. Prezentační vrstva – úvod do portletů	100
7.6.7. Přehled vytvořených portletů	100
7.7. IBM WebSphere Studio	100
7.7.1. Vývoj portletů ve WebSphere Studiu	101
7.7.2. Instalace WSAD	101
7.8. IBM Rational SDP	102
7.9. Nastavení JDBC Datasource	103

II Řešení projektu PORT@L 105

8 Řešení projektu 107

8.1. Postup řešení	107
8.1.1. Pořízení a instalace WPS	108
8.1.2. Vývoj aplikací	108
8.1.3. Získávání know-how	109
8.1.4. Pilotní projekty	109
8.2. Adresář řešitelů	110
8.3. Změny v projektu	110
8.4. Čerpání finančních prostředků	110
8.4.1. Investiční prostředky	110
8.4.2. Neinvestiční prostředky	111
8.5. Spolupráce s IBM	111
8.6. Hledání pomoci u partnerů IBM	112
8.7. Diskuse splnění cílů projektu	113
8.8. Shrnutí	115
8.8.1. Udělali jsme	115
8.8.2. Nestihli jsme	115
8.8.3. Podařilo se	116
8.8.4. Nepovedlo se	116
8.8.5. „Lessons learned“	116
8.9. Další využití výsledků projektu	116

9 Výstupy projektu 119

9.1. Portál ZČU	120
9.1.1. Obecné informace	120

9.1.2. Struktura portálu	121
9.1.3. Zavedení decentralizovaných principů	122
9.1.4. Personalizace	122
9.1.5. Co se nepodařilo nebo jsme nestihli	122
9.2. Portál OSU	123
9.2.1. Struktura portálu	124
9.2.2. Personalizace	125
9.2.3. Přístupové role a profily	125
9.2.4. Co se nepodařilo nebo jsme nestihli	125
9.3. Portál UPOL (návrh)	125
9.3.1. Pravidla pro nasazení WSP v prostředí UPOL	126
9.3.2. Rozvrstvení portálu	127
9.3.3. Přihlašování uživatelů a přiřazení rolí	127
9.3.4. Integrace stávajících aplikací a SSO přístup	127
9.3.5. Další rozvoj portálu	128
9.4. Prezentace na konferencích	128
9.5. Publikované články	129
9.6. Vytvořené portlety	129
10 Katalog vytvořených portletů	131
10.1. Univerzální portlety	132
10.1.1. Přihlašovací portlet	132
10.1.2. Sitemap	132
10.1.3. WhoAmI	133
10.1.4. Statický text	133
10.1.5. Diskusní portlet	134
10.2. IS/STAG portlety	134
10.2.1. Informace o studentovi	135
10.2.2. Plnění studijního programu studenta	135
10.2.3. Rozvrh studenta	136
10.2.4. Znamky studenta	136
10.2.5. Zkoušky studenta	136
10.2.6. Informace o učiteli	137
10.2.7. Rozvrh učitele	137
10.2.8. Diplomové práce učitele	137
10.2.9. Informace o předmětu	138
10.3. MAGION portlety	138
10.3.1. Personální informace	139

10.3.2	Cestovní náhrady	139
10.3.3	Kursovni listek	139
10.3.4	Stravenky	140
10.3.5	Vývěska Magion	140
10.4	Aplikace ZČU	140
10.4.1	Telefonni seznam	141
10.4.2	Telefonni seznam pracoviště	141
10.5	Aplikace OSU	141
10.5.1	Statistiky OSU	141
10.5.2	Akreditace	142
10.5.3	Databáze závěrečných prací	142
10.5.4	Státnicové otázky	143
10.5.5	Informace o studentovi	143
10.5.6	Ověření platnosti ISIC karty	144
10.5.7	Jídelniček	144
10.5.8	KNOS	144
10.5.9	Peoplefinder	145
10.5.10	Restování síťových služeb	145
10.5.11	Hlášení poruch	146
10.5.12	Rezervace počítačových učeben	147
10.5.13	Webové statistiky	147
10.6	Aplikace UPOL	147
10.6.1	SAP Zaměstnanci	148
10.6.2	SAP Nepřítomnosti	148
10.6.3	SAP Kvalifikace	148
11	Závěr	151
11.1	Shrnutí	151
11.2	Portál – ano či ne?	152
11.3	Vyplatí se portál?	152
11.4	WebSphere Portál – ano nebo ne?	152
11.5	WebSphere Portál – stojí za ty peníze?	153
11.6	Naše doporučení	153
III	Přílohy	161
	Řešitelské týmy	163
A.1.	Západočeská univerzita v Plzni	165

A.2. Univerzita Palackého v Olomouci	166
A.3. Ostravská univerzita v Ostravě	166
JSP a IBM Portlet API	169
B.1. Java Servlety	171
B.2. JavaServer pages	171
B.2.1. JSP kontejner	171
B.2.2. Postup zpracování JSP	171
B.2.3. JSP direktivy	172
B.2.4. JSP skriptovací značky	172
B.3. Standard MVC	172
B.4. Portlet MVC	173
B.5. Portlet API	173
B.5.1. Pojmy používané v souvislosti s portlety	173
B.5.2. Sestavení stránky	174
B.5.3. Meziportletová komunikace	174
B.5.4. Obsluha událostí	174
B.5.5. Bezpečnost	174
B.5.6. Portletová aplikace	175
B.5.7. Srovnání servletu a portletu	175
B.6. Základní objekty portletu	175
B.6.1. Portlet	175
B.6.2. PortletAdapter	175
B.6.3. PortletRequest	176
B.6.4. PortletResponse	177
B.6.5. PortletSession	178
B.6.6. Client	179
B.6.7. PortletConfig	179
B.6.8. PortletContext	179
B.6.9. PortletSettings	180
B.6.10PortletApplicationSettings	180
B.6.11PortletData	181
B.6.12PortletLog	181
B.6.13PortletException	181
B.6.14UnavailableException	181
B.6.15PortletWindow	182
B.6.16User	182
Tutorial: Vývoj portletů pro WebSphere Portal	183

C.1. Hello world projekt	185
C.2. První spuštění	185
C.2.1. Preview portlet	185
C.2.2. Run on server	185
C.3. Uchování obsahu proměnných	186
C.3.1. PortletSession	186
C.3.2. PortletSettings	187
C.3.3. PortletData	188
C.4. Konfigurace aplikace	189
C.4.1. portlet.xml	189
C.5. Příprava pro práci s databází	190
C.5.1. Vytvoření spojení	190
C.5.2. Tvorba databáze	191
C.5.3. Java Beans pro komunikaci s databází	191
C.6. Lokalizace	191
C.6.1. NLS (National Language Support)	191
C.6.2. Překlad celého souboru	193
C.7. Jedinečné názvy proměnných a funkcí	193
C.7.1. JavaScript	193
C.8. Konkrétní úpravy projektu	194
C.8.1. HelloWorldPortlet.java	194
3.8.1.1. Import knihoven	194
3.8.1.2. Globální proměnné	194
3.8.1.3. Metoda doView()	196
3.8.1.4. Metoda doConfigure()	196
3.8.1.5. Metoda actionPerformed()	197
C.8.2. SQLInsert.java	199
C.8.3. SQLSelect.java	199
C.8.4. HelloWorldPortletViewBean.java	200
C.8.5. HelloWorldPortletConfigBean.java	200
C.8.6. HelloWorldPortletSessionBean.java	201
C.8.7. HelloWorldPortletView.jsp	201
C.8.8. HelloWorldPortletConfig.jsp	202
C.9. Vytvoření WAR souboru a instalace	204
C.10. Možné problémy	205
C.11. Rozdíly databází Cloudscape a Oracle	205

D.1. Účel	209
D.2. Obsah	209
D.3. Bibliografické informace	209
Zprávy z konferencí	211
E.1. IBM WebSphere Technical Conference	213
E.2. IBM WebSphere Portal and Lotus Technical Conference	213
E.3. Pan European Portals Conference – PEPC2004	213
E.4. Directory and Single Sign-On in European Universities	214
E.5. Univerzitní informační systémy – konference UNINFOS 04	214
Různé	215
F.1. Hlášení chyby do laboratoří IBM	217
F.2. Nabídka na služby Groupsoft	219
F.3. Nabídka Cleverbee – portlet FileExplorer	222
F.4. Další zdroje informací	223

Seznam obrázků

2.1 Schéma WebSphere Portal	9
2.2 Základní network deployment	9
3.1 Uživatelské rozhraní administrační konzole	20
3.2 Funkce administrační konzole	21
3.3 Runtime konfigurační prostředí uzlu (node)	22
3.4 Struktura member services	24
5.1 Webové administrační rozhraní portálu	52
5.2 Hierarchie prostředků chráněných portálem	54
5.3 Definice role (vlevo) a její přiřazení (vpravo)	55
5.4 Vazby mezi rolemi	56
5.5 Základní grafické schéma portálu ZČU	59
5.6 Základní grafické schéma portálu OSU	59
5.7 Základní skiny portletů používané v portálu ZČU	60
5.8 Princip mapování URL	62
5.9 Natavení mapování URL	63
6.1 Rozhraní portletu Proužková reklama (edit mode)	67
6.2 Rozhraní portletu Static Text (edit mode)	68
6.3 Rozhraní portletu HTML Content (view mode)	68
6.4 Rozhraní portletu HTML Content (configure mode)	69
6.5 Rozhraní portletu HTML Content (edit mode)	69
6.6 Rozhraní portletu Prohlížeč CSV (edit mode)	70
6.7 IBM JSP server – konfigurace portlet.xml	70
6.8 IBM RSS server – konfigurace portlet.xml	71
6.9 Rozhraní portletu RSS (view mode)	71
6.10 Portal Document Manager (view mode)	71

6.11 Portal Document Manager (edit mode)	72
6.12 Prohlížeč souborů PDF (view mode)	72
6.13 Prohlížeč souborů PDF (view mode)	73
6.14 Prohlížeč souborů PDF (edit mode)	73
6.15 Rozhraní portletu File Explorer (view mode)	73
6.16 Rozhraní portletu File Explorer (configure mode)	74
6.17 Rozhraní portletu File Explorer (edit mode)	74
6.18 Frontend a backend domény SSO	76
6.19 Rozhraní portletu Hledání dokumentů (view mode)	77
6.20 Správa kolekcí dokumentů (view mode)	77
7.1 Vrstvy J2EE aplikace	80
7.2 SQL portlet (view mode)	83
7.3 JDBC Portlet Builder	84
7.4 iFrame portlet (view mode)	85
7.5 iFrame portlet (edit mode)	85
7.6 Editor webových výstřižků (Web Clipping)	86
7.7 Vytváření klipu	86
7.8 Ukázka webového klipu	86
7.9 Propojení WPS a SAP	87
7.10 Možnosti napojení se na SAP	87
7.11 JDBC Portlet Builder – editace portletu	89
7.12 JDBC Portlet Builder – editace business objektu	89
7.13 SAP Portlet Builder	92
7.14 SAP Business Object Editor	94
7.15 Vícevrstvá architektura v prostředí J2EE	95
7.16 Typy J2EE kontejnerů a vazby mezi nimi	96
7.17 Rodina WebSphere Studio	101
7.18 Portletová perspektiva WSAD	102
7.19 J2EE perspektiva v Rational Application Developer	103
9.1 Ukázka portálové stránky	120
9.2 Ukázka portálové stránky	123
9.3 Ukázka portálové stránky	126
10.1 Přihlašovací portlet	133
10.2 Sitemap portlet	133
10.3 Who Am I portlet	133
10.4 Statický text	134

10.5	Diskusní portlet	134
10.6	Informace o studentovi	135
10.7	Přehled plnění předmětů	135
10.8	Rozvrh studenta	136
10.9	Známky studenta	136
10.10	Žkousky studenta	137
10.11	Informace o učiteli	137
10.12	Rozvrh učitele	137
10.13	Diplomové práce učitele	138
10.14	Informace o předmětu (normální zobrazení)	138
10.15	Personální informace	139
10.16	Cestovní náhrady	139
10.17	Kursovni lístek	140
10.18	Stravenky	140
10.19	Vývěska MAGION	140
10.20	Telefonní seznam	141
10.21	Telefonní seznam pracoviště	141
10.22	Statistiky OSU	142
10.23	Akreditace	142
10.24	Databáze závěrečných prací	143
10.25	Státnicové otázky	143
10.26	Informace o studentovi	144
10.27.	144
10.28	Řídelníček	145
10.29	Knihovni objednávkový systém	145
10.30	Hledání osob	146
10.31	Testování síťových služeb	146
10.32	Hlášení poruch	146
10.33	Rezervace počítačových učeben	147
10.34	Webové statistiky	147
10.35	SAP zaměstnanci na středisku	148
10.36	SAP nepřítomnosti	149
10.37	SAP kvalifikace	149
B.1	Objekty Portlet API	176
B.2	Životní cyklus portletu	177
C.1	Nastavení parametrů pro <i>portlet preview</i>	186
C.2	Nový server a jeho nastavení	187

C.3	Parametrizace portletu	188
C.4	Přidání proměnných a jejich hodnot do souboru portlet.xml .	189
C.5	Konfigurace připojení k databázi.	190
C.6	Zakomponování knihovny jstl.jar do projektu.	192
C.7	Blokové schéma HelloWorld portletu.	195
C.8	<i>View</i> režim HelloWorld portletu	205
C.9	<i>Configure</i> režim HelloWorld portletu	206

Úvod

První kapitola podává základní informace o pozadí projektu PORT@L, počátečních experimentech s WebSphere Portal ve verzi 4.2 a o celkové portálové situaci v ČR všeobecně.

Obsah

1.1. Co bylo na začátku	1
1.2. Portály a vysoké školy v roce 2003	2
1.3. Výběr produktu, pokusy s WSP4.2	3

Tento dokument plní funkci závěrečné zprávy projektu¹ FR CESNET číslo 043R1/2003 a zároveň naplňuje závazek řešite dodat rozsáhlou technickou zprávu².

Dokument vznikl jako společné dílo řešitele a spoluřešitelů projektu, které zaznamenává implementaci portálového řešení IBM na třech českých univerzitách. Snažili jsme se utřídit poznámky tří řešitelských týmů do jednoho logického celku, který čtenáře postupně provede jednotlivými fázemi implementace portálu.

Jednotlivé body požadované struktury závěrečné zprávy jsou naplněny takto:

- postup řešení, způsob řešení – kapitola 8
- dosažené cíle – kapitola 9
- zdůvodnění případných změn v projektu – kapitola 8
- konkrétní výstupy, další využitelnost – kapitoly 9 a 10
- přínosy projektu, vlastní hodnocení – kapitoly 9 a 11

1.1. Co bylo na začátku

Západočeská univerzita v Plzni se myšlenkou portálu zabývá přibližně od roku 2001, první konkrétní kroky byly podniknuty na podzim 2002. V té době jsme detailně zkoumali Oracle 9iAS portál, ale nakonec jsme si vybrali portálové řešení IBM.

¹pro oblast I projektů FR CESNET

²viz smlouva o spolupráci při řešení výzkumného projektu

Na jaře 2003 jsme podali grantovou přihlášku na projekt PORT@L, jako společný projekt Západočeské univerzity v Plzni, Ostravské univerzity v Ostravě a Univerzity Palackého v Olomouci. Toto spojení se překvapivě ukázalo jako nevýhodné (zřejmě kvůli celkové výši sdružené investice), grant byl přijat pouze podmíněčně. V září 2003 byla vypracována požadovaná studie portálových řešení na českém trhu (viz [ZCU-1]), na jejímž základě byl grant nakonec schválen. K podpisu smlouvy mezi Fondem rozvoje sdružení CESNET a ZČU došlo dne 27.2.2004, čili téměř po roce.

Vzhledem k tomu, že ZČU a OSU byly rozhodnuty realizovat portál i v případě nepřijetí grantu, probíhaly práce fakticky už od roku 2003.

1.2. Portály a vysoké školy v roce 2003

Mít vlastní portál chce v posledních letech každá větší instituce. Filosofický problém je v tom, že firmy neprodávají portály, ale portálové produkty, které se obvykle zaměřují pouze na firemní prostředí, kde má portál řešit konkrétní obchodní problémy (business case), tomu má být podřízen výběr produktu; je chybou zvolit produkt a poté se snažit nahodilými akcemi využít jeho možnosti.

Univerzitní prostředí se od firemního v mnohém liší. Přestože jsme si vědomi výše uvedené skutečnosti, spočívá část naší práce právě v experimentu zjistit, nakolik (a jak úspěšně) lze v univerzitním prostředí nasadit podnikové řešení, v našem konkrétním případě IBM WebSphere Portal.

Ve firmě velikosti univerzity (stovky až tisíce kancelářských zaměstnanců), se portál zavádí obvykle jen za jedním účelem nebo v jednom oddělení. V tomto ohledu je naše řešení také jiné, neboť naše portály jsou koncipovány celouniverzitně. Při nutné heterogenitě oddělení uvnitř univerzity i vzájemné odlišnosti univerzit potom logicky vzniká situace, že portál nasazujeme současně v několika různých oblastech a snažíme se jím paradoxně řešit všechny problémy zároveň. Současně analyzujeme průběžné výsledky a hledáme další možnosti jeho využití.

Hlavní důvody pro nasazení portálu jsou podrobně rozebrány v grantové přihlášce, zde je jen shrneme:

- potřeba personalizovaného informačního prostředí (podpora studia a spolupráce)
- konsolidace webového prostředí škol (místo desítek webů jeden portál)
- přechod na třívrstvou softwarovou architekturu (neudržitelnost velkých systémů při značné diverzitě použitých technologií - PL/SQL, PHP, Perl, JSP)
- robustní a perspektivní aplikační rámcová aplikační architektura (J2EE)
- integrační platforma pro aplikace
- zpřehlednění a zjednodušení autorizace a přístupu k informacím na intranetu
- samoobslužné webové publikování

1.3. Výběr produktu, pokusy s WSP4.2

O výběru portálového produktu bylo rozhodnuto již v době podání grantové přihlášky; na základě průzkumu českého trhu, opensource projektů i osobních zkušeností s Oracle portálem, jsme vybrali *IBM WebSphere Portal for Multiplatforms* v licenčním modelu *Express+*.

Verze *Express+* představuje licenční schéma omezené na jednoserverovou architekturu (single box do čtyř CPU) – prakticky to znamená pouze nemožnost tvořit clustery. Součástí balíku je kromě aplikačního serveru *WebSphere Application Server* a portálového serveru *WebSphere Portal* též *IBM HTTP Server*, *IBM Directory Server*, databáze *DB2*, vývojový nástroj *WebSphere Studio Application Developer* a kolaborativní nástroje *Lotus: QuickPlace* a *Sametime*.

Naše první setkání s *WebSphere Portal* bylo ve verzi 4.2, kterou jsme měli v létě roku 2003 zapůjčenou od IBM. Portál nám vyhovoval svými funkcemi, ale provozovali jsme jej pouze v laboratorním prostředí nad databází *DB2* a *IBM Directory* serverem. Pokoušeli jsme se sice o napojení na *Oracle* a *OpenLDAP* (*Novell eDirectory*), ale to se nám nikdy nepovedlo. Ve verzi 4.2 muselo napojení na infrastrukturu proběhnout již během instalace, které vždy havarovalo. Postupným řešením četných technických problémů v ještě nevyzrálém produktu jsme se propracovali vždy do další fáze instalace, ale bohužel ne k úspěšnému konci.

V okamžiku přijetí grantu byl *WebSphere Portal* na trhu již ve výrazně inovované verzi 5.0, kterou jsme také použili k řešení projektu.

PORT@L – technická zpráva projektu FR CESNET 043R1/2003

Část I

Budování portálu

Instalace

Instalace WebSphere Portal a WebSphere Application Server je netriviální úloha. Tato kapitola popisuje technické záležitosti rozběhnutí portálu; jestliže vás tato oblast nezajímá, můžete ji klidně přeskočit...

Obsah

2.1. Volba OS	8
2.2. WebSphere Portal 5.0 a 5.02	8
2.3. Instalace na UPOL	10
2.4. Instalace WPS 5.0.2 na OSU	12
2.4.1. Instalace na RedHat 8.0	12
2.4.2. Instalace na SuSE Linux – SLES8	14
2.4.3. Nastavení IBM HTTP serveru (IHS)	14
2.5. Instalace na ZČU	14
2.6. Fixpacky a patche	15
2.7. Poznámky	17

Tato kapitola popisuje pouze samotnou instalaci produktu IBM WebSphere Portal – Express 5.0, která se skládá z následujících kroků:

1. příprava OS, instalace požadovaných balíčků
2. spuštění instalačního programu, který provede
 - instalace WAS, WPS, IHS, CloudScape
 - volitelná instalace IDS, DB2
 - automatická instalace opravných balíčků WAS a WPS
3. případná úprava konfigurace WAS, WPS, IHS (napojení na infrastrukturu viz kapitola 4)
4. ruční instalace nejnovějších opravných balíčků

Od verze WebSphere Portal 5 probíhá instalace tím způsobem, že se nejprve nainstaluje WebSphere Application Server a WebSphere Portal v základní variantě, tj. databáze i adresář uživatelů je napojen na přibalenou databázi CloudScape; teprve potom se databáze a uživatelé odmigrují do enterprise databáze (např. Oracle) a jiného registru uživatelů (obvykle LDAP).

Instalace celého prostředí se může ubírat více cestami, proto zde přikládáme dva odlišné scénáře instalace (jeden pro ZČU a jeden pro OSU).

2.1. Volba OS

WebSphere Portal ve verzi Express je certifikován pro Windows 2000, Suse Linux Enterprise Server a RedHat Advanced Server. Samotný WebSphere Portal je Java aplikace a proto by se měla na obou platformách chovat stejně, nicméně v praxi se ukázalo, že zejména instalace je na platformách Linux a Windows různě obtížná.

Volíme tedy platformu Linux jako stabilní prostředí pro dlouhodobý provoz produkčního portálu a pro vývoj portletů platformu Windows, na které jde WebSphere Portal rychleji nainstalovat, ale obtížněji dlouhodobě provozovat.

Instalační program je pro obě platformy podobný, ale na různých PC se chová nevyzpytatelně – například se bez zjevného důvodu zastavuje až na desítky minut (bez využití CPU nebo IO subsystému).

Při pokusné instalaci na SuSE Linux Enterprise Server 7.0 (SLE7) instalační program detekuje „enterprise“ verzi OS a začne (pro nás zbytečně) instalovat middlewarové komponenty IBM MQ, navíc neúspěšně. Protože jsme tyto funkce nepotřebovali, vrátili jsme se zpět k obyčejnému RedHatu 8.0, kterým lze nahradit RedHat Advanced Server.

WebSphere Portal funguje i pod Windows XP, ale jsou počítače, kam nešel přes několik pokusů včetně opakované reinstalace OS vůbec nainstalovat.¹

Pro instalaci je doporučován „čistý“ dedikovaný stroj; je potřeba vyhovět instalačním specifikacím, které si instalační program po startu ověřuje (bohužel ne všechny).

2.2. WebSphere Portal 5.0 a 5.02

WebSphere Portal je vlastně aplikace napsaná pro aplikační server WebSphere Application Server. Obchodně je balík WebSphere Portal souborem několika produktů – k samotné aplikaci je dodáván aplikační server a další části potřebné infrastruktury.

IBM dělí produktovou řadu WebSphere Portal do dvou základních větví:

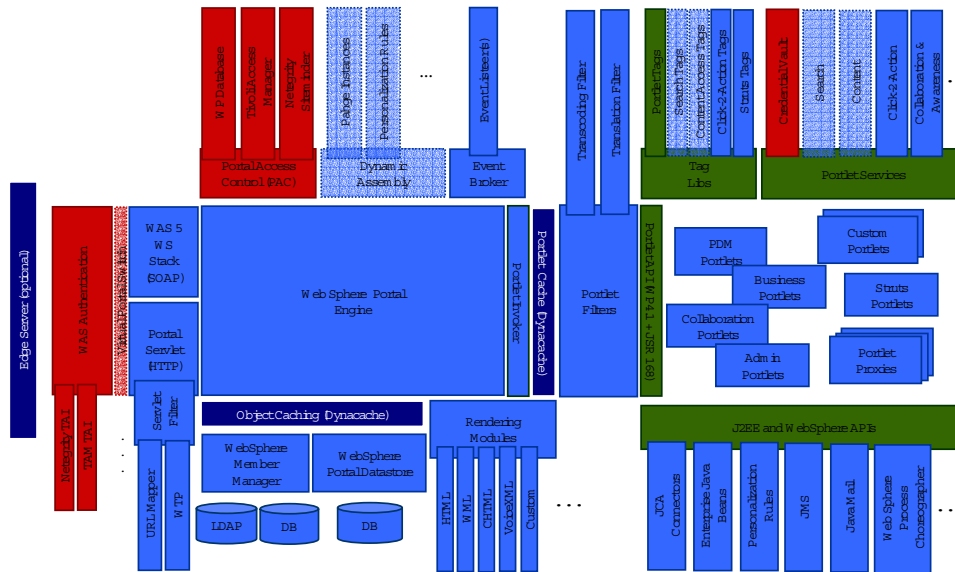
- *WebSphere Portal for Multiplatforms*, který je nabízen ve třech různých balení: *Enable*, *Extend* a *Experience* podle množství začleněných komponent
- *WebSphere Portal – Express for Multiplatforms*; tato verze se liší hlavně licencováním, funkčně odpovídá variantě *Enable*

Prvně jmenovaný produkt představuje komplexní portálová řešení pro oblasti *business-to-employee (B2E)*, *business-to-business (B2B)* a *business-to-customer (B2C)*. Druhý typ produktu je pak určen pro malé a střední podniky a jsou z něj vyčleněny některé komponenty.

Jak ukazuje schéma na obrázku 2.1, je WebSphere Portal komplexní software, jehož jádrem je *WebSphere Portal Engine*, které spolupracuje s dal-

¹V jednom případě se ukázalo, že na vině je vadný modul DIMM, se kterým počítač předtím celý rok bez problémů fungoval.

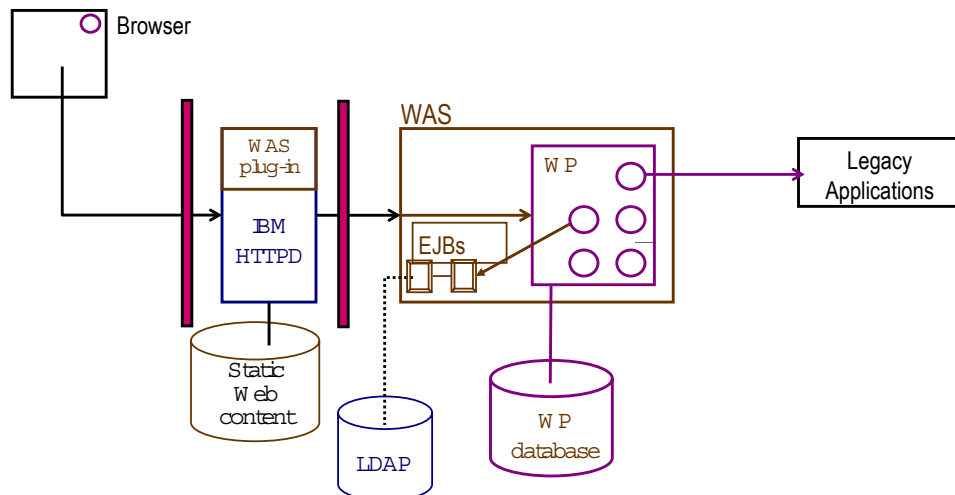
ššími důležitými komponentami jako je *HTTP Server*, autentizační a autorizační modul, portletový a *J2EE* kontejner a databáze.



Obrázek 2.1: Schéma WebSphere Portal

WebSphere Portal podporuje řadu instalačních scénářů od instalace všech komponent na jednom serveru až po jejich distribuci na různé uzly v síti včetně spojování do klusterů.

Pro naše účely jsme použili základní *network deployment*, jehož schéma je vidět na obrázku 2.2 s tím rozdílem, že ve fázi pilotního provozu běží WebSphere Portal a *HTTP server* na stejném serveru. Jako registr uživatelů slouží univerzitní *LDAP* server, databáze portálu byla převedena do databáze *Oracle*, která běží na jednom z vyčleněných databázových serverů.



Obrázek 2.2: Základní network deployment

Celkem tedy budeme portál provozovat minimálně na třech počítačích: na jednom *WAS+WPS+IHS*, na druhém databázi a na třetím *LDAP*.

Instalace portálu proběhla na každé univerzitě mírně odlišně. Abychom se neopakovali, je v následujícím textu na každé univerzitě ilustrována jiná fáze instalace.

2.3. Instalace na UPOL

Prvotní instalace v Olomouci byla provedena dne 13.05.2004 a jednalo se o základní instalaci na operační systém Linux Red Hat 8.0. Instalace byla provedena podle základního instalačního scénáře pro WebSphere Portal 5.0.

1. Je nutno vypnout veškerou ochranu (firewall) na lokálním počítači; pokud k deaktivaci ochranné bariéry nedojde a instalační program zjistí jeho přítomnost, zobrazí se během instalace zpráva s odpovídajícím varováním.²
2. Spustíte následující příkaz z kořenového adresáře cdSetup (cd0):
`./install.sh`
3. Vyberte jazyk instalace a klepněte na OK. Zobrazí se uvítací panel.
4. Chcete-li zobrazit modul InfoCenter pro WebSphere Portal, můžete jej otevřít ve webovém prohlížeči klepnutím na *Spustit InfoCenter*. V opačném případě pokračujte klepnutím na *Další*.
5. Potvrďte souhlas s podmínkami licenčního ujednání. Klepněte na *Další*.
6. Instalační program kontroluje počítač na vstupní specifikace operačního systému a softwaru. Vyberte *Úplná* a klepněte na *Další*.³
7. Zadejte adresář, do kterého chcete instalovat WebSphere Application Server. Doporučujeme ponechat `/opt/WebSphere/AppServer`. Pokud zadaný adresář neexistuje, bude vytvořen. Pokud instalujete v systému Windows, nepoužívejte v instalační cestě tečky. Klepněte na *Další*.
8. Zadejte adresář, do kterého chcete nainstalovat IBM HTTP Server. Pokud zadaný adresář neexistuje, bude vytvořen. Klepněte na *Další*.
9. Určete následující informace pro WebSphere Application Server a klepněte na *Další*:
 - Jméno uzlu (node) v rámci buňky (cell) WebSphere Application Server, kam má být nainstalován aplikační server WebSphere Portal. Tato hodnota musí být jedinečná mezi ostatními názvy uzlů v téže buňce. Tato hodnota je zpravidla stejná jako jméno hostitele pro počítač. V této buňce budou vytvořeny dva procesy WebSphere Application Server: *WebSphere_Portal* a *server1* (viz obrázek 3.3).

²Tato dektekce ale není spolehlivá.

³Expertí IBM nedoporučují provádět uživatelskou instalaci "pokud opravdu nevíte, co děláte". To ale nikdy vědět nebudete, protože to není nikde dokumentováno. Uživatelská instalace v principu umožňuje nainstalovat WebSphere Portal bez základní konfigurace instalačním programem; administrátor potom vše musí udělat ručně.

- Úplné jméno hostitele nebo IP adresu počítače, na němž běží WebSphere Application Server. Doporučujeme zvolit jméno hostitele, ať zůstane v případě změny IP adresy cesta otevřena.
10. Zadejte adresář, do kterého chcete nainstalovat aplikaci WebSphere Portal. Pokud zadaný adresář neexistuje, bude vytvořen. Doporučujeme zanechat adresář `/opt/WebSphere/PortalServer`. Klepněte na *Další*.
 11. Zadejte ID uživatele a heslo administrátora WebSphere Portal. V poli ID uživatele ani v poli heslo nepoužívejte mezery a zajistěte, aby heslo obsahovalo alespoň pět znaků. Toto ID uživatele se po instalaci používá na přístup k WebSphere Portal s oprávněním administrátora. Pověšměte si, že toto ID uživatele se pouze používá na přihlášení do WebSphere Portal a nemá žádný vztah k jakémukoliv ID uživatele používaného pro přístup do operačního systému samotného. Klepněte na *Další*.
 12. Zkontrolujte komponenty, které se mají instalovat, a klepněte na *Další*. Instalační program zahájí instalaci vybraných komponent. Během celého procesu instalace a konfigurace zobrazuje instalační program indikátory postupu pro jednotlivé komponenty.
 13. Po skončení instalace zobrazí instalační program panel s potvrzením včetně seznamu komponent, jež byly nainstalovány. Než klepnete na *Dokončit*, pamatujte na následující:

Číslo portu používané pro přístup k WebSphere Portal je zobrazeno na posledním informačním panelu. Poznamenejte si jej pro potřeby ověřování adresy URL WebSphere Portal. Tato hodnota je také uložena ve vlastnosti `WpsHostPort` v souboru `<WPS_ROOT>/config/wpsconfig.properties`.
 14. Klepněte na *Dokončit*.
 15. Pro ověření, že WebSphere Portal běží, otevřete v prohlížeči následující adresu URL `http://<host>:<port>/wps/portal`, kde `<host>` je úplný název hostitele počítače, na němž běží WebSphere Portal a `<port>` je číslo portu zobrazené na panelu s potvrzením. Příklad: `http://portal.osu.cz:9081/wps/portal`.
 16. WebSphere Portal používá pro výchozí zpracování žádostí interní listener HTTP v rámci serveru WebSphere Application Server, ale máte-li k dispozici externí webový server, jako je IBM HTTP Server, můžete portál zkonfigurovat pro použití daného serveru. Adresa URL potom může vypadat například `http://portal.osu.cz:/wps/portal`.

Přesně podle tohoto postupu byla provedena prvotní instalace na serveru `jay.upol.cz` a byla předána kolegům z Olomouce k testování. Nebyla provedena migrace databáze z Cloudscape na jinou ani nebyl instalován žádný ověřovací mechanismus.

Následně byla zanechána na UPOL ještě instalační sada pro OS Windows, která by měla sloužit jako testovací (vývojářská) verze portálu WebSphere 5.0.

2.4. Instalace WPS 5.0.2 na OSU

Portál byl instalován postupně na třech operačních systémech. Poprvé pokusně na Windows 2000 Server, základní instalace se několikrát nezdařila. Zjištěny špatné instalační média a po jejich opětovném poskytnutí firmou IBM, instalace proběhla úspěšně. Nicméně v tomto případě se jednalo pouze o základní instalaci bez jakéhokoli upgradování.

2.4.1. Instalace na RedHat 8.0

Nainstalujeme WebSphere Portal 5.0 podle výše popsaného postupu. Instalujeme do standardního adresáře `/opt/WebSphere`. Podmínkou pro správnou instalaci jsou dostatečná práva uživatele, který instaluje, a dostatečné volné místo na HDD.

Jakmile máme nainstalováno (tj. úspěšně doběhne instalační program), pustíme se do patchování a fixování WAS. Postupně aplikujeme tyto fixpacky:

1. Was50_fp1_linux.zip
2. Was50_nd_fp1_linux.zip
3. Was50_pme_nd_fp1_linux.zip
4. Was50_fp2_linux.zip
5. Was50_nd_fp2_linux.zip
6. Was50_pme_nd_fp2_linux.zip
7. Was502_cf5_linux.zip

Fixpacky tvoří komplexní balíčky, každý z těchto balíčků obsahuje program `updateWizard.sh` a další potřebné programy pro instalaci. Proto vždy daný balík nakopírujeme do adresáře `/opt/WebSphere/AppServer/update`, tam spustíme skript `updateWizard.sh` a nainstalujeme daný fixpack.

Všechny fixpacky aplikujeme postupně jak jsou psané (mají na sebe návaznosti) a výsledkem by měla být verze WAS 5.0.2 + WAS Enterprise 5.0.2. Všechny fixpacky musí skončit úspěšným koncem, tedy „BUILD SUCCESSFUL“. U některých fixpacků jsme se setkali s nesprávným ukončením, poté je nejlepší daný fixpack odinstalovat, restartovat server a následně jej nainstalovat, pak se instalace podaří.

Před tím, než se budeme věnovat portálovému serveru musíme kromě fixpacků aplikovat také efíxy, které jsou nutné k další aktualizaci portálu. Proto vezmeme kompletní efíxy (máme uchováno v instalační sadě, kterou jsme si vytvořili) a nakopírujeme je opět do adresáře `/opt/WebSphere/AppServer/update`. Necháme v adresáři pomocné programy (zejména `updateWizard.sh`) a pomocí něj nainstalujeme všechny efíxy. Efíxy mají také svou posloupnost a doporučuji jít vzestupně od nejmenších čísel a skončit kumulativními fixy. Pokud se podaří vše nainstalovat máme aplikační server úspěšně updatován.

Ve chvíli, kdy je updatován aplikační server se může přejít k updatování serveru portálového:

1. V okamžiku instalace portálu existuje fixpack verze 2.0, který máme opět připravený v instalační sadě. Tento fixpack je ve formě zkomprimovaného souboru TAR a rozbalíme jej do adresáře `/opt/WebSphere/PortalServer/update`, který pravděpodobně nebude existovat, proto ho vytvoříme a rozbalíme tento zkomprimovaný soubor do něj. Následuje přesný postup instalace fixpacku číslo 2:
 - (a) Před instalací fixpacku je nutné, aby byl hlavní konfigurační soubor `wpconfig.properties` v naprostém pořádku. Chcete-li toto nastavení zálohovat, spusťte příkaz `./WPSconfig.sh -DparentProperties=jmeno_souboru -DsaveParentProperties=true`
 - (b) Musíme nastavit prostředí Javy pro aktualizaci pomocí souboru `/opt/WebSphere/AppServer/bin/setupCmdLine.sh`
 - (c) Následně je potřeba zastavit oba aplikační servery (server1, tak i server `WebSphere_Portal`) pomocí skriptu `/opt/WebSphere/AppServer/bin/stopServer.sh`
 - (d) Je nutno nastavit proměnnou `$WAS_HOME` na domovský adresář aplikačního serveru.
 - (e) Následně spustíme z adresáře `/opt/WebSphere/PortalServer/~/update/` příkaz `./updatePortal.sh -fixpack -installDir "/opt/WebSphere/PortalServer" -fixpackDir "/opt/WebSphere/PortalServer/update" -install -fixpackID WP_PTF_502`
 - (f) V této chvíli je potřeba zastavit HTTP server, pokud běží.
 - (g) A nyní se přistoupí k nejdelšímu a největšímu kroku z adresáře `/opt/WebSphere/PortalServer/config` spustit příkaz `./WPSconfig.sh WP-PTF-502 -DportalAdminPwd=~/PortalAdminPassword` (my jsme dávali na počátku heslo `wpsadmin`), heslo administrátora obsaženo v souboru `wpconfig.properties`.

Pokud dojde během průběhu tohoto příkazu k jakékoli chybě, tak je nutno tento krok totálně odinstalovat příkazem `./WPSconfig.sh UNINSTALL-WP-PTF-502` a po opravě chyby vybrané v logu opět spustit příkaz ze začátku tohoto kroku.
 - (h) Po nakopírování fixpacku 5021 spustíme příkaz `/opt/WebSphere/PortalServer/update> ./updatePortal.sh -fixpack -installDir "/opt/WebSphere/PortalServer" -fixpackDir "/opt/WebSphere/PortalServer/update" -install -fixpackID WP_PTF_5021`
 - (i) V případě provozování externího web serveru je třeba jej restartovat.
 - (j) Následně spustíme příkaz `./WPSconfig.sh WP-PTF-5021 -DPortalAdminPwd=portalAdminPassword`
 - (k) Odinstalace stejně jako předchozí fixpack...

Pokud vše proběhlo bez problémů, máme portál aktualizován na verzi 5.0.2, stejně tak jako WAS a WAS Enterprise by měli být ve verzi 5.0.2.

2.4.2. Instalace na SuSE Linux – SLES8

Nainstalujte balíčky: openmotif-2.2.2-124 a lesstif-0.93.14-205
Problém nepodporované verze operačního systému ignorujeme, náš systém není certifikován, ale měl by být podporován. Stáhněte soubor `prereqChecker.dtd` a soubor `prereqChecker.xml` z adresy http://www-1.ibm.com/support/docview.wss?rs=180&context=~/SSEQTP&q=&uid=swg24004743&loc=en_US&cs=utf-8&lang=en+en

Se standardně dodávanou Javou (`/usr/lib/java`) instalace nefunguje (nenainstaluje aplikační server), nutno pozměnit jak proměnnou `$JAVA_HOME` a další, tak i v proměnné `$PATH` nastavit naši Javu před původní. Systémové proměnné musí být definovány implicitně (při startu shellu musí už být nastaveny).

Pokud při instalaci aplikačního serveru dojde k chybovému hlášení v logu v `/opt/WebSphere/PortalServer/log` „Java not found“, pak je nutné při nové instalaci vytvořit adresář `/opt/WebSphere/AppServer/java` a do něj zkopírovat funkční verzi Javy.

Po úspěšné instalaci nainstalujeme FixPacksy a eFixies stejně jako u Linux RedHat 8.0.

2.4.3. Nastavení IBM HTTP serveru (IHS)

WebSphere Portál využívá pro HTTP přenosy listenery (tzv. *endpoints*) aplikačního serveru, který HTTP požadavky zpracovává. Z nejrůznějších důvodů je doporučováno nainstalovat externí webserver (IBM HTTP Server) a nakonfigurovat portál tak, aby využíval tento externí webserver. Můžeme využít jak lokálního tak i vzdáleného serveru (instalovaný na jiném počítači). Pro komunikaci s WAS/WPS používá IHS modul `ibm_app_server_http_module` řízený konfiguračním souborem `plugin-cfg.xml`

Konfigurace portálu pro lokální webserver: V adresáři `<wp>/config` otevřeme soubor `wpsconfig.properties` a nastavíme

```
WpsHostName = jméno počítače  
WpsHostPort = 80
```

Provedeme `./WPSconfig.sh httpserver-config` a restartujeme server WebSphere_Portal i IHS.

Další dokumentace viz <http://publib.boulder.ibm.com/pvc/wp/500/~\smb/en/InfoCenter/index.html>

2.5. Instalace na ZČU

Instalovali jsme produkční portál na platformě Linux, vývojový a pokusný portál na platformě Windows 2000.

Následující text shrnuje několik našich postřehů z instalace na RedHat Linux 8.0:

Instalační JRE občas nevykreslí některá tlačítka nebo chybně vykreslí texty v národní abecedě. Instalujeme zásadně z anglického nastavení systému

nebo použijeme v shellu příkaz `export LANG=C`

Během instalace vypněte firewall, zvláště, má-li stoj více síťových rozhraní. Různé komponenty WebSphere Portal a WebSphere Application Server se napojují na různé zdrojové IP, což může vyvolat nepředvídatelné TCP spojení, které administrátor firewallu neočekával.

2.6. Fixpacky a patche

Z instalačních médií (je jedno zda použijeme CD-ROM z mediapacku nebo stáhneme obrazy z Passport Advantage) nainstalujeme vždy původní verzi produktu, do které je potřeba doinstalovat opravné balíčky, kterých je několik druhů (fix, eFix, FixPack, interim fix, ...) Situaci dále komplikuje fakt, že je potřeba záplatovat aplikační server i portálový server (jako enterprise aplikaci běžící v aplikačním serveru).⁴

Aplikace fixpacků a patchů je velice choulostivou záležitostí, protože se od ní odvíjí dost podstatná část správné funkčnosti WebSphere Portal. Aktuální verze fixpacků jsou k dispozici na adrese <http://www-106.ibm.com/developerworks/websphere/zones/~portal/proddoc.html#500>, což je oficiální stránka IBM o nejnovějším vývoji v produktu WebSphere Portal 5.0. Naleznete zde nejnovější fixpacky, hardwarové a softwarové požadavky, release notes a informační centra k jednotlivým verzím produktu. IBM se snaží dokumentaci lokalizovat do mnoha jazyků, proto zde s mírným zpožděním můžete nalézt také dokumentaci v českém jazyce.

Strategií IBM je rozdělení updatů do dvou forem a to patchů, který zastává roli konkrétní záplaty na daný problém, tyto patche mají názvy PQ91124 a podobně. Druhou formou je kompletní fixpack, který je vlastně sdružením všech předchozích patchů do jednoho velkého fixpacku. Fixpack je popsán například takto :

Cumulative fixes for WebSphere Portal 5.0.2.2

```
70944 iSeries: Code is throwing an exception because of invalid
      CCSID wps.base.install
74744 WSRP Doesn't detect if a portlet is integrated
79193f_2 update-portlets-5021 generates 15 errors in cluster.
80658 XMLC: XML 1.1 (JDK1.4) compliance
80677 Selfcare not working due to stricter JSP handling in WAS 5.1
80679 WSRP: SSL with client certificate fails
88815 Cluster: Failover does not work when managing page
82313 Xmlaccess Externalize=true fails with no logging
83419 Virtual Resource extensions made in 5021 for LWP incomplete
83709 CST WSRP Credential Vault connect be used by WSRP
84022 TranHelper should catch Error not Exception to function
      properly
84044 Producer seems not handle cloneBeforeWrite correctly
84193 Remove worldclock from TestPortlets
84343 Fix TranManager to work with WAS 5.0.2 and WAS 5.1
84368 Ensure all JDBC resources are closed
```

⁴Dále jsme ještě zjistili, že je potřeba upgradovat i instalované portletové aplikace, které je však potřeba stáhnout z IBM Portlet katalogu, a které nejsou zahrnuty do systémů záplat na WebSphere Application Server a WebSphere Portal.

85222 Blank help page shown in consume or install mode
86199 Remove Financial Times portlet from WebSphere Portal
86753 Update WCL to support WBISF 5.1
PQ85982 Need to bounce portal server after war update
PQ86371 wmm_encrypt.sh fails on iseries
PQ86849 Defect 81146 Event is not triggered by the
UpdatePortletApplicationCommand
PQ87167 ArrayIndexOutOfBoundsException exception during xml export
PQ87270 Extend the PUMA User and Group Objects to support Multivalued
PQ87323 Problems with URL mappings
PQ87531 Deadlock occurrences when running load tests with
Microsoft SQL Server
PQ87997 ArrayIndexOutOfBoundsException in cluster.
PQ88188 Defect 79188 for PQ88188
PQ88221 Portal elements not updated in site analyzer
PQ88359 getCallbackHandler() is defined static
PQ90323 Portlets cannot be edited when page is in edit mode

Zde můžete vidět podrobný popis fixpacku WebSphere Portal 5.0.2.2, který vlastně sdružuje jednotlivé patche (fixy), proto je fixpack také nazýván *cumulative fixem*. V aplikaci jednotlivých fixpacků existuje posloupnost, která se musí dodržet, zde jsou pravidla, které musíte dodržet:

- vycházíme z verze portálu 5.0, musíme rozdělit fixování na 2 části, fixování aplikačního serveru a fixování portálového serveru
- aktualizace aplikačního serveru musí postupovat po kompletních fixacích až do posledního nalezeného fixpacku a to paralelně pro oba servery, jak normální aplikační server, tak také pro aplikační server Enterprise. Takže z verze 5.0 se dostanete na 5.0.1 u verze normální i Enterprise, poté na 5.0.2 a až pokud nemáte další kompletní fixpack, začnete fixovat jednotlivé dílčí fixy a dostanete se například na verzi 5.0.2.
- stejný postup fixování použijete i u serveru portálového

K instalování fixpacků, či fixů je určen nástroj Update Wizard, který vám umožní pohodlnější uživatelské ovládání. Zde je stručný popis jak aktualizovat pomocí Update Wizardu:

1. zastavete server1: `./stopServer.sh server1`
2. zastavete WebSphere_Portal: `./stopServer.sh WebSphere_Portal`
3. zastavete webserver
4. vytvoříte adresář `<WAS_ROOT>/update`
5. zkopírujete do tohoto adresáře rozbalený fixpack
6. nastavte proměnnou `$JAVA_HOME`:
`export JAVA_HOME=/usr/WebSphere/AppServer/java`
7. spustíte `./updateWizard.sh`
8. vyberte si server k updatu a klikněte na *Next*
9. vyberte si *Install fix packs* a stiskněte *Next*

10. počkejte si na oznamovací zprávu, že se vše nainstalovalo úspěšně
11. pokud nastaly jakékoli problémy, projděte logy
12. pokud bylo vše v pořádku, nainstalujte oba servery

Stejným způsobem se nainstaluje jakýkoli fixpack, či patch který IBM vydá. UpdateWizard je dodáván přímo s fixpacky. Dejte si pozor na nové verze UpdateWizardu, některé starší verze nefungují s novějšími FixPacky. Stejně tak v některých případech se stává, že UpdateWizard začne hlásit výjimku a nemůže korektně nainstalovat fixpacky (zvláště pokud instalujete několik fixpacků za sebou), pak spolehlivě pomůže ukončení aplikace, restart serverů a nové spuštění.

2.7. Poznámky

- instalační program si ukládá informace do adresáře `/tmp` a `/root/wpd.properties`
- raději instalujte z CD, při instalaci z CD nakopírovaných na pevný disk (což je normálně funkční a výrazně rychlejší postup) instalace z nevyjasněných důvodů zamrzala
- verze 5.02 využívá disky CD-ROM verze 5.0 a lze je instalovat přímo; během instalace se aplikují další fixy z disku CD1-6

PORT@L – technická zpráva projektu FR CESNET 043R1/2003

Administrace WAS

Než přistoupíme k nastavování portálu, je potřeba porozumět systému konfigurace a vzbít na okolí.

Obsah

3.1. Administrační konzole WAS	19
3.2. Systém konfiguračních souborů	22
3.2.1. Hierarchie adresářů	22
3.2.2. Konfigurační soubory	23
3.3. Uživatelské registry portálu	23
3.3.1. Vztah registrů WAS a WPS	25
3.3.2. WebSphere Member Management (WMM)	25
3.3.3. Jak probíhá autorizace	26
3.4. Instalace JDBC ovladačů	26
3.5. Instalace JCO	27
3.5.1. Postup instalace	27
3.5.2. Nastavení v SAP	27

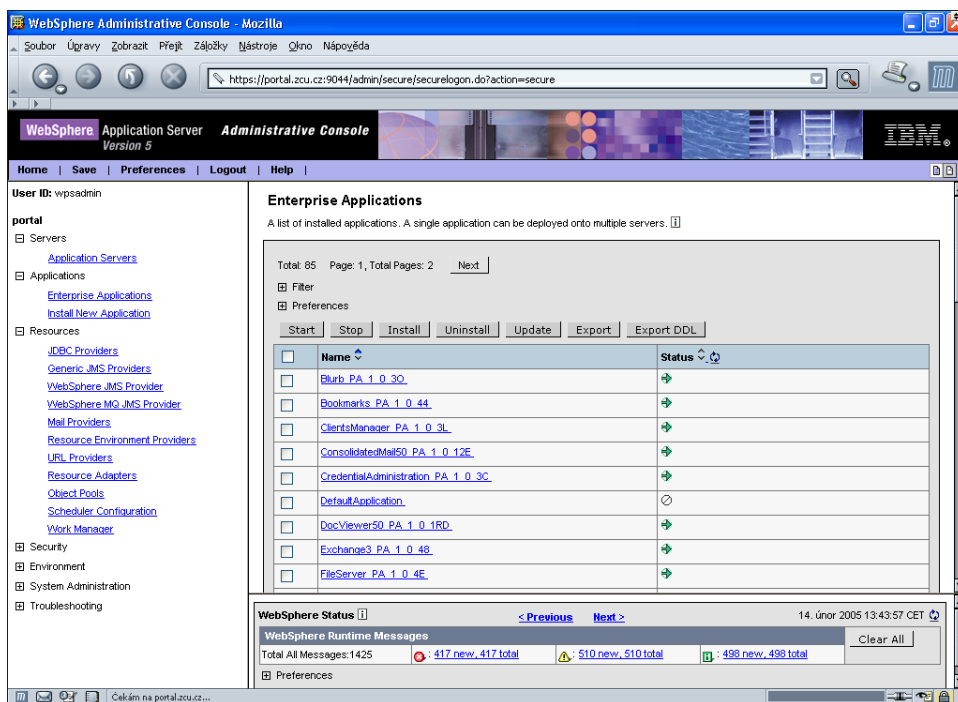
Ve srovnání s opensource aplikačními servery (Tomcat, JBoss) je WebSphere Application Server mnohem složitější a začínající administrátor zpočátku vůbec netuší, kde začít a co vlastně konfiguruje. V této kapitole se proto pokoušíme stručně vysvětlit princip správy WAS tak, jak jsme jej sami postupně po kouscích odhalovali.

V závěru kapitoly je návod na instalaci JDBC ovladačů pomocí administrační konzole.

3.1. Administrační konzole WAS

Webová administrační konzole (obr. 3.1) je nástrojem, který pomocí webového rozhraní umožňuje přístup ke konfiguračním souborům WAS ve formátu XML – viz obrázek 3.2. Při dostatečné znalosti konfiguračních souborů není potřeba administrační konzoli používat. Provedení jakékoli úpravy konfigurace změní příslušný konfigurační soubor, proto je pro projevení změn přímo na aplikačním serveru obvykle nutno server restartovat.

Administrační konzole umožňuje komfortnější editaci parametrů aplikačních serverů potřebných pro funkci portálu, než editace konfiguračních souborů. Webové rozhraní je logicky členěno a k příslušným atributům se dostanete poměrně přímou cestou.



Obrázek 3.1: Uživatelské rozhraní administrační konzole

Každá instance aplikačního serveru má na jednom portu spuštěnou administrační konzoli, která se napojuje na správce konfigurace buňky (viz obrázek 3.3). Po instalaci kompletního portálu jsou přístupné dvě administrační konzole, jedna od procesu server1 (port 9090 nebo 9043 při zapnutém SSL) a druhá od instance portálového serveru (port 9091 nebo 9044 při zapnutém SSL).

Administrační konzole je logicky členěna do sedmi částí:

- **Servery**

První část se věnuje aplikačním serverům v buňce, kterou obhospodaruje tato konzole. Pro běžný provoz portálu jsou potřeba dvě instance aplikačního serveru. První, standardně pojmenovaný *server1*, je aplikační server, který zajišťuje běh konfiguračních nástrojů portálu, demo aplikací, ukázek technologií a podobně.

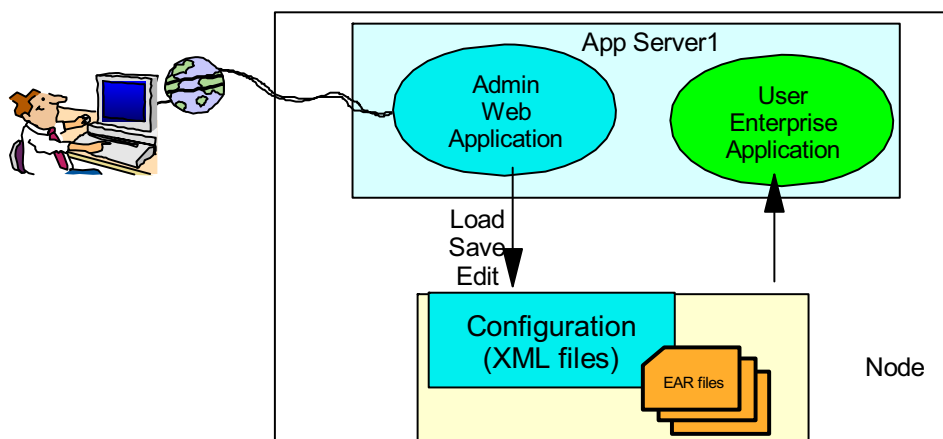
Druhým server je standardně nazývaný *WebSphere Portal*. Základní aplikace tvoří sada pomocných servletů a hlavně webová aplikace samotného portálu *wps*. Portálový server provozuje portlety jako webové aplikace v tomto serveru.

- **Aplikace**

V této sekci se nacházejí samotné enterprise aplikace. Každý nově instalovaný portlet (resp. portletová aplikace) pomocí portálového administračního rozhraní se objeví na tomto místě. Aplikace je možno instalovat, spouštět, zastavovat, mazat a měnit jejich nastavení. Každá aplikace je namapována na nějaký server nebo cluster, na kterém běží.

- **Zdroje**

Definice zdrojů (resources) je významným rysem J2EE architektury.



Obrázek 3.2: Funkce administrační konzole

Tato část konzole obhospodařuje zdroje dat, které je možno nadefinovat v rámci aplikačního serveru. V rámci našeho projektu jsme využívali datasources založené na JDBC providerech. Jelikož se naše aplikační data nacházejí v databázích Oracle, zvolili jsme datasources na technologii tenkého JDBC klienta pro databáze Oracle. Vytvoření datasource popisujeme v kapitole 7.9.

- **Bezpečnost**

Pomocí administrační konzole je možno definovat bezpečnost stejně tak, jako pomocí konfiguračních souborů. Je potřeba si uvědomit, že celý proces zapínání bezpečnosti je obtížnější záležitostí a v rámci administrátorské konzole je možno provést pouze prvotní konfiguraci a následné kroky zapnutí bezpečnosti pro portál je nutné provést z příkazové řádky skriptem `WPSconfig.sh`.

Podrobněji viz [5].

- **Prostředí**

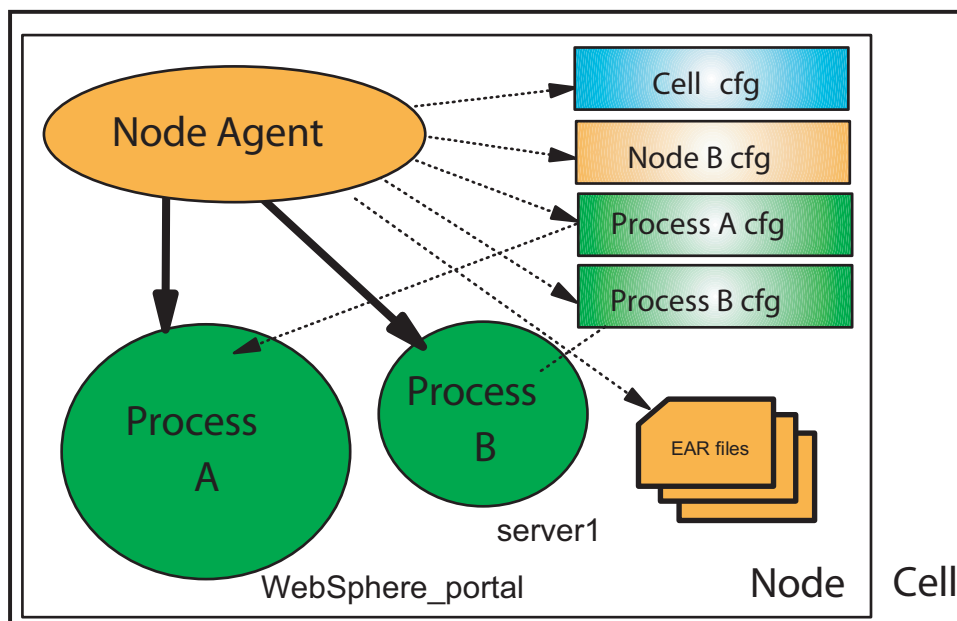
Už z názvu této sekce vyplývá, že zde jsou definovány proměnné prostředí, které zde lze editovat, zvláštní pozor si dávejte na to jak jsou nastavené proměnné JVM, na to je portál opravdu háklivý. Dále je zde možnost konfigurovat plug-in, který se exportuje pro HTTP server z aplikačním serveru.

- **Systémová administrace**

V této části se definuje správa aplikačních serverů, to znamená definice uživatelů nebo skupin uživatelů, kteří mají možnost vstupovat do administrátorské konzole. V prvopočátku instalace, když ještě není napojen externí ověřovací mechanismus, se uživatelé definují ručně. Po zapnutí global security se zde pouze vyjmenují příslušní uživatelé.

- **Problémy**

Pod záložkou Problémy se skrývá kompletní nastavení logovacích a trasovacích souborů aplikačních serverů. Je zde možno nastavit si jména souborů kam se zprávy ukládají a intervaly rotace. Také zde můžete nastavit, co všechno se má logovat. Pro optimalizaci logování je potřeba nastudovat redbook [2].



Obrázek 3.3: Runtime konfigurační prostředí uzlu (node)

3.2. Systém konfiguračních souborů

WebSphere Application Server má konfiguraci v mnoha dokumentech uložených v adresářové struktuře, kam je aplikační server nainstalován. Konfigurační soubory popisují samotný aplikační server, jeho konfiguraci a nastavení jeho částí. Naprostá většina konfiguračních souborů je ve formátu XML.

3.2.1. Hierarchie adresářů

Hierarchie adresářů podporuje replikaci více uzlů jednoho aplikačního serveru, tak aby bylo možno synchronizovat aktivity v rámci jedné buňky. Tento systém je využit v rámci síťové instalace (network deployment) celého portálového řešení, kdy jsou využívány dva aplikační servery.

V případě fyzicky oddělených serverů má každý své konfigurační soubory uloženy na svém hardwaru, ale po změně konfigurace se soubory distribuovaným způsobem aktualizují na všech příslušných místech v rámci jedné buňky.

Na začátku celé hierarchie adresářů se nachází adresář `cells`. Tento adresář obsahuje podadresář pro každou buňku; jména těchto podadresářů odpovídají jménům buněk. Celé toto nastavení závisí od hodnot zadaných při instalaci portálu. Pokud máte aktivován jenom jeden uzel v rámci vašeho portálu, pak naleznete v dalším podadresáři jménem `nodes` další podadresář se jménem vašeho uzlu. Zde bude další podadresář `servers` ve kterém se nalézají konfigurace vašich aplikačních serverů – v případě standardní instalace naleznete podadresáře `server1` a `WebSphere_Portal`.

3.2.2. Konfigurační soubory

Soubory `security.xml`, `virtualhosts.xml`, `resources.xml` a soubor `variables.xml` jsou použity ke konfiguraci celé buňky a zajišťují nastavení bezpečnosti, virtuálních adres buňky a podobně. Tyto soubory jsou uloženy v adresáři se jménem dané buňky.

Pokud instalujete portál s podporou clusteringu, pak najdete podadresář `clusters` a dále jména vytvořených clustrů. Každý podadresář obsahuje soubor `cluster.xml` obsahující konfigurační data pro příslušný cluster.

Podadresář `nodes` obsahuje informace pro příslušný uzel. Jména podadresářů opět odpovídají názvům jednotlivých uzlů. Každý podadresář uzlu obsahuje soubory `variables.xml` a `resources.xml`, které udržují konfigurační data v rámci jednoho uzlu. Stejně soubory se nacházejí také v rodičovském adresáři vaší buňky. Význam tohoto zdvojení souborů je pro distribuované šíření parametrů a zobecnění funkce konfiguračních nástrojů. Soubory mají vždy stejný název, ale nemusí obsahovat stejné informace; v rámci jedné buňky mohou být příslušné uzly individuálně dokonfigurovány.

Pokud v rámci uzlu přepíšete parametr, který byl definován v buňce, bude platit tato lokální konfigurace, pokud ovšem některý parametr v rámci uzlu vůbec nekonfiguruje, pak platí nastavení celé buňky.¹

Každý adresář uzlu obsahuje jeden podadresář pro každý server, jména opět odpovídají názvům serverů. Každý takový adresář obsahuje soubor `server.xml`, který popisuje specifikaci daný server. Dále tyto podadresáře mohou obsahovat soubory `security.xml`, `resources.xml` a `variables.xml`, které obsahují specifická data pouze pro tento daný server. Konfigurace specifikovaná pro server opět přepíše parametry specifikované v rámci rodičovských adresářů.

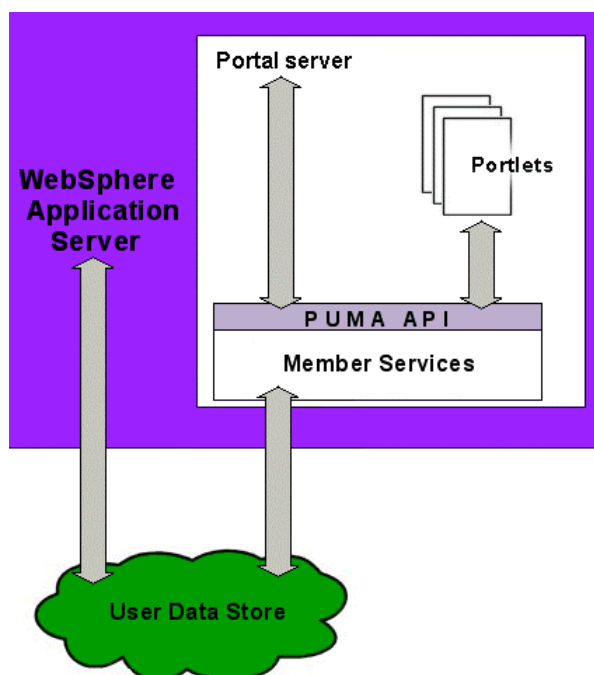
Poslední podadresář se jmenuje `applications` a obsahuje podadresáře pro každou aplikaci, která je v rámci buňky nainstalována. Jména adresářů jsou specifikována pomocí názvů implementovaných aplikací. Každý podadresář nainstalované aplikace obsahuje soubor `deployment.xml`, který uschovává data o instalaci aplikace. Každý adresář aplikace obsahuje podadresář `META-INF`, ve kterém jsou uloženy tzv. deskriptory (soubory popisující tuto aplikaci pro aplikační server) J2EE, dále soubory popisující specifika instalace na aplikační server IBM a soubory návazností. V rámci adresáře aplikace je vytvořen také adresář pro každý web archiv (WAR soubor). Binární soubory (například JAR) jsou také součástí těchto adresářů.

Takto tedy vypadají konfigurační soubory z pohledu aplikačního serveru (WAS, WebSphere Application Server). Konfigurační soubory portálového serveru, resp. aplikace „portál“ (`wps`, WebSphere Portal) jsou podrobněji komentovány na straně 52.

3.3. Uživatelské registry portálu

Aplikační server i portál využívají k autentizaci a autorizaci uživatelů registry uživatelů, obvykle LDAP. Jedná se asi o nejsložitější a nejméně přehlednou

¹Jinými slovy: parametry se dělí shora dolů, přičemž lokální nastavení přepíše shora děděnou hodnotu.



Obrázek 3.4: Struktura member services

část celého portálu: WAS i WPS mají každý svoji konfiguraci, přičemž uživatel se musí postupně ověřit ve WAS i WPS. WPS navíc využívá přes rozhraní PUMA (*Portal User Management*) a WMM (*WebSphere Member Management*) registr WAS – WMR (*WebSphere Member Registry*). Další problémy působí trojí *uid*, komplexní mechanismy vyrovnávacích pamětí, rozdíly mezi LDAP servery a také různá podpora implementace registru ve WAS a WPS.

Registr může být obecně buď LDAP server, databáze, operační systém (*/etc/passwd*) nebo custom registry. Pokud LDAP schéma nepodporuje všechny atributy, je možné přesto použít LDAP a část atributů namapovat do *lookaside database*.

Základní struktura vazeb na registr je na obrázku 3.4, pro hlubší porozumění je nezbytné si přečíst kromě [1, 2, 4, 5] zejména [23, 36].

Nejdříve se podívejme, jak se používají uživatelské registry portálu a co se od nich vyžaduje:

- validace uživatelského jména a hesla:
 - webová autentizace, v případě že není použito TAI
 - administrační konzole, aplikační servery a J2EE klienti stále potřebují uživatelské jméno a heslo
- dotazy o uživateli
 - unikátní ID uživatele – jedná se o plné DN (*distinguished name*) získávané z LDAP serveru
 - jméno uživatele – typicky CN nebo UID
 - dotazy na sadu ostatních atributů standardně ukládaných s uživatelem

- Dotazy na skupiny
 - členství jednotlivých uživatelů ve skupinách
 - dotazy na jména skupin, či na jiné atributy

Podporované operační systémy, LDAP servery a jiné ověřovací mechanismy viz infocenter.

3.3.1. Vztah registrů WAS a WPS

Jelikož běh portálu vyžaduje aplikační i portálový server, vyvstává otázka, jak tyto servery využívají uživatelské registry:

Oba servery potřebují jeden uživatelský registr, ke kterému se přistupuje. Aplikační server využívá registr k prvotní autentizaci, portálový server jej aktivně využívá k autentizaci, autorizaci, členství ve skupinách, čtení všech atributů...

Protože oba servery mají vlastní napojení na registr, je možno provozovat jednu z těchto dvou konfigurací:

1. Společný LDAP server pro oba servery (mezi LDAP serverem a portálem však stojí WMM, které je zkonfigurováno tak, aby využívalo LDAP).

Podporovány jsou Active Directory, Sun ONE, IBM Directory, Domino a Novell eDirectory. Ostatní LDAP servery je možno použít, pokud se vám povede nakonfigurovat dohromady LDAP server funkčně proti aplikačnímu serveru. Podmínkou je podpora schéma *inetOrgPerson*.

2. Portálový server používá LDAP (nebo LDAP a lookaside databázi) a WAS používá buď jiný LDAP, nebo operační systém (*/etc/passwd*). Tato konfigurace není doporučována.

3.3.2. WebSphere Member Management (WMM)

WebSphere Member Management je komponentou WebSphere Portal, která je určena ke správě uživatelů a skupin. Uživatelé a skupiny jsou asociovány ve WMM jako členové. Member manager uchovává atributy ke každé skupině či uživateli, nikoli však spojení s jednotlivou rolí k výslednému objektu. Následující vlastnosti jsou spojeny s WMM:

- Správa profilů: Administrátor spravuje profily všech objektů pomocí portletů určených pro správu zdrojů.
- Uživatelské úložiště: Uživatelské úložiště je kolekcí uživatelských profilů, skupinových profilů a informací o organizaci. Data uživatelského úložiště mohou být uloženy v databázi, nebo v LDAP serveru.
- Členství ve skupině: WMM obhospodařuje členství uživatelů ve skupinách. Členství ve skupině může být použito k přístupu k jakémukoli objektu v rámci portálu (portlety, stránky, portletové aplikace, ...).

3.3.3. Jak probíhá autorizace

Autorizace je důležitou funkcí portálů, proto si pojdme přiblížit, jak to funguje v rámci WebSphere Portal:

Aplikační server má založenu autorizaci na J2EE rolí:

- role jsou definovány skrz J2EE aplikace
- role jsou mapovány v průběhu instalace aplikace
- veškerá data jsou ukládány v XML souborech

Portálový server má autorizaci založenu portálových rolích:

- role jsou pevně dány, jejich přiřazení se realizuje v administračním rozhraní portálu (viz 5.3.2.)
- jednotlivé role jsou přiřazeny všem objektům, které se vyskytují v portálu jako stránky, portlety atd...
- data jsou ukládána do portálové databáze a tyto záznamy mají plnou návaznost na LDAP server

3.4. Instalace JDBC ovladačů

Knihovny s JDBC ovladači se pro konkrétní databázi musí vybírat podle verze *JVM – Java Virtual Machine* pro kterou jsou určeny. Portál obsahuje JVM verze 1.3 a JDBC ovladače jsou obvykle nabízeny pro poslední verze JVM.

JDBC ovladače by měly obsahovat podporu pro sdílení datových propojení², aby je bylo možno použít při konfiguraci datových zdrojů v aplikačním serveru Websphere.

Knihovny s JDBC ovladači mohou být buď Java nativní, (pak je jedno na jakém operačním systému se použijí) nebo mohou obsahovat systémově závislou knihovnu.

U systémově závislých knihoven se při instalaci musí postupovat podle návodu dodavatele ovladače (umístění a instalace knihovny systému). Nativní knihovny je vhodné umístit do adresáře `<WAS_ROOT>/lib/`.

Po restartu aplikačního a portálového serveru jsou pak knihovny přístupné pro konfiguraci datových zdrojů a pro přímé použití z portletů/servetů aplikací.

V portálech byly tímto způsobem nainstalovány JDBC ovladače pro přístup k databázi *Oracle* (`classes12.jar`), na portálu UP Olomouc navíc ještě i k databázi *Sybase* (`jconn2.jar`).

Datové zdroje se definují v administrativní konzoli aplikačního serveru Websphere. Postup je podrobně popsán v kapitole 7.9.

²connection pool

3.5. Instalace JCO

Instalace *Java SAP Connector* je nutná pro nativní přístup k aplikaci SAP. Soubory potřebné pro instalaci jsou k dispozici na webových stránkách SAP určených pro klienty a partnery.

3.5.1. Postup instalace

Postup instalace není složitý a pro jednotlivé operační systémy se liší jen minimálně:

Windows:

- nakopírovat knihovnu `librfc32.dll` do adresářů `<WAS_ROOT>\bin` a `<WPS_ROOT>\bin`
- nakopírovat soubory `sapjco.jar` a `sapjcorfc.dll` do adresářů `<WAS_ROOT>\lib` a `<WPS_ROOT>\shared\app`
- restartovat aplikační server

Linux:

- nakopírovat soubor `librfccm.so` do adresářů `<WAS_ROOT>/bin` a `<WPS_ROOT>/bin`.
- nakopírovat soubory `sapjco.jar` a `libsapjcorfc.so` do adresářů `<WAS_ROOT>/lib` a `<WPS_ROOT>/shared/app`
- restartovat aplikační server

3.5.2. Nastavení v SAP

Uživatel, který přistupuje z portálu do SAP přes JCO konektor, musí mít v SAP přiřazen do rolí objekt `S RFC` s parametry:

1. *Činnost:* *
2. *Název objektu RFC:* SLST
3. *Typ objektu RFC:* FUGR

Napojení na infrastrukturu

Úspěšnou instalací produktu práce nekončí, ale naopak začíná; je třeba portál napojit na infrastrukturu organizace, což je opět netriviální úkol, tentokrát nejen pro administrátora portálu.

Obsah

4.1. Migrace do databázového systému Oracle	30
4.1.1. Příprava databáze – ZČU	30
4.1.2. Migrace – ZČU	32
4.1.3. Nastavení a Migrace do Oracle – OSU	36
4.2. Nastavení LDAP	38
4.2.1. OpenLDAP na ZČU	38
4.2.2. Poznámky	39
4.2.3. Novell eDirectory na OSU	39
4.3. Generování SSL klíčů a IKEYMAN	39
4.4. Zapnutí Global Security	40
4.5. Zapnutí HTTPS browser-IHS (ZČU)	40
4.5.1. Nastavení IHS	40
4.5.2. WAS web admin	41
4.5.3. httpd.conf	41
4.5.4. Přihlašování do WPS	41
4.5.5. Finalizace	41
4.6. Zapnutí HTTPS browser-IHS (OSU)	42
4.6.1. Nastavení IHS pomocí konfiguračního souboru	42
4.6.2. Nastavení IHS pomocí konzole (doporučeno)	43
4.7. Zapnutí SSL WAS-LDAP a WAS-IHS (ZČU)	43
4.7.1. Registrace CA chain	44
4.7.2. Úprava konfiguračních souborů	44
4.8. Zapnutí SSL WAS-LDAP a WAS-IHS (OSU)	44
4.8.1. Úprava Web server plug in souboru	45
4.8.2. Nastavení portálu pro LDAP SSL	46
4.8.3. Nastavení portálu pro SSL provoz	47
4.9. Některé problémy	47
4.9.1. Novell a WPS5.0	47
4.9.2. Problém komunikace SSL	49
4.9.3. Problémy s Oracle databází	49

Bez napojení na IT infrastrukturu konkrétní organizace by portál nepřinesl velký užitek, proto se této problematice poměrně podrobně věnujeme. Na první pohled by se mohlo zdát, že stačí jen postupovat podle manuálu, ale v praxi se ukázalo, že to zdaleka není tak jednoduché. Dostupné návody často nejsou vyčerpávající nebo se nevztahují k poslední verzi software; návody se často zaměřují na konkrétní izolovaný konfigurační poskup, ale nepodávají celkový pohled na problematiku. Někdy je problém v samotném produktu a je třeba čekat na fix, který chybu odstraní.

Uspokojivé vyřešení infrastrukturálních návazností nám zabralo několik měsíců a stejně jsme nedosáhli plně uspokojivého výsledku. Tato kapitola tedy není přepis manuálu, ale kompilací našich současných znalostí, které jsme pracně vydolovali z mnoha informačních zdrojů a ověřili v nesčetných cyklech pokus–omyl.

4.1. Migrace do databázového systému Oracle

Migraci agendy WebSphere Portal z databáze *CloudScape* do jiného databázového systému lze uskutečnit ve dvou krocích: příprava databáze a migrace dat. Jak se později ukázalo u WebSphere Portal 4.2 – v ideálním případě. Někdy lze totiž nabýt dojmu, že standardy interoperabilitu nezajišťují, ale jen ji pouze nevyklučují.

V následujících odstavcích je pro ilustraci popsáno řešení technických problémů při migraci agendy WebSphere Portal verze 4.2. Na tomto místě však musíme i zdůraznit, že u verze 5.0 jsme migraci již mohli uskutečnit bez potíží, ale zde popsané postupy nadále platí.

4.1.1. Příprava databáze – ZČU

Před vlastní migrací agendy portálu (tzv. databázového backendu) do relační báze dat firmy Oracle je nutné tuto bázi dat nejdříve vytvořit. Předpokladem je, že je již k dispozici server s nainstalovaným databázovým serverem Oracle verze 8 a vyšší.

WebSphere Portal vyžaduje UTF8 jako nativní znakovou sadu databáze, kterou je nutné nastavit již při definici databáze:

```
create database "WPS50"
  maxinstances 1
  maxlogfiles 16
  maxdatafiles 50
  character set "UTF8"
  datafile
    '/home/oracle/data/WPS50/sysWPS50.dbf' size 100M
    autoextend on next 20M maxsize 500M
  logfile
    '/home/oracle/data/WPS50/log1WPS50.dbf' size 2M,
    '/home/oracle/data/WPS50/log2WPS50.dbf' size 2M,
    '/home/oracle/data/WPS50/log3WPS50.dbf' size 2M,
    '/home/oracle/data/WPS50/log4WPS50.dbf' size 2M;
```


Tímto skriptem vytvoříme databázi (v terminologii *Oracle* tzv. databázovou instanci) identifikovanou řetězcem `WPS50`, označovaným jako *SID* – *Service ID*.

V souboru `init.ora` nastavíme také rovnou parametr `open_cursors` na hodnotu 800, abychom se vyhnuli chybě `ORA-01000: maximum open cursors exceeded` během migrace.

V současné době běží databázový server na serveru `deucalion.zcu.cz` a jeho aktuální verze je *Oracle9i Enterprise Edition Release 9.2.x.x.x – Production*, jméno instance je `WPS50`, Oracle Listener naslouchá na portu s číslem 1526. *Java DataBase Connectivity (JDBC) URL*, které je nutno uvádět v konfiguraci datových zdrojů tedy bude vypadat: `jdbc:oracle:thin:@deucalion.zcu.cz:1526:WPS50`

Posledním krokem na straně databázového serveru je vytvoření uživatelských účtů (v terminologii *Oracle* *schémata*), které WebSphere Portal využívá. K tomuto poslouží následující SQL skript:

```
drop user db2admin cascade;
drop user wpsdbusr cascade;
drop user wmmdbusr cascade;
drop user pznadmin cascade;
drop user ejb cascade;
drop user wcmdbadm cascade;
drop user feedback cascade;

create user db2admin identified by "tajne-heslo"
default tablespace usr_data
temporary tablespace tmp
profile default;

grant connect, resource to db2admin;

create user wpsdbusr identified by "tajne-heslo"
default tablespace usr_data
temporary tablespace tmp
profile default;

grant connect, resource to wpsdbusr;

create user wmmdbusr identified by "tajne-heslo"
default tablespace usr_data
temporary tablespace tmp
profile default;

grant connect, resource to wmmdbusr;

create user pznadmin identified by "tajne-heslo"
default tablespace usr_data
temporary tablespace tmp
profile default;

grant connect, resource to pznadmin;
```

```
create user ejb identified by "tajne-heslo"
default tablespace usr_data
temporary tablespace tmp
profile default;

grant connect, resource to ejb;

create user wcmdbadm identified by "tajne-heslo"
default tablespace usr_data
temporary tablespace tmp
profile default;

grant connect, resource to wcmdbadm;

create user feedback identified by "tajne-heslo"
default tablespace usr_data
temporary tablespace tmp
profile default;

grant connect, resource to feedback;
```

Pro bezproblémový import dat do databáze je nutné dát uživateli db2admin práva databázového administrátora příkazem:

```
grant dba to db2admin
```

a poté mu je opět vzít (z bezpečnostních důvodů) příkazem:

```
revoke dba from db2admin
```

4.1.2. Migrace – ZČU

Migraci dat agendy portálu provedeme tak, že nejprve provedeme její export, změním konfiguraci WebSphere Portal na databázový systém *Oracle*, do kterého ji nakonec importujeme. K tomu slouží skript `WPSconfig.sh` (v prostředí MS Windows `WPSconfig.bat`), který se nachází v podadresáři `config` instalace WebSphere Portal. V témže adresáři je i konfigurační soubor `wpsconfig.properties`.

Nejprve tedy spustíme export z databáze *Cloudscape* příkazem:

```
./WPSconfig.sh database-transfer-export
```

Poté je nutné provést změnu konfigurace nastavením relevantních parametrů v souboru `wpsconfig.properties`. Jsou to:

- Sekce *Database Properties*:
 - *DbType* – typ databáze, ve které je uložena agenda WebSphere Portal; nastavíme na `oracle`
 - *WpsDbName* – jméno databáze (Oracle SID); v našem případě `WPS50`
- Sekce *Database Transfer*:

Kapitola 4. Napojení na infrastrukturu

- `cswps.jdbc.schema.names` – seznam uživatelských účtů, které WebSphere Portal používá; nedoporučuje se měnit
- `DbDriver` – jméno Java třídy implementující JDBC driver; pro Oracle je to `oracle.jdbc.driver.OracleDriver`
- `DbDriverDs` – jméno Java třídy implementující datový JDBC zdroj v podobě sdíleného svazku spojení do databáze; pro Oracle je to `oracle.jdbc.pool.OracleConnectionPoolDataSource`
- `DbUrl` – JDBC URL – jednoznačný identifikátor databáze, podle kterého JDBC connection manager najde odpovídající JDBC driver a vytvoří spojení do databáze; nastaveno na hodnotu `jdbc:oracle:thin:@deucalion.zcu.cz:1526:WPS50`
- `DbUser` – uživatelský účet administrátora agendy portálu; nedoporučuje se měnit, ponechána hodnota `db2admin`
- `DbPassword` – heslo k uživatelskému účtu administrátora
- `DbLibrary` – cesta na implementaci JDBC driveru daného databázového serveru; nastaveno na adresář `/opt/WebSphere/~/PortalServer/shared/app/oracle/ojdbc.jar`. JDBC drivery k Oracle jsou volně dostupné na *Oracle Technology Network* – <http://otn.oracle.com>

• Sekce *WPCP Database Properties*:

- `WpcpDbName` – identifikátor databáze pro komponentu WebSphere Portal Content Publishing – *WPCP*; musí být stejný jako identifikátor (Oracle SID) specifikovaný v parametru `WpcpDbUrl`, nastaveno na `WPS50`
- `WpcpDbUser` – uživatelský účet, ve kterém se nachází báze dat komponenty WebSphere Portal Content Publishing; nastaveno na `wcmdbadm`
- `WpcpDbPassword` – heslo k předchozímu uživatelskému účtu
- `WpcpDbUrl` – JDBC URL báze dat *WPCP*; nastaveno na hodnotu `jdbc:oracle:thin:@deucalion.zcu.cz:1526:WPS50`
- `WpcpDbEjbPassword` – heslo k uživatelskému účtu `ejb`
- `WpcpDbPznadminPassword` – heslo k uživatelskému účtu `pznadmin`
- `FeedbackDbName` – identifikátor databáze *Feedback*, který musí být stejný jako *Oracle SID* v parametru `FeedbackDbUrl`; nastaveno na hodnotu `WPS50`
- `FeedbackDbUser` – uživatelský účet, ve kterém se nachází databáze *Feedback*; nastaveno na hodnotu `feedback`
- `FeedbackDbPassword` – heslo k předchozímu uživatelskému účtu
- `FeedbackDbUrl` – JDBC URL databáze *Feedback*; hodnota `jdbc:oracle:thin:@deucalion.zcu.cz:1526:WPS50`

• Sekce *WMM Properties*:

- `WmmDbName` – identifikátor databáze *WebSphere Member Manager* – *WMM*; musí být stejný jako identifikátor (Oracle SID) specifikovaný v parametru `WmmDbUrl`, nastaveno na `WPS50`

- *WmmDBUser* – uživatelský účet, ve kterém se nachází báze dat *WMM*
- *WmmDbPassword* – heslo k předchozímu uživatelskému účtu
- *WmmDbUrl* – JDBC URL databáze *WMM*; nastaveno na hodnotu `jdbc:oracle:thin:@deucalion.zcu.cz:1526:WPS50`

Novou konfiguraci je vhodné ověřit otestováním spojení do nové databáze pomocí příkazu:

```
./WPSconfig.sh validate-database-connection-wps
```

Po změně konfigurace následuje import dat do nové databáze příkazem:

```
./WPSconfig.sh database-transfer-import
```

V případě WebSphere Portal 4.2 bylo pro úspěšnou migraci nutno vyřešit několik technických problémů. Především migrační nástroj využívá vlastnosti rozhraní JDBC dostupné až v JDK verze 1.4, součástí runtime je však verze JDK 1.3.1. To se projevilo následující chybou během migrace:

```
Exception in thread "main" java.lang.NoClassDefFoundError:  
java/sql/Savepoint
```

Pro účel migrace byl proto vzat balík *java.sql* z verze JDK 1.4.2 firmy *Sun*, kterým byl nahrazen stejnojmenný balík v JDK 1.3.1 firmy *IBM*. Po úspěšné migraci byl zmiňovaný balík vrácen do původního stavu. Rovněž bylo zjištěno, že v konfiguraci nelze použít v JDBC URL ve specifikaci databáze formát ve stylu *Oracle TNS Names*, protože blíže neidentifikovaný shell skript správně nezpracuje JDBC URL. Chceme-li tedy pro datové zdroje portálu nakonfigurovat *Oracle shared* resp. *dedicated* server, musíme tak učinit v konfiguraci procesu *Oracle Listener*.

Tímto byla migrace agendy portálu úspěšně vykonána, nicméně poté nebylo možné se do WebSphere Portal přihlásit. Po výměně *Oracle* JDBC driveru za driver umožňující sledování komunikace s databázovým serverem byla chyba identifikována.

Při identifikaci této a později i jiných chyb byl použit následující postup:

Trasování JDBC

Na straně databázového serveru *Oracle* byl aktivován tzv. *tracing* nastavením parametrů `sql_trace` a `oracle_trace_enable` v souboru *initWPS50.ora* na hodnotu `true`. Pak jsou veškeré operace v databázi zaznamenávány do souborů `ora_nnnnn.trc`. Čitelnější podobu těchto záznamů získáme pomocí příkazu `tkprof` například takto:

```
tkprof ora_17573.trc ora_17573.txt
```

Na straně klienta, tedy WebSphere Portal, je možné sledovat operace prováděné v databázi pomocí zvláštních JDBC driverů dostupných na *Oracle Technology Network*. Jsou to `classess12.g.zip` pro JDK 1.2 a 1.3 nebo `ojdbc14.g.jar` pro JDK 1.4. Zaznamenávání prováděných operací do logovacího souboru aktivujeme nastavením dodatečných parametrů v následujících konfiguračních souborech aplikačního serveru a portálu:

```
<WAS_ROOT>/config/cells/portal/nodes/portal/servers/~  
server1/server.xml
```

```
<WAS_ROOT>/config/cells/portal/nodes/portal/servers/~  
WebSphere_Portal/server.xml
```

V těchto souborech přidáme následující řádky:

```
<jvmEntries ...>
...
<systemProperties xmi:id="Property_22"
name="oracle.jdbc.Trace" value="true"/>
<systemProperties xmi:id="Property_23"
name="oracle.jdbc.LogFile" value="/tmp/jdbc.trc"/>
<systemProperties xmi:id="Property_24"
name="oracle.jdbc.oraclelogCategoryMask" value="255"/>
<systemProperties xmi:id="Property_25"
name="oracle.jdbc.oraclelogPrintMask" value="255"/>
</jvmEntries>
```

Veškeré operace s databází pak JDBC driver zaznamenává do souboru /tmp/jdbc.trc.

Výše uvedený problém částečně vyřešil *WebSphere Portal Version 5.0 Interim Fix 1* označený jako *PQ77683* z 6.1.2004, který obsahoval opravu problému č.57287 (*After database transfer to Oracle 9i, cannot log in to WebSphere Portal*) K úplnému odstranění problému bylo nutno ještě změnit v souboru *LocalizerService.properties* nastavit parametr *locale.default.language* na hodnotu *en*. Pak již bylo možno se k portálu přihlásit. Nicméně implicitní nastavení portálu na anglický jazyk nebyl náš záměr a proto následoval prostřednictvím zaměstnance firmy *IBM* e-mail směřovaný do vývojových laboratoří s podrobným popisem chyby a jejího odstranění (viz strana 217):

Odpovědi se nám sice nedostalo, ale 20.1.2004 vyšel nový *WMM Cumulative iFix 2* pod číslem *PQ81181*, u kterého je v seznamu opravených chyb uvedeno i *Correct the TO_DATE function in Oracle database*. Po jeho aplikaci je již vše v pořádku.

Dalším problémem, který se objevil po migraci, byla chyba *ORA-00942: table or view does not exist*, která se začala objevovat v logu *WebSphere Portal*. Opět pomocí logování JDBC driveru bylo zjištěno, že je nutné, aby uživatel *ejb* dal uživateli *pznadmin* následující práva, což nebylo v dokumentaci k *WebSphere Portal* nikde zmíněno:

```
grant select, insert, update, delete on brbeans_rule
to pznadmin;
```

```
grant select, insert, update, delete on brbeans_rulefolder
to pznadmin;
```

Na závěr této podkapitoly ještě malá poznámka:

Vyplatí se databázi po úspěšné migraci a otestování plné funkčnosti portálu zazálohovat pomocí příkazu:

```
exp FILE=wps50-exp.dmp LOG=wps50-exp.log
OWNER=(DB2ADMIN,WPSDBUSR,WMMDBUSR,PZNADMIN,EJB,WCMDBADM,FEEDBACK)
GRANTS=Y
```

Potom ji v případě havárie či nezdařené manipulace můžeme snadno obnovit příkazem:

```
imp FILE=wps50-exp.dmp FULL=Y GRANTS=Y LOG=wps50-imp.log
```

4.1.3. Nastavení a Migrace do Oracle – OSU

Je nutno mít funkční instanci databáze Oracle; podporované verze jsou Oracle Enterprise Edition 8i Release 3 (8.1.7), Oracle Enterprise Edition 9i Release 2 (9.2.0.1). Pro verzi 9.2.0.1 je potřeba mít k JDBC driverům také OCI driver.

1. Vytvoříme databáze wps50, wpcp50, fdbk50 a do těchto databází vytvořit uživatele:

wpsdbusr: uživatel jádra, není potřeba jmenná konvence
wmdbusr: správce uživatelů, není potřeba jmenná konvence
pznadmin: uživatel správy dokumentů, potřeba jméno
EJB: uživatel správy dokumentů, potřeba jméno
wcmdbadm: uživatel správy dokumentů, není potřeba jmenná konvence
feedback: uživatel správy dokumentů, potřeba jméno

Struktura databází a uživatelů:

Databáze	Uživatel
wps50	wpsdbusr, wmdbusr
wpcp50	pznadmin, EJB, wcmdbadm
fdbk50	Feedback

2. Tvorba databází (wps50, wpcp50, fdbk50) pomocí oracle utility dbca; při tvorbě je nutné zadat kódování UTF-8 !!

Tvorba uživatelů – databáze WPS50

```
create user WPSDBUSR identified by PASSWORD default  
tablespace USERS temporary tablespace TEMP;
```

```
create user WMDBUSR identified by PASSWORD default  
tablespace USERS temporary tablespace TEMP;
```

Tvorba uživatelů – databáze WPCPC50

```
create user PZNADMIN identified by PASSWORD default  
tablespace USERS temporary tablespace TEMP;
```

```
create user EJB identified by PASSWORD default  
tablespace USERS temporary tablespace TEMP;
```

```
create user WCMDBADM identified by PASSWORD default  
tablespace USERS temporary tablespace TEMP;
```

Tvorba uživatelů – databáze FDBK50

```
create user FEEDBACK identified by PASSWORD default  
tablespace USERS temporary tablespace TEMP;
```

Přidělení rolí uživatelům – databáze WPS50

Kapitola 4. Napojení na infrastrukturu

```
grant connect, resource to WPSDBUSR;  
grant connect, resource to WMMDBUSR;
```

Přidělení rolí uživatelům – databáze WPCP50

```
grant connect, resource to PZNADMIN;  
grant connect, resource to EJB;  
grant connect, resource to WCMDBADM;  
grant insert any table to WCMDBADM;
```

Přidělení rolí uživatelům ? databáze FDBK50

```
grant connect, resource to FEEDBACK
```

3. Export dat a nastavení konfiguračních souborů pro Portál: Spustíme příkaz `<WPS_ROOT>/config/WPSconfig.sh~`
`database-transfer-export-linux`
4. Najdeme soubor `<WPS_ROOT>/config/wpconfig.properties` a v něm provedeme tyto úpravy:

```
DbSafeMode=false  
DbType= oracle  
WpsDbName=wps50  
DbDriver= oracle.jdbc.driver.OracleDriver  
DbDriverDs=  
    oracle.jdbc.pool.OracleConnectionPoolDataSource  
DbUrl= jdbc:oracle:thin:@cl106237.osu.cz:1521:wps50  
DbUser=wpsdbusr  
DbPassword=PASSWORD  
DbLibrary=/jdbc/lib/classes12.zip  
WpsDsName=wps50DS  
WpcpDbName=wpcp50  
WpcpDbUser=wcmbadm  
WpcpDbPassword=PASSWORD  
WpcpDbUrl=jdbc:oracle:thin:@cl106237.osu.cz:1521:wpcp50  
WpcpDbEjbPassword=PASSWORD  
WpcpDbPznadminPassword=PASSWORD  
FeedbackDbName=fdbk50  
FeedbackDbUser=FEEDBACK  
FeedbackDbPassword=PASSWORD  
FeedbackDbUrl=  
    jdbc:oracle:thin:@cl106237.osu.cz:1521:fdbk50  
WmmDsName=wmmDS  
WmmDbName=wps50  
WmmDbUser= wmmdbusr  
WmmDbPassword=PASSWORD  
WmmDbUrl= jdbc:oracle:thin:@cl106237.osu.cz:1521:wps50
```

5. Spustíme příkaz `<WPS_ROOT>/config/WPSconfig.sh~`
`database-transfer-import.`

Spustíme znovu server WebSphere.Portal a databáze by již měli být propojené.

V adresáři <WPS_ROOT>/config můžeme ověřit propojení s databázemi příkazy:

```
WPSconfig.sh validate-database-connection-wps
WPSconfig.sh validate-database-connection-wmm
WPSconfig.sh validate-database-connection-wpcp
```

4.2. Nastavení LDAP

Nastavit uživatelské repozitory WAS a WPS do adresáře LDAP jde udělat buď na začátku napojování na lokální infrastrukturu, nebo nejpozději v okamžiku zapínání tzv. *global security*, viz kapitola 4.4. Na ZČU byl použit první způsob, na OSU ten druhý.

4.2.1. OpenLDAP na ZČU

Postupujeme dle návodu „Configuring WebSphere Portal for SUN ONE“ z Infocentra.

V souboru <WPS_ROOT>/config/wpconfig.properties upravíme:

```
WEBSPHERE APPLICATION SERVER PROPERTIES
WasUserId=
    uid=wpsadmin,ou=users,ou=portal,ou=services,dc=zcu,dc=cz
WasPassword=*****

PORTAL CONFIG PROPERTIES
PortalAdminId=
    uid=wpsadmin,ou=users,ou=portal,ou=services,dc=zcu,dc=cz
PortalAdminPwd=was2003
PortalAdminGroupId=
    uid=wpsadmins,ou=groups,ou=portal,ou=services,dc=zcu,dc=cz

SECURITY LTPA AND SSO CONFIGURATION
LTPAPassword=***** (bug, nechceme, pro dalsi krok je potreba)
SSODomainName=zcu.cz (kdyby to bylo v~budoucnu potreba)

LDAP PROPERTIES CONFIGURATION
LookAside=true
LDAPHostName=amor.zcu.cz
LDAPPort=389 (potom bude 636)

LDAPAdminUIId=
    uid=wpsadmin,ou=users,ou=portal,ou=services,dc=zcu,dc=cz
LDAPAdminPwd=*****

LDAPServerType=IPLANET
LDAPBindID=
    uid=wpsadmin,ou=users,ou=portal,ou=services,dc=zcu,dc=cz
LDAPBindPassword=*****
```



```
ADVANCED LDAP CONFIGURATION
LDAPSuffix=ou=portal,ou=services,dc=zcu,dc=cz
LDAPUserSuffix=ou=users
```

```
LDAPsslEnabled=false (zatim)
```

Restartujeme oba servery pomocí příkazů `startServer.sh server1` a `stopServer.sh WebSphere_Portal`

a otestujeme výsledek:

```
WPSconfig.sh validate-ldap → BUILD SUCESSFUL
WPSconfig.sh enable-security-ldap → BUILD SUCESSFUL
```

Jsou-li v logu ORACLE exception, tak je potřeba nastavit dočasně oprávnění *dba* pro uživatele `db2admin`, `wpsdbusr`, `wmmdbusr`, `pnzadmin`, `ejb`, `wcmdbadm` a `feedback`.

Nyní by to už mělo fungovat, ale ve webové konzoli WAS ještě v sekci security → user → registry → LDAP změním `iPlanet` na `Netscape`.¹

4.2.2. Poznámky

- Do registru LDAP dejte na začátku jen několik osob a skupin. Bez aplikace dalších záplat má WPS problémy s objemným adresářem LDAP, což může způsobit zbytečné problémy při migraci.
- Byla zapnuta Global Security ve WAS, čili na webové admin konzoli nyní musíme zadávat login a heslo odpovídající LDAPu (`wpsadmin`).

4.2.3. Novell eDirectory na OSU

Na Ostravské univerzitě byl LDAP konfigurován až v průběhu zapínání security, viz sekce 4.8.

4.3. Generování SSL klíčů a IKEYMAN

Za účelem vytváření a správy kryptografických klíčů a certifikátů IBM dodává ke svým produktům program IKEYMAN, který lze přirovnat například k OpenSSL; je ovšem napsán v jazyce Java a má grafické uživatelské rozhraní.

Jak jsme po dlouhém experimentování zjistili, existuje několik verzí nástroje IKEYMAN, který je přibalen buď k WAS nebo k IHS a je bezpodmínečně nutné spouštět jej se správným JRE – tj. například IKEYMAN pro WAS musí používat JRE od WAS. Jinak nástroj buď nefunguje správně nebo díky různé podpoře GSS API neumí pracovat s potřebnými typy certifikátů a keystore.

¹`wpsconfig.properties` totiž nemá volbu `Netscape`.

4.4. Zapnutí Global Security

Zapnutí *Global Security* (chápejme zde jako terminus technicus) znamená, že WAS začne chránit své vstupní body hesly a transportní protokoly obalí SSL. Nejviditelnější změnou je možnost přistupovat k portálu přes HTTPS a převedení komunikace s LDAP serverem pod SSL (tj. místo portu 389 se bude používat 636).

To, že WAS poslouchá na SSL portech, si vynucuje následnou úpravu konfigurace IHS. Obvykle chceme, aby HTTPS port IHS byl namapován na HTTPS listener WAS/WPS. Protože IHS funguje jako aplikační proxy mezi klientem a WAS, lze mezi sebou protokoly HTTP a HTTPS různě převádět.

Nejprve zapneme SSL (listen port 443) na IHS a budeme jej mapovat na nezabezpečený HTTP port WAS. Následně zapneme security a tím dojde k převodu této komunikace (IHS–WAS) na HTTPS. Současně bude zabezpečeno i spojení WAS–LDAP.

Před zapnutím security je potřeba vygenerovat potřebné certifikáty a na správná místa importovat veřejné klíče certifikační autority.

4.5. Zapnutí HTTPS browser–IHS (ZČU)

Postupujeme podle návodu „Setting up SSL for WebSphere portal“ z info-centra. Akce má několik fází, které si postupně popíšeme:

4.5.1. Nastavení IHS

1. vygenerovat certifikát pomocí programu IKEYMAN
2. spustit IKEYMAN z <IHS_ROOT>/bin/ikeyman, použít JRE z WAS!!
export JAVA_HOME=/opt/WebSphere/AppServer/java/jre
3. new CMS key database file portal.kdb
4. případně konverze certifikátu mezi CA a IKEYMAN pomocí OpenSSL commandline tools
5. heslo k portal.kdb bude stashováno do portal.sth příkazem:
<IHS_ROOT>/bin/sslstash .ihskspw
6. zkontrolujeme nebo přidáme direktivu listen 443 v konfiguraci IHS a vytvoříme definici virtuálního webserveru pro SSL komunikaci:

```
<VirtualHost portal.zcu.cz:443>
LogLevel debug
ServerName portal.zcu.cz
ServerPath /opt/IBMHttpServer
DocumentRoot /opt/WWW-data
ErrorLog /opt/IBMHttpServer/logs/ssl-error_log
ServerAdmin valdman@civ.zcu.cz
Keyfile /opt/IBMHttpServer/conf/ssl3/portal.kdb
SSLV2Timeout 100
SSLV3Timeout 1000
```

```
#SSLServerCert portal.zcu.cz
SSLEnable
</VirtualHost>
```

4.5.2. WAS web admin

Další fáze spočívá v přidání HTTPS listeneru pro WAS a vygenerování nového pluginu pro IHS. Použijeme webovou administrační konzoli WAS:

1. Enviroment → Virtual Hosts → default_host → Host Aliases → new ("*",443)
2. Environment → Update WebServer Plugin (bude v adresáři <WAS_ROOT>/config/cells/plugin-cfg.xml)

4.5.3. httpd.conf

V konfiguraci IHS změním cestu v direktivě WebSpherePluginConfig do WAS, nebo lépe plugin-cfg.xml fyzicky zkopírujeme a restartujeme IHS:
<IHS_ROOT>/bin/apachectl restart

4.5.4. Přihlašování do WPS

V tuto chvíli upravujeme chování webové aplikace WPS nainstalované do WAS tak, aby byly login/logout mechanismy² přesměrovány na HTTPS.

Nejprve upravíme soubor <WPS_ROOT>/share/app/config/services/~> ConfigService.properties

```
redirect.login.ssl=true
host.port.https=443
```

Obdobně je vhodné zapnout redirect i pro logout. V souboru <WAS_ROOT>/installedApps/wps.ear/wps/war/WEB-INF/web.xml v části <security-constraint> změním původní hodnotu NONE na <transport-guarantee>CONFIDENTIAL</transport-guarantee> (pozor, ne CONFIDENTAL)

Poslední částí je přidání atributu ssl="true" do JSP tagů pro login obrázky všech motivů vzhledu:

V adresáři <WAS_ROOT>/installedApps/wps.ear/wps/war/themes hledat screen="Login" a v cca deseti souborech JSP přidat ssl="true" do v markupu HTML.

Ještě smažeme cache WAS, aby se změny projevily (viz odst.5.4.1. na str. 59):

```
rm -r <WAS_ROOT>/temp/portal/WebSphere_Portal/wps/wps.war/*
```

4.5.5. Finalizace

Nakonec je potřeba znovu vygenerovat plugin pro IHS, restartovat portál, a restartovat IHS.

²v tomto případě se fyzicky jedná o odklady generované v rámci HTML kódu

HTTPS mezi IHS a WAS na jednom stroji sice na první pohled není potřeba řešit, ale v praxi se ukázalo, že po zapnutí security některé komponenty WAS/WPS přepnou své listenery na HTTPS.

Postup konfigurace SSL mezi IHS a WAS je popsán v další části této kapitoly.

4.6. Zapnutí HTTPS browser-IHS (OSU)

IHS lze konfigurovat buď pomocí klasické editace konfiguračního souboru Apache `httpd.conf` (a případně dalších) nebo využít webové rozhraní administračního virtuálního serveru, který běží na portu 8888, a který umožňuje nakonfigurovat webserver podobně jako například opensource nástroj Commanche.

4.6.1. Nastavení IHS pomocí konfiguračního souboru

Postupujeme dle [7], appendix A, „Konfigurace HTTP serveru pro SSL“.

1. Nastavení jména serveru: zastavit HTTP server a potom v souboru `httpd.conf` nastavit `ServerName` `cl105175.osu.cz` (pro ostrý portál to bude `portal.osu.cz`)
2. Nastavení serveru pro SSL: Zastavit server, v `httpd.conf` zkontrolujeme řádek `Listen 80` (ujištění, že server poslouchá na portu 80) a `LoadModule` `ibm_ssl_module` `libexec/mod_ibm_ssl128.so`
3. Nastavení virtuálního serveru na portu 443: Na konec `httpd.conf` přidáme

```
<VirtualHost cl105175.osu.cz:443>  
    SSLEnable  
</VirtualHost>  
SSLDisable  
keyfile "/opt/IBMHttpServer/conf/keys/HttpKeyBase.kdb"  
SSLV2Timeout 100  
SSLV3Timeout 1000
```

a soubor uložíme.

4. Nástrojem IKEYMAN nyní vytvoříme certifikát pro IHS:
 - (a) nechat vytvořit nový certifikát
 - (b) vytvořit novou databázi klíčů pod názvem `HttpKeyBase.kdb`
 - (c) uložit do adresáře definovaného v `httpd.conf` – to znamená do `conf/keys`.
 - (d) zaškrtnout uložení hesla do souboru a platnost nenastavovat.
 - (e) uložit certifikát
5. Vytvoření self-signed certifikátu (pro skutečné použití není potřeba)
 - (a) popis klíče: `HttpSelfSignedCert` (můj název)

- (b) společné jméno `cl105175.osu.cz` (u portálu `portal.osu.cz`)
- (c) organizace nastavit na OSU
- (d) země na CZ
- (e) uložit a spustit IHS

Vyzkoušet, jestli jede připojení na portech 80 a 443. Možno též vyzkoušet verifikaci daných portů pomocí `netstat`, podrobnosti v [7].

4.6.2. Nastavení IHS pomocí konzole (doporučeno)

Ujistíme se, zda-li máme spuštěnou webovou administrační konzoli IHS:
`./bin/adminctl start`

Na konzoli se připojíme přes adresu `http://localhost` a vybereme volbu `configure server`. Heslo pro přístup administrátora je potřeba nejprve vytvořit pomocí `./htpasswd -c conf/admin.passwd <user>`

Vlastní nastavení SSL:

1. `basic settings`→`core settings` – ujistím se, zda-li je správné jméno serveru `cl106226.osu.cz`
2. `basic settings`→`advanced properties: specify additional ports...` Vybereme `Add` a vyplníme port 443→`apply`→`submit` (dejte si pozor na práva, musí být povolena přímo na serveru k souboru `httpd.conf`)
3. `basic settings` →`module sequences`→`add` – vybereme ze seznamu `ibm_ssl` (`IBMModuleSSL128.so`), ujistíme se, zda-li je vybrána správná knihovna, `apply`→`submit`
4. `configuration structure`→`create scope`→vybrat virtualhost ze seznamu, vyplnit úplný název virtual hosta (`cl106226.osu.cz`), port (443), opět název serveru (`cl106226.osu.cz`) a cestu k serveru (`/opt/IBMHttpServer/htdocs`)→`submit`
5. `security`→`server security: položka scope` musí mít hodnotu `global`, `enable ssl` dáme `yes`, do `Keyfile` dosadíme hodnotu `/opt/IBMHttpServer/conf/keys/HttpKeyBase.kdb` (data-báze klíčů pro HTTP server, vytvořeno předtím v `IKEYMAN`), vyplníme `timeouty` pro `SSLv2` (100) a pro `SSLv3` (1000)→`submit`
6. `security`→`host Authorization: scope` nastavené na virtualhosta `enable security yes`→`submit`

Restartujeme IHS a vyzkoušíme se připojit prohlížečem na adresu `https://localhost`

4.7. Zapnutí SSL WAS–LDAP a WAS–IHS (ZČU)

Postupujeme podle dokumentu „Setting Up LDAP over SSL with Sun One Server“ z infocentra. Podmínkou je funkční komunikace s LDAP serverem bez SSL.

Přihlásíme se na webovou konzoli WAS (`http://localhost:9090/admin` pro `server1` nebo `http://localhost:9091/admin` pro `WebSphere_Portal`) případně vypneme `Global security` (abychom mohli používat opět `wpsconfig.sh enable-security-ldap`).

Princip operace je následující: Při SSL handshake mezi LDAP a WAS použije LDAP server svůj certifikát, proto WAS potřebuje mít certifikát s veřejným klíčem příslušné CA (CA certificate chain), aby ověřil pravost LDAP serveru.

Akce má následující kroky:

1. zavést CA chain pro WAS
2. zavést CA chain pro WPS
3. upravit `wpsconfig.properties` a spustit `wpsconfig.sh`
4. oprava konfigurace WMM a restart

4.7.1. Registrace CA chain

Pomocí programu `<WAS_ROOT>/bin/ikeyman` připravíme keystore pro WAS `<WAS_ROOT>/etc/ZCUTrustFile.jks` a zaregistruje jej v admin konzoli WAS. Defaultní heslo pro trustfile je `WebAS`.

Podobně upravíme také keystore využívané portálovým JVM: `<WAS_ROOT>/java/jre/lib/security/cacerts` (nemá příponu `.jks`)

Poznámky:

Všechny keystore WAS jsou ve formátu JKS (Java Key Store). Privátní klíče pro server není nutné generovat znovu, protože se použije jen pro admin-konzoli a LDAP handshake – nejste-li paranoidní, můžete ponechat defaultní.

4.7.2. Úprava konfiguračních souborů

V souboru `<WPS_ROOT>/config/wpconfig.properties` nastavíme LDAP porty na 636 (místo 389) a `LDAPsslEnabled=true`.

Vyzkoušíme spojení na LDAP: `wpconfig.sh validate-ldap` (zkontroluje SSL spojení, ale nevyzkouší všechno, třeba nepozná chybějící trustfile)

Restartujeme `server1` (nyní vyžaduje login a heslo) a pokud je vše OK, tak v souboru `<WPS_ROOT>/shared/app/wmm/wmm.xml` přidáme na konec tagu `<ldapRepository name="wmmLDAP">` atribut `java.naming.security.protocol="ssl"`, aby zafungovalo JNDI.

Restartujeme server `WebSphere_Portal`.

Případné problémy viz logy `server1` a WPS + kontrola souboru `<WAS_ROOT>/config/cells/portal/security.xml`

4.8. Zapnutí SSL WAS–LDAP a WAS–IHS (OSU)

Čerpáme z dokumentace [5], kapitola 10.11.

V této sekci nastavujeme zabezpečené spojení mezi HTTP serverem a aplikačním serverem. Využijeme dříve vytvořený certifikát `HttpKeyBase.kdb`, vyexportujeme veřejný klíč, který vložíme do `WASWebContaineru`.

Dále vytvoříme certifikát na straně aplikačního serveru s názvem `WASWebContainer` a vyextrahujeme u obou klíč.

Extrahované klíče je potřeba navzájem prohodit. Otevřete soubor `HttpKeyBase.kdb` a přepnout se do *Signer certificates* a dejte *přidat*, vyberte `WASWebContainer.der` a ten dejte *vložit*. To samé udělejte u `WASWebContainer.jks`.

Pro komunikaci browseru s portálem je nutné vytvořit ještě klientský certifikát, pro vnější komunikaci. Vytvoříme tedy keystore `WASWebContainerClient.jks`, vytvoříme v něm self-signed certifikát a vyexportujeme jeho veřejný klíč, který vložíme do `WASWebContaineru`. Dále do něj naimportujeme veřejný klíč `WASWebContaineru`.

4.8.1. Úprava Web server plug in souboru

V adresáři `/opt/WebSphere/AppServer/config/cells` budeme editovat soubor `plugin-cfg.xml`

1. V sekci `<Transport Hostname="cl105175" Port="9443"~> Protocol="https">` je potřeba změnit defaultní nastavení k certifikátům a k uloženému heslu ve formě souboru *stashfile* (.sth)
2. Spustíme webovou administrátorskou konzoli WAS (server1) na portu 9090.
3. Zvolíme Security/SSL vytvoříme novou položku pomocí *new*, naspecifikujeme známé cesty k databázi klíčů (`WASWebContainer.jks`) a vyplníme příslušné hesla (nesplést).
4. Uložit změny; projeví se v souboru `cell/cl105175/security.xml`
5. Dále v menu `servers→server1→WebContainer→HTTP transports` tam vybereme port, který chceme provozovat na SSL. Musí být zaškrtnuto *Enable SSL* a zvolíme naše nastavení a místo toho defaultního.
6. V adresářích `<WAS_ROOT>/properties` a `<WAS-ROOT>/bin/~> wsinstance-propdefaults` jsou soubory `sas.client.props`, `sas.tools.properties`, `sas.stdclient.properties` a soubor `soap.client.properties`, ve kterých je nutno změnit odkazy na platné certifikáty – potřeba nastavit cesty k námi vytvořenému klientskému certifikátu `WASWebContainerClient.jks`.
7. Uložíme konfiguraci a restartujeme WAS.

Vyzkoušíme pomocí demo servletu Snoop na portu 9443 s HTTPS.

Dále se můžeme provést test s vynuceným ověřením klienta. Pomocí *ikeymanu* vyexportuje certifikát kompatibilní pro prohlížeč, který do něj naimportujeme.

4.8.2. Nastavení portálu pro LDAP SSL

Naimportování certifikátu z Novellu do souboru WASWebContainer (toto je pro WAS): Spustíme utilitu IKEYMAN aplikačního serveru, otevřeme svůj WASWebContainer a přidáme do certifikátů podepisujícího certifikát pro spojení s LDAP.

Naimportování LDAP certifikátu pro portál (WPS): Utilitou IKEYMAN soubor <WAS_ROOT>/java/jre/lib/security/cacerts otevřeme,³ a potom přidáme ten samý certifikát jako předchozím bodě u WAS.

Změna souboru wpconfig.properties:

```
WEBSPHERE APPLICATION SERVER PROPERTIES
WASuserid=uid=wpsbind,ou=users,dc=portal,o=ou
WASpassword=wpsbind
```

```
PORTAL CONFIGURATION PROPERTIES
PortalAdminid=uid=wpsadmin,ou=users,dc=portal,o=ou
PortalAdminidshort=wpsadmin
PortalAdminpwd=wpsadmin
PortalAdmingroupid=cn=wpsadmins,ou=groups,dc=portal,o=ou
PortalAdmingroupidshort=wpsadmins
```

```
SECURITY LTPA AND SSO CONFIGURATION
LTPApassword=wpsadmin
SSODomainName=.osu.cz
```

```
LDAP PROPERTIES CONFIGURATION
LDAPHostName=cl106237.osu.cz
LDAPPort=636
LDAPAdminuid=uid=wpsadmin,ou=users,dc=portal,o=ou
LDAPAdminpwd=wpsadmin
LDAPServerType=NDS
LDAPBindid=uid=wpsbind,ou=users,dc=portal,o=ou
LDAPBindPassword=wpsbind
```

```
ADVANCED LDAP CONFIGURATION
LDAPSuffix=dc=portal,o=ou
LDAPUserprefix=uid
LDAPUsersuffix=ou=users
LDAPgroupprefix=cn
LDAPgroupsuffix=ou=groups
LDAPuserobjectclass=inetOrgPerson
LDAPgroupobjectclass=groupOfNames
LDAPgroupmember=member
LDAPSSLenabled=true
```

```
CREDENTIALS FOR WAS ADMINISTRATION SECURE SOAP CONNECTION
TrustStore=/etc/WASWebContainer.jks
TrustStorepwd=portal123
KeyStore=/etc/WASWebContainer.jks
KeyStorepwd=portal123
```

³default heslo je changeit

Dále postupujeme:

1. spustit server1 a stopnout WebSphere_Portal
2. validovat nastavení LDAP pomocí `wpsconfig.sh validate_ldap` v adresáři `<WPS_ROOT>/config/`
3. v adresáři `<WPS_ROOT>/config/templates/wmm/` v souborech pro NDS připsat řádek `java.naming.security.protocol="SSL"`
4. aktivujeme SSL nastavení služby LDAP pomocí příkazu `<WPS_ROOT>/config/wpsconfig.sh enable_security_ldap`
5. restartovat server WebSphere_Portal i s udáním uživatele `wpsbind` a hesla
6. V souboru `<WPS_ROOT>/shared/app/wmm/wmm.xml` najdeme sekci *LDAP repository*, ve které pod `LDAPport=636` řádek vložíme další řádek `java.naming.security.protocol="SSL"`.

4.8.3. Nastavení portálu pro SSL provoz

Postup je stejný jako v sekci 4.5.4.

4.9. Některé problémy

Následující část popisuje několik problémů, jejichž vyřešení nám zabralo desítky hodin práce, a které v některých ohledech zdrželi projekt o celé měsíce. Vesměs se jedná o nedokumentované nebo špatně dokumentované záležitosti.

Většina problémů (chyb a nekompatibilit software), na které jsme narazili, se týkala napojení na LDAP server, potažmo komponenty WebSphere Member Manager – WMM. To platí pro Novell eDirectory na OSU i OpenLDAP na ZČU.

4.9.1. Novell a WPS5.0

Ostravská univerzita využívá systém Novell pro ukládání uživatelských kont, poštu a jiné. Struktura LDAP serveru byla již vytvořena a bylo by zbytečné začít provozovat další LDAP server a tvořit nové struktury pro portál.

Při zkušební instalaci WebSphere Portálu 4.2 se nám, i přes podporu tohoto LDAP serveru, nepodařilo připojit uživatelskou bázi k portálu, proto jsme čekali na uvolnění verze 5.0, která měla mít implementovanou kvalitnější konektivitu.

WebSphere Portál 5.0 přinesl řadu změn a spojení s eDirectory se po počátečních problémech podařilo.

Při navazování spojení je vhodné, nejdříve se snažit komunikovat s LDAP serverem nezabezpečeně (bez SSL), až po úspěšném navázání spojení přejít na zabezpečenou komunikaci.

4.9.1.1. Problém uniqueID

Hlavním problémem verze portálu 4.2 bylo mapování atributu *uid* (*uniqueID*). Naše verze Novellu považuje za své *uid* atribut *cn*. Přes atribut *uid* jsme nebyli schopni jakkoliv dohledat uživatele, tudíž i navázat spojení. S příchodem další verze portálu jsme za pomoci konfiguračního skriptu (`wpconfig.properties`) nastavili tento název atributu na *cn* a tím se nám podařilo spojení navázat. Úzce spojeny s tímto problémem jsou vyhledávací filtry, které jsou popsány v dalším bodě.

4.9.1.2. Problém vyhledávacích filtrů

K vyhledávání uživatelů (skupin uživatelů) v databázi slouží vyhledávací filtry. Tyto filtry byly původně (v5.0) nastavitelné pouze za pomoci správcovské konzole aplikačního serveru. Po nainstalování opravných balíčků byly tyto filtry zahrnuty do souboru nastavení `wpconfig.properties`. Implicitní nastavení těchto filtrů bylo s ohledem na typ LDAP serveru. Bohužel u Novell eDirectory Serveru (NDS) byly filtry nastaveny k vyhledávání za pomoci atributu *uid*. Bylo proto nutné tyto nastavení změnit na *cn*.

4.9.1.3. Problém nastavení WMM

Aby úspěšně proběhla úloha `WPSconfig.sh enable-security-ldap`, bylo nutné změnit následující konfigurační soubory WebSphere Portal: `wmm_LDAP.xml.NDS.1.wmm` a `wmm_LDAP.xml.3.wmm` v adresáři `<WPS_ROOT>/config/templates/wmm`.

Tyto soubory musí pro úspěšné provedení úlohy obsahovat řádek (v sekci `LdapRepository`):

```
java.naming.security.protocol="SSL"
```

4.9.1.4. Výkonový problém s OpenLDAP

Na ZČU jsme téměř celý rok 2004 zápasili s problémem, jak přinutit WebSphere Portal efektivně pracovat s LDAP. WebSphere Portal totiž velmi neefektivně zjišťoval členství uživatelů ve skupinách, což omezovalo velikost báze uživatelů cca na 1000 až 2000 osob v několik málo desítkách skupin. Při překročení tohoto množství docházelo k timeout výjimkám během pokusů o přiřazování přístupových práv a tedy i nemožnosti jejich nastavení.

Problém jsme se snažili řešit zapnutím atributu *memberOf* v LDAP i WebSphere Portal, ale ukázalo se, že jej portál nerespektuje. Na podzim 2004 jsme se bezúspěšně snažili sehnat nějakou firmu, která by uměla problém řešit.

Spásou byl pro nás až kumulativní fix PQ98004 z 6.1.2005, který řeší všechny známe problémy WMM. Poté zafungoval atribut *memberOf* a vše již je jak má být.

4.9.2. Problém komunikace SSL

Další třídou problémů představuje zapnutí Security, které v rámci aplikačního serveru provede množství zásadních změn.

4.9.2.1. Zapnutí SSL pro LDAP

Pro úspěšnou komunikaci s LDAP serverem přes SSL je nutné naimportovat správné certifikáty do dvou portálových úložišť klíčů. První je úložiště aplikačního serveru a druhým úložiště Javy (`cacerts`). NDS disponuje se dvěma typy certifikátů, je nutné vybrat ten, podle něž se provádí ověřování. Jeho veřejný klíč pak přidat do výše uvedených souborů za pomoci nástroje `IKEYMAN`.

Před vlastním spuštěním úlohy `WPSconfig.sh enable-security-ldap` je užitečné spustit úlohu `WPSconfig.sh validate-ldap`. Dojde k ověření spojení s LDAP serverem.

4.9.2.2. Portal Document Manager

Komponentu Portal Document Manager se nepodařilo na ZČU uchodit pro HTTPS klienty. Přestože si myslíme, že jsou všechny potřebné certifikáty jsou na svých místech, portál PDM nemůže komunikovat s WPCP. Strávili jsme s tím dost času, ale ani velmi podrobné logování nám nepomohlo.

Mezitím přišla zpráva z OSU, že PDM výkonově nevyhovuje a oznámení o zcela přepracovaném PDM ve verzi 5.1, takže jsme do řešení tohoto problému více úsilí neinvestovali.

4.9.3. Problémy s Oracle databází

4.9.3.1. Nastavení locale

Portál po migraci do Oracle 9.i nerespektoval české nastavení prostředí databáze. Částečně pomohl fix `PQ777683` z 6.1.2004, ale plným řešením byl až kumulativní fix z 20.1.2004.

4.9.3.2. Uživatel `db2admin`

Přestože lze na několika konfiguračních místech zadat jména databázových schémat, které mají různé komponenty portálu využívat, ne vždy to funguje. Některé komponenty se tvrdohlavě připojují jako uživatel `db2admin`.

První věc, kterou je potřeba zkontrolovat při migraci nebo rozchozování další komponenty (WPCP, personalization, ...), je právě kontrola používaného schématu a dostatečných oprávnění (obvykle `dba`).

Správa WebSphere Portal

V této kapitole jsme shrnuli některé základní oblasti správy WebSphere Portal.

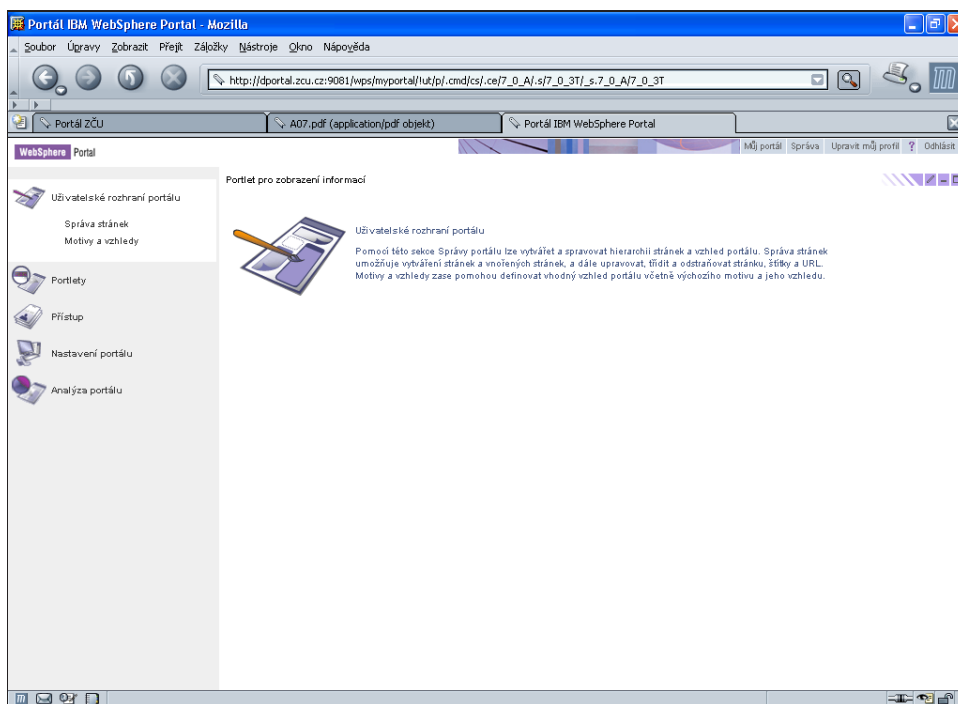
Obsah

5.1. Administrační rozhraní WPS	52
5.2. Konfigurační soubory WPS	52
5.3. Autorizace	53
5.3.1. Prostředky chráněné WPS	53
5.3.2. Role Based Access Control	54
5.3.3. Dědění oprávnění	57
5.3.4. Jak nastavovat přístupová práva	57
5.3.5. Nastavení práv – příklad ZČU	58
5.3.6. Jak delegovat řízení přístupových práv	58
5.3.7. Doporučení	58
5.4. Branding portálu	59
5.4.1. Jak na to	59
5.4.2. Tipy pro zrychlení práce	60
5.5. Vypnutí samoobslužných mechanismů	61
5.6. Mapování URL	62
5.7. Konfigurační rozhraní XML	62

Po úspěšné instalaci portálu a napojení na infrastrukturu stojí jeho administrátor před úkolem, jak zařídit, aby portál správně vypadal a měl všechny požadované funkce (nyní neuvažujeme vývoj vlastních aplikací).

Cílem této kapitoly není nahrazovat manuál k portálu, ale poskytnout čtenáři základní orientaci v problematice a nasměrovat jej na příslušnou dokumentaci, která je bohužel rozdělena do několika částí: Infocenter WebSphere Portal a WebSphere Application Server, RedBooks, články na DeveloperWorks PortalZone, on-line nápověda, RedPapers, nepublikované interní dokumenty a whitepapers, „oficiální“ knihy, ... Základní ovládání portálu a vytváření stránek je poměrně dobře dokumentováno, doporučujeme nastudovat infocenter, [9] a [12].

Jako vhodné čtení doporučujeme [1, 12, 13] nebo [10], případně i [9]. Je-li administrátor konfrontován s konkrétním problémem, je nejlepší hledat nějakou prezentaci z IBM konferencí; ostatní dokumentace je poměrně povrchní: nejde do hloubky a nevysvětluje vnitřní princip fungování – obvykle se omezuje jen na popis klikání myši v uživatelském rozhraní.



Obrázek 5.1: Webové administrační rozhraní portálu

5.1. Administrační rozhraní WPS

Portál má mnoho úrovní konfigurace, ze kterých je pouze část dokumentována. Naštěstí se většina rutinní správy dělá pomocí webového administračního rozhraní (viz obrázek 5.1), nicméně část postinstalačních úkonů spočívá v úpravě podivných parametrů v nedokumentovaných XML souborech dislokovaných v desítkách adresářů.

Pro opravdové „hardcore“ administrátory je zde ještě konfigurační rozhraní XML, které zmiňujeme na straně 62.

5.2. Konfigurační soubory WPS

Portál jako Java enterprise aplikace a soubor dalších modulů trpí všemi neduhy J2EE architektury – konfigurační direktivy jsou roztroušeny po stovkách souborů (různých *properties*a a *deployment deskriptorů*), dle jednotlivých komponent/modulů a jejich začlenění do architektury WAS/WPS. Nejčastější umístění je:

- `<WAS_ROOT>/...` pokud se jedná o modul WPS, který je samostatnou J2EE aplikací
- `<WAS_ROOT>/config/cells/node/applications/wps.ear/...` pokud se jedná o nastavení WPS jako J2EE aplikace, resp. využívající standardní konfigurační mechanismy J2EE
- `<WAS_ROOT>/installedApps/portal/wps.ear/wps.war/...` – webová část J2EE aplikace WPS: JSP, obrázky..., motivy a skiny vzhledu

- `<WPS_ROOT>/share/app/config`, pokud se jedná o modul WPS využívající vlastní konfigurační mechanismy WPS
- `<WPS_ROOT>/installedApps/portletX.ear/portletX.war/` – webová část nainstalovaných portletů, resp. jim odpovídajících portletových aplikací. Mnoho standardních portletů (Fileserver, BannerAdd, prohlížeče dokumentů,..) mají právě sem nastaven kontext.¹
- *PortletSettings* (`portlet.xml`) jednotlivých portletů
- Deployment deskriptory (`web.xml`) jednotlivých komponent WAS a WPS
- `<WPS_ROOT>/config/wpsconfig.properties` – nastavení infrastruktury WAS a WPS (databáze, LDAP, ...)
- `<WPS_ROOT>/share/app/wmm` a `WPS_ROOT/wmm/` nastavení WebSphere Member Manager (komponenta, pomocí které WPS přistupuje k LDAP; využívá ji i Portal User Management (PUMA) a další komponenty, viz strana 23)
- mnoho dalších míst, obvykle nedokumentovaných...

Podrobný popis struktury instalačního adresáře WebSphere Portal lze nélt v Infocentru.

Jedním z nejdůležitějších konfiguračních souborů portálu je soubor `<WPS_ROOT>/share/app/config/services/ConfigService.properties`, ve kterém je množství parametrů ovlivňujících uživatelské rozhraní portálu. Je zde například přesměrování po login/logout (užitečné pro nastavení SSL), styl generování navigační struktury a podobně.

5.3. Autorizace

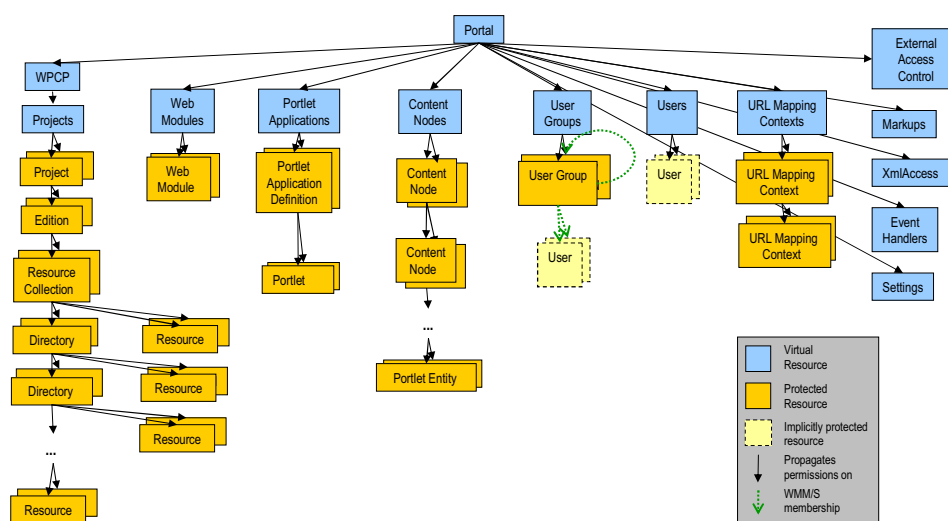
WebSphere Portal je z pohledu aplikačního serveru jedna z mnoha webových aplikací, k níž WAS řídí přístup prostředky J2EE – takto je například vyžádaná autentizace, pokud neznámý uživatel chce přistupovat ke chráněnému prostředku.

Z hlediska WAS má portál chráněné a nechráněné prostředky, k nechráněným se přistupuje přes URL `http://portal:9081/wps/portal`, k těm chráněným přes `http://portal:9081/wps/myportal`. Všimněte si, že i chráněné prostředky jsou přístupné přes protokol HTTP. Chceme-li používat protokol HTTPS, je potřeba příslušně nakonfigurovat WAS a IHS, viz kapitola 4.

5.3.1. Prostředky chráněné WPS

Obecně platí, že portál chrání své „portálové“ prostředky sám, tj. řídí k nim přístup. Takových prostředků je celá řada, viz. obrázek 5.2. Nás ale budou nejvíce zajímat následující prostředky:

¹Aby se administrátor z toho nezbláznil, je vhodné udělat v souborovém systému link na nějaký adresář, kde budou umístěné soubory pro tyto portlety...



Obrázek 5.2: Hierarchie prostředků chráněných portálem

- portálová stránka
- portálová aplikace
- portlet
- uživatel a skupina uživatelů²

Prostředky mohou být navíc veřejné nebo privátní. Privátní prostředky patří uživateli, který je vytvořil, a ani administrátor k nim nemá přístup.³

Privátní prostředky jsou teoreticky mazány při zániku jejich vlastníka, prakticky ale WebSphere Portal ve verzi 5.0 neobsahuje žádnou takovou úklidovou proceduru.⁴ Dále si myslíme, že privátní prostředky se smažou v okamžiku odstranění veřejného prostředku, od kterého byly odvozeny.

Každý prostředek má svého vlastníka.

Kromě výše uvedených prostředků existují ještě tzv. virtuální prostředky, které zastřešují všechny konkrétní prostředky jednoho typu, například všechny uživatele nebo všechny portlety. Všechny prostředky zahrnuje virtuální prostředek *PORTAL*.⁵

5.3.2. Role Based Access Control

Portál používá k přehlednému řízení přístupu systém rolí, které jsou abstrakcí zastřešující konkrétní oprávnění na konkrétní operace nad nějakými prostředky (viz obrázek 5.3).

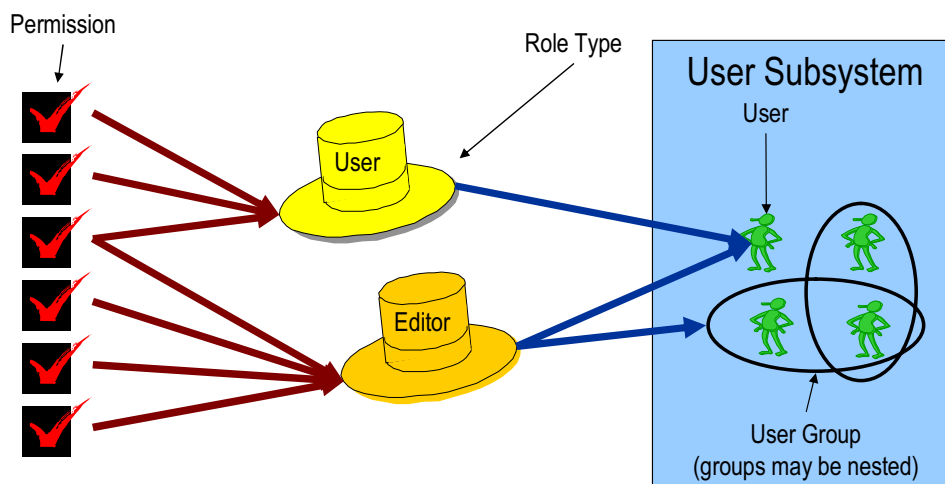
Na tomto místě je vhodné udělat malou odbočku a definovat termíny vztahující se k řízení práv ve WebSphere Portal:

²pouze v případě, že budeme chtít pomocí portálu spravovat uživatele v repozitáři LDAP serveru

³Samozřejmě se k nim nakonec dostane, ale pouze v podobě XML vyexportovaného nástrojem `xmlaccess`.

⁴Naopak se ukazuje, že pokud uživatel „zmizí“ z Novell eDirectory, zůstávají v portálu jeho přístupová oprávnění, která nelze odstranit bez opětovného zavedení uživatele do LDAP.

⁵Skupina administrátorů portálu je tedy v roli `administrátor@PORTAL`.



Obrázek 5.3: Definice role (vlevo) a její přiřazení (vpravo)

resource entita, ke které portál řídí přístup (access control), nejčastěji stránka nebo portlet

principal entita v rámci portálu, které může být umožněn přístup k prostředku, například uživatel nebo skupina uživatelů

permission je definováno jako konkrétní akce nad konkrétním prostředkem (zobrazit stránku X)

role type typ role definuje několik způsobů interakce s prostředky

role sada oprávnění (permission) nad konkrétním prostředkem, například (editor@stránka X)

role assignment mapování mezi principalem a rolí; dojde k udělení všech oprávnění spojených s rolí k danému prostředku

Role je tedy soubor oprávnění nad prostředkem a přiřazují se do ní jednotliví uživatelé nebo celé skupiny uživatelů. Nyní se podíváme podrobně na role, které jsou stejné pro všechny typy prostředků, a mají podobnou sémantiku:⁶

user role uživatel představuje právo zobrazit prostředek (tj. read-only přístup); není-li uživatel alespoň v této roli, je pro něj prostředek skrytý (tj. jako by neexistoval)

privileged user privilegovaný uživatel má právo vytvářet si vlastní privátní kopie veřejných prostředků, které si potom může upravovat; podobně viz customizace na straně 75.

V případě stránky to znamená možnost její duplikace a personalizace (na stránce se objeví odkazy upravit stránku, přidat stránku), v případě portletu to znamená možnost vstoupit do editačního⁷ režimu (ikona „tužky“ nebo „šroubováku“).

⁶ta se samozřejmě mírně liší v závislosti na typu prostředku – být editorem portletu znamená něco jiného, než být editorem stránky...

⁷poněkud nešťastně zvolený termín, lepší by bylo „kustomizačního“

Z pohledu programátora tato role umožňuje vytvářet kopie objektu PortletData.

editor editor může vytvářet a upravovat veřejné prostředky, mazat může jen ty, kterých je vlastníkem

V případě stránek to znamená měnit atributy stránky a přidávat na ní portlety, v případě portletů to znamená přístup do konfiguračního režimu (ikona „klíče na matky“).

manager manager může vytvářet, upravovat a mazat veřejné prostředky

delegator delegátoři mohou předávat svá oprávnění dalším principalům

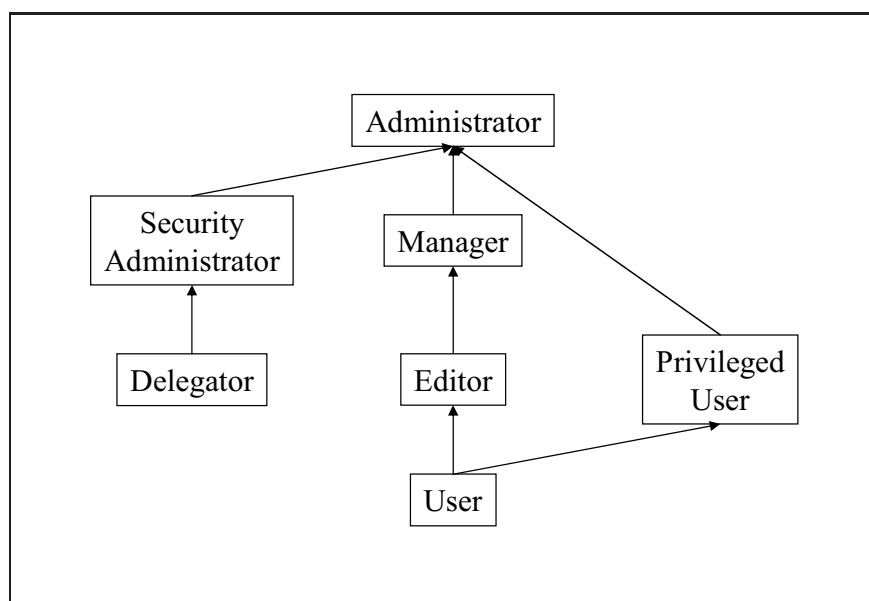
Tuto roli budete zřejmě používat ve spojení se skupinami uživatelů, abyste mohli omezit, komu lze předat vlastní oprávnění.

security administrator Bezpečnostní administrátor může řídit přístupová práva k prostředku, tj. přiřazovat principaly do rolí; k tomu musí být zároveň alespoň delegátorem principalu, který chce přiřadit

Tato role neumožňuje měnit nebo mazat prostředek, ke kterému je přiřazena – to může dělat jen editor nebo manažer.

administrátor Administrátor může dělat všechno, je superpozicí výše uvedených rolí. Dědění a propagace této role nelze zakázat.

Vztahy mezi rolmi ukazuje orientovaný graf na obrázku 5.4. Z něj je patrné, která role je nadmnožinou ostatních. Je tedy například vidět, že role



Obrázek 5.4: Vazby mezi rolmi

editor není nadřazená roli privilegovaný uživatel. V praxi to znamená například to, že editor portletu nemůže vstoupit do editačního režimu⁸ portletu, není-li zároveň jeho privilegovaným uživatelem.

Zda-li to bude vadit záleží na návrhu portletu (ve kterém režimu se pracuje s PortletData, a ve kterém s PortletSettings), ale obvykle to vadit bude.

⁸zde opět narážíme na nevhodné pojmenování

Je-li uživatel současně v roli editor i privilegovaný uživatel, převáží role editor; uživatel tedy bude pracovat s veřejným prostředkem.

5.3.3. Dědění oprávnění

Přiřazení práva funguje podobně jako na souborových systémech, které také používají několik nezávislých stromů oprávnění (read, write, erase, access control,...). Akt přiřazení uživatele do role tedy znamená udělení oprávnění do konkrétního místa stromu prostředku.

U každého prostředku lze určit, která práva se mají šířit na potomky (propagation) a která lze oddělit od rodiče. V terminologii WebSphere Portal se tomu říká *propagation blocks*.

Kromě dědění práv (u prostředků stejného typu – stránek) se oprávnění přenáší i mezi různými typy prostředků, například od portletové aplikace na portlety.

5.3.4. Jak nastavovat přístupová práva

Rízení práv se dělá na úrovni prostředku – nejčastěji portlet nebo stránka. WebSphere Portal verze 5 nabízí pohled přes prostředky (Administration→Access→Resource Permissions), přes uživatele nebo skupiny (Administration→Access→User and Group Permissions) i kontextově nastavení pomocí odkazu „Přiřadit práva“ na portálové stránce. Nejpraktičtější je asi kontextový způsob.

V portálu existuje fiktivní *anonymous portal user* a fiktivní skupina *all authenticated users*, pomocí kterých lze dávat oprávnění anonymnímu (tj. nepřihlášenému) uživateli nebo všem přihlášeným uživatelům. Tyto objekty jsou ale disjunktní: má-li být prostředek přístupný všem přihlášeným i nepřihlášeným uživatelům, je potřeba nastavit práva buď pro *all portal users and groups* nebo pro *anonymous portal user* a současně i pro *all authenticated users*.

U portletů je potřeba dobře nejprve zjistit, jak jsou data sdílena mezi instancemi portletů, resp. v jakém režimu se pracuje s PortletData a v jakém s PortletSettings. Jednotlivé portlety se chovají různě, někdy jsou editační i konfigurační režimy téměř totožné, jindy bývá konfigurační režim nadřazen editačnímu, ve kterém se jen doladí detaily.

PortletSettings (obvykle nastavované v konfiguračním režimu) jsou sdílená všemi instancemi portletů. Je-li požadováno více variant nastavení v PortletSettings, nezbyvá než duplikovat portlet nebo celou portletovou aplikaci (to standardně může jen administrátor portálu). Takové portlety jsou potom na sobě zcela nezávislé. Nezapomeňte je nějak rozumně pojmenovat, t.j. nastavit unikátní titulek pro češtinu a angličtinu, jinak je od sebe nepoznáte.

V konfiguračním režimu lze Pracovat s PortletSettings, ale pouze s těmi položkami, které mají zobrazení v HTML formuláři. Programátor může používat další konfigurační proměnné uložené v souboru `portlet.xml`, čímž dosáhne jejich skrytí před editory a managery portletu. Nastavení v `portlet.xml` může standardně dělat pouze administrátor portálu pomocí stránek a portletů pro správu portálu.

5.3.5. Nastavení práv – příklad ZČU

Portál ZČU je nastaven tak, že jsou přiřazena práva na hlavní stránky (záložky na červené liště), je zakázáno jejich dědění a naopak povolena jejich propagace.

Do kořene stránek *My portal* jsou ponechána defaultní práva včetně jejich propagace, protože přesně nevíme, co by jejich přenastavení mohlo způsobit.

Práva ke stránkách nastavujeme pomocí role *user* pro skupiny *all authenticated users*, *students*, *staff*, případně uživatele *anonymous portal user*.

Právo *privileged user* pro *all authenticated users* na stránku *Moje stránka* v sekci *Welcome*. U jiných stránek zatím customizaci neplánujeme.

U portletů jsou práva nastavena dle potřeby: právo *user* těm, kteří potřebují přístup do *view* režimu, právo *privileged user* těm, kteří potřebují editační režim, a právo *editor* těm, kteří potřebují konfigurační režim.

Je-li potřeba pro jeden portlet nastavit na různých portálových stránkách různá práva, je zkopírován a varianty označené v jeho titulku (např. *veřejný*, *public*, *shared*, *jen přihlášení*, *CIV...*).

Nastavení práv pro jiné druhy prostředků než stránky a portlety jsme nedělali. Správa uživatelů bude mimo portál.

5.3.6. Jak delegovat řízení přístupových práv

Aby se zobrazil odkaz „Přiradit práva“, musí mít uživatel *mj.* právo *user* na stránku *Content Root*→*Administration*→*Access*.

Nastavte roli *security administrator* nad prostředkem, ke kterému chceme řídit práva.

Nastavte roli *delegator* nad tou skupinou/uživatelem *komu lze přidávat práva*.

My jsme dali skupině zaměstnanců (*staff*) právo *delegator@USER GROUPS* (virtuální prostředek) a *delegator@USERS* (rovněž virtuální prostředek). Dále jsme skupině *staff* dali právo *user@access* (to je stránka v části *administration*), díky čemuž se zobrazí odkaz „Přiradit práva“. Oprávnění k řízení přístupu potom řídíme pouze zařazením uživatele do role *security administrator@prostředek*.

5.3.7. Doporučení

Obecně je dobré snažit se řídit přístup pomocí oprávnění ke stránkám, než k portletům. U stránek se *apriori* počítá, že ke každé stránce budete individuálně nastavovat práva, kdežto u portletů jejich vynucená duplikace vede na nepřehlednou situaci.

WebSphere Portal ve verzi 5.0 bohužel nemá nástroj na celkový report nastavení práv; v případě problémů je potřeba se proklikávat nastavením a dohledávat chyby. Pomocí administrační konzole *WAS* lze zapnout logování autorizačních rozhodnutí portálu, ale moc užitečné to není, protože log je rozsáhlý a nepřehledný.

- dávejte práva raději skupinám než konkrétním uživatelům
- používejte raději kombinaci rolí security administrator + manager než administrator (rolí manager lze blokovat)
- mají-li mít manažeři právo delegovat dále svá práva, potřebují k tomu oprávnění security administrator
- aby mohl uživatel předat všechna svá oprávnění, dejte mu roli security administrator na všechny virtuální prostředky kromě PORTAL
- pokud nechcete využívat roli delegator, dejte *all authenticated uses* roli delegator@PORTAL

Podrobné informace k této problematice lze najít téměř výhradně jen v materiálech z konferencí IBM, například [22].

5.4. Branding portálu

Termínem *portal branding* je označováno vizuální přizpůsobení WebSphere Portal firemnímu vzhledu (corporate identity). Po instalaci portálu je k dispozici několik motivů vzhledu (podnikové, věda, technika, finance), které jsou tvořeny JSP stránkami se speciálními tagy portálu a souborem grafických prvků.

Pro portlety jsou definovány podobným způsobem tzv. skiny, které tvoří dekoraci kolem vlastního obsahu portletu a zahrnují též ovládací tlačítka.

Témata i skiny lze snadno předefinovat nebo vytvořit zcela nové. Díky tomu, že portál umí pomocí JSP tagu `<wps:FindInTheme>` vyhledat „správný“ soubor v hierarchii adresářů, lze velkou část prvků opětovně použít a předefinovat jen požadovanou část.



Obrázek 5.5: Základní grafické schéma portálu ZČU



Obrázek 5.6: Základní grafické schéma portálu OSU

5.4.1. Jak na to

Branding portálu se dělá pomocí editace JSP stránek jednotlivých motivů vzhledu (themes), změnou CSS stylů a vytvořením nových grafických prvků. Postup je popsán například v [31].

HTML markup portálové stránky je rozdělen do několika částí, takže jeho přetvoření není složité. Problémy tvoří spíše technické aspekty editace JSP a CSS: portál vše ukládá do cache, kterou je potřeba neustále ručně mazat, nebo je nutné použít jiný přístup.

Témata vzhledu jsou uložena jako soubory CSS, JSP a GIF nebo JPG v podadresářích `<WAS_ROOT>/installedApps/wps.ear/wps.war/`. Portál obsahuje mechanismus vyhledání „nejlepšího“ souboru v rámci motivu, přihlíží se k jazyku a typu prohlížeče (viz JSP tag z portálové TDL `<wps:findInTheme>`).

Asi nejrychlejší postup pro první fázi práce je uložit si vhodnou portálovou stránku jako *úplnou stránku HTML* a změny provádět lokálně. HTML markup, resp. jeho ASCII reprezentace není příliš přehledná, proto je téměř nezbytné použít dobrý HTML editor (např. Macromedia Dreamweaver). Základní CSS definuje několik stovek různých selektorů, proto pokud chcete téma vzhledu výrazně „přebarvit“, je dobré použít nějaký profesionální nástroj jako například TopStyle3.

Dále jsme pozorovali, že WebSphere Portal má problémy s novějšími standardy JPEG, proto je potřeba například v Adobe Photoshop zásadně používat příkaz *Save for web*, což vyrobí soubor s minimem metainformací a volitelných rozšíření.

Jestliže vytváříme nový motiv vzhledu, je potřeba jej také zaregistrovat pomocí webového administračního rozhraní WebSphere Portal.



Obrázek 5.7: Základní skiny portletů používané v portálu ZČU

5.4.2. Tipy pro zrychlení práce

Portal server poskytuje maximum dat ze své cache; aby se projevovaly změny, je proto potřeba tento mechanismus obejít:

- Dále je potřeba nastavit parametr `reloadingEnable=true` v souboru `<WAS_ROOT>/config/cells/node/applications/wps.ear/deployments/wps/wps.war/WEB-INF/ibm-web-ext.xmi`, což by mělo vyřadit používání cache v situaci, kdy prohlížeč dělá reload stránky. Jedná se o nedokumentovaný postup.
- Udělat `touch default.jsp` v případě motivu a `touch control.jsp` v případě skinu, aby se vynutilo jejich překompilování (a navíc musí být vypnuta paměťová i disková cache, viz výše).

- Smazat diskovou cache <WAS_ROOT>/temp a <WAS_ROOT>/cache. Nepomůže, pokud už jsou data nakešována v RAM.

Soubor `Styles.css` si stahuje prohlížeč přímo a jde tedy s výhodou editovat lokálně (v situaci, kdy není nutné zasahovat do JSP). Při editaci na serveru dejte pozor na to, abyste modifikovali správně soubory vzhledem k locale, typu prohlížeče atd. Velkým pomocníkem je Mozilla/Firefox resp. jejich plugin Webdeveloper.

Podrobné informace a postupy viz prezentace [31] nebo na webu [33, 34].

5.5. Vypnutí samoobslužných mechanismů

Po instalaci má portál nastaveno několik mechanismů, které slouží k automatizaci správy uživatelů:

- samoobslužná registrace nových uživatelů (*self sign-in*)
- „zapomněl(a) jsem heslo“
- „upravit můj profil“ (jen přihlášení uživatelé)
- změna hesla (prostřednictvím správy uživatelského profilu)

Tyto mechanismy zřejmě nebude chtít každý používat, proto si ukážeme způsob jejich vypnutí na příkladě odstranění položky *Sign in* ze vstupní stránky portálu.

1. V adresáři <WAS_ROOT>/installedApps/portal/wps.ear/wps.war~/Themes/html nalezneme soubor `ToolBarInclude.jsp`.

2. nalezneme sekci *enroll buton*, která vypadá následovně:

```
<%-- enroll button --%>
<wps:if loggedIn="no">
<%
String dt =
com.ibm.wps.puma.UserManager.instance().getDirectoryType();
if (dt==null) { dt = ""; }
if (!dt.equals("SSPM")){
%>
<td class="wpsToolBar" valign="middle"
align="<%=bidiAlignRight%>" nowrap>
<a class="wpsToolBarLink" href='<wps:url
command="PrepareEnrollment" home="public" reqid="no"/>'>
<wps:text key="link.enrollment" bundle="nls.engine"/>
</a>
</td>
<%
}
%>
</wps:if>
```

3. Následně celou sekci zakomentujeme, jedná se totiž o tlačítko, které umožní vstup na obrazovku pro zápis do LDAP registru a to nechceme.

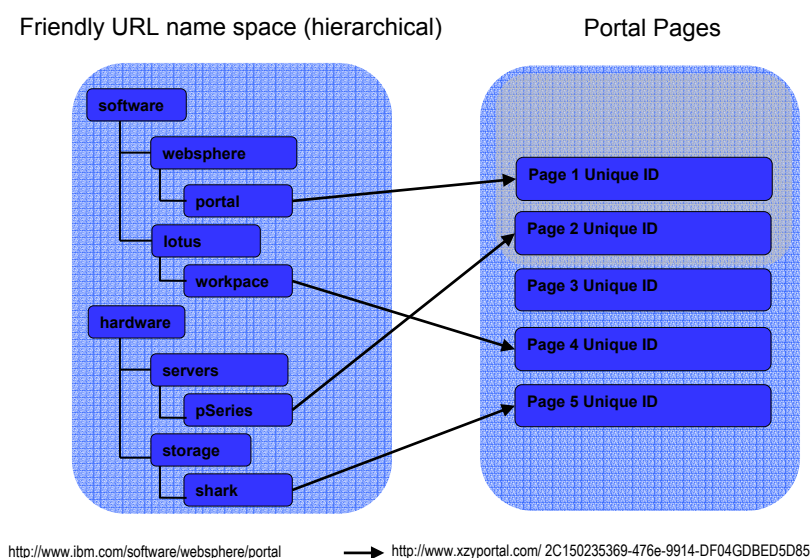
4. Následně odstraníme všechny soubory z adresáře <WAS_ROOT>/temp, abychom vynutili překompilování JSP v cache.
5. Restartujeme server WebSphere_Portal a úvodní obrazovka je bez položky *Sign in*.

5.6. Mapování URL

WebSphere Portal generuje velmi ošklivé adresy URL (viz například snímky obrazovek v kapitole 9), což některým uživatelům hrozně vadí. Oprávněně si stěžují, že tyto adresy prakticky nelze nikam opsat nebo telefonicky sdělit.

V řetězci URL je zakódována adresa stránky a navigační stav všech portletů na stránce. Nepodařilo se nám zjistit, proč portál používá zrovna tento způsob, když se nabízejí lepší (cookies, session proměnné apod.) – snad je to kvůli podpoře různých klientů a markupů.

Přestože jsou tyto URL stabilní (tj. většina jich je tzv. bookmarkovatelná), nelze je nijak předělat. Řešením je pouze mechanismus mapování URL, který umožňuje vytvořit jakousi virtuální strukturu stránek (viz obrázek 5.8), které se potom mapují na existující stránky.



Obrázek 5.8: Princip mapování URL

Nastavení mapování URL se provádí v administrační části portálu pomocí stejnojmenného portletu, viz obrázek 5.9.

5.7. Konfigurační rozhraní XML

Nástroj *XMLaccess* (skript `xmlaccess`) umožňuje exportovat a importovat nastavení části portálu. K dispozici jsou dva typy požadavků (export a update request), které je potřeba připravit jako soubor XML. *XMLaccess* potom jako odpověď vrátí také XML soubor.



Obrázek 5.9: Nastavení mapování URL

Nástroj XMLaccess může sloužit k zálohování portálu,⁹ dávkovému nastavování, nastavování bezpečnosti nebo třeba instalaci komponent.

Ukázka jednoduché žádosti o export:

```
<?xml version="1.0" encoding="UTF-8"?>
<request
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation="PortalConfig_1.2.xsd"
  type="export">

  <portal action="locate">
    <content-node action="export"
      uniquename="genericdepartment"
      export-descendants="true"
    />
  </portal>
</request>
```

Pro vytvoření žádosti je potřeba znát tzv. unikátní identifikátor prostředku, což je řetězec, který portál přiřadí každé stránce, portletu, aplikaci, ser- vletu, kontextu... Exportovat lze asi 20 různých typů objektů.

Aby si administrátor nemusel pamatovat systémové identifikátory, lze v administračním rozhraní portálu nastavit *přizpůsobená jedinečná jména* (custom unique names).

Poměrně dobrý popis XML rozhraní naleznete tentokrát v Infocentru, nej- lépe k verzi portálu 5.0.2.

⁹pokud zazálohujete celý portál + binární soubory portletů + motivy a skiny

Správa obsahu a další komponenty

Tato kapitola pojednává o správě obsahu a technologiemi, kterými portál umožňuje řídit obsah předkládaný uživateli – personalization, customization a další.

Obsah

6.1. Možnosti správy obsahu	66
6.2. Publikujeme statický obsah	66
6.2.1. Fileserver portlet	66
6.2.2. Proužková reklama	66
6.2.3. iFrame portlet	67
6.2.4. StaticText portlet	67
6.2.5. HTMLcontent portlet	68
6.2.6. CSV portlet	68
6.3. Publikujeme dynamický obsah	69
6.3.1. JSP Server portlet	70
6.3.2. RSS portlet	70
6.4. Správa dokumentů	70
6.4.1. Portal Document Manager (PDM)	71
6.4.2. Prohlížeče souborů	72
6.4.3. Cleverbee FileExplorer portlet	73
6.5. Personalization	74
6.6. Customization	75
6.7. Single-Sign-On	75
6.8. Credential Vault	76
6.9. Hledání	77

Na portál lze nahlížet jako na nástroj pro správu webového obsahu (web content management), protože interně obsahuje nástroje pro vytváření a správu stránek. Je ovšem nutné si uvědomit odlišnou filosofii portálu:

- veškerý obsah musí být zapouzdřen do portletu
- portál pouze dělá autorizační rozhodnutí, zda uživateli smí být portlet zobrazen

Portál WebSphere Portal tedy dělá jistý content management na úrovni stránek, ale nemá žádné mechanismy například pro authoring obsahu, schvalovací workflow, obměnu obsahu (rotaci) v čase. Požadovaná funkcionality musí být naprogramována v portletech.

Je-li potřeba vizuálně emulovat klasické webové stránky (bez portletů), je možné dekoraci portletů vypnout. Tím ale nezměníme filosofii portálu – skládání stránky z blíže graficky určených kousků. V porovnání s CMS systémy portál neumožňuje plnou kontrolu nad stránkou.

6.1. Možnosti správy obsahu

Na portálu můžeme z obecného hlediska publikovat obsah několika základními způsoby, kterými jsou (v pořadí od nejjednodušších k složitějším) vložení odkazu na daný dokument, vložení celé nebo části externího *HTML* dokumentu do portletu, uložení *HTML* dokumentů na portál, transformace „dynamických dat“ do portletu nebo použití systému pro správu obsahu. V následujících podkapitolách jsou prezentovány všechny tyto možnosti.

6.2. Publikujeme statický obsah

Běžnou potřebou na intranetovém portálu je dát do popředí nějakou důležitou informaci, kterou je možno rychle a snadno měnit. Je poněkud paradoxní, že takovouto triviální úlohu je poměrně obtížné pomocí prostředků dostupných ve WebSphere Portal vyřešit. Obvyčejný editovatelný statický *HTML* text, který se zobrazí v portletu například na titulní straně, se stane velkým problémem. Ten jsme nakonec vyřešili vlastním portletem. Část této podkapitoly je proto věnována portletům, které snadno a rychle řeší některé typické „content feed“ úlohy. Ty portlety, které jsou standardně součástí WebSphere Portal jsou podrobně popsány v knize [10].

6.2.1. Fileserver portlet

Tento portlet sice umí zobrazit statickou *HTML* stránku, kterou však musí administrátor portálu nakopírovat na server, kde portál běží, a to do adresáře `<WAS_HOME>/installedApps/FileServer_WPS_<portlet_id>.ear/~FileServer.war/FileServerPortlet/html`, kde `<portlet_id>` musíme zjistit z *URL* právě nainstalovaného *FileServer* portletu.

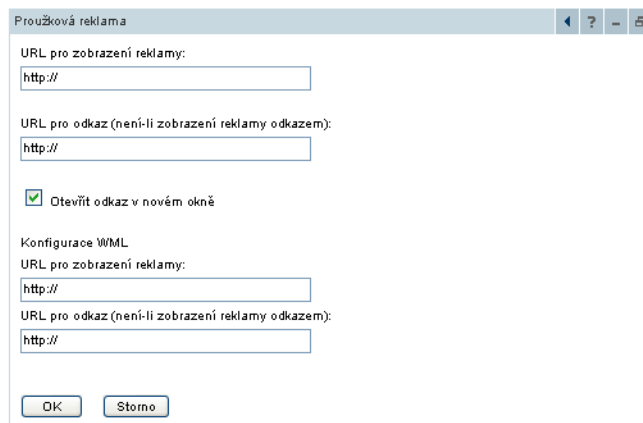
Cestu na zobrazovanou stránku musíme potom ještě nastavit v parametrech portletu (`portlet.xml`), podobně jako u portletu *JSPserver* – viz obrázek 6.7.

O masivním využívání tohoto nepraktického portletu tedy asi nebude nikdo dlouho přemýšlet.

6.2.2. Proužková reklama

Portletu *Proužková reklama* (*Banner Add*) je standardní součástí WebSphere Portal. V editačním režimu se nastavuje *URL* obrázku a případně i odkaz na webovou stránku odkazovanou obrázkem.

Využití portletu je širší, než jen proužková reklama, lze jej použít kdykoli je potřeba na stránku umístit nějaký obrázek. Potlačíme-li dekoraci portletu, dosáhneme efektu obyčejného obrázku na obyčejné HTML stránce.



Obrázek 6.1: Rozhraní portletu Proužková reklama (edit mode)

O tomto portletu bude ještě zmínka v následující kapitole.

6.2.3. iFrame portlet

Možnost začlenění obsahu externích webových stránek do portálu pomocí `<IFRAME>` je také diskutována v následující kapitole. K dispozici je opět portlet, který je standardní součástí WebSphere Portal, který se dá charakterizovat podobně jako *FileServer portlet* či *JSP Server portlet* (obr. 6.7), neboť jeho konfigurace probíhá stejným nemotorným způsobem.

Podstatně lepší vlastnosti má *iFrame Multilingual* portlet (obr. 7.4), který je volně nabízen španělskou firmou Sowre Consulting¹. V editačním režimu (obr. 7.5) je možné nakonfigurovat jeho titulek, atributy `<IFRAME>` a URL stránky, která má být v portletu zobrazena pro každý jazyk zvlášť.

6.2.4. StaticText portlet

Výše uvedené možnosti standardně dodávané s portálem nás příliš neuspokojily, neboť zvolený způsob publikování obsahu není vhodný běžného uživatele. Hledali jsme něco opravu pohotového a jednoduchého.

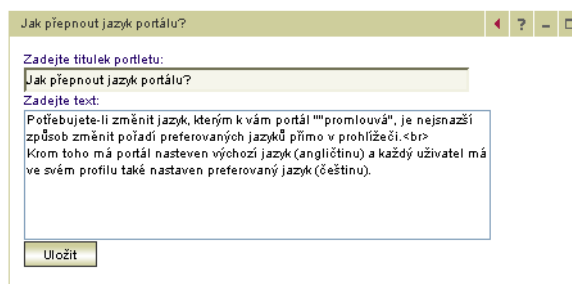
Proto jsme nakonec vytvořili vlastní portlet nazvaný *StaticText portlet* (obr. 10.4) pro jednoduché a rychlé publikování informací v podobě HTML fragmentů.

V editovacím režimu (obr. 6.2) je možné zadat *HTML* fragment a titulek portletu v anglické a české verzi. Uživateli se pak zobrazí varianta podle zvoleného jazykového prostředí. V současné době je připravována verze s *HTML WYSIWYG* editorem *HTMLarea*² podporující současně více jazyků.

Dalším krokem pak bude implementace primitivního *workflow*, které umožní především schvalování obsahu, jeho publikování nebo skrytí k danému datu.

¹www.sowre.es

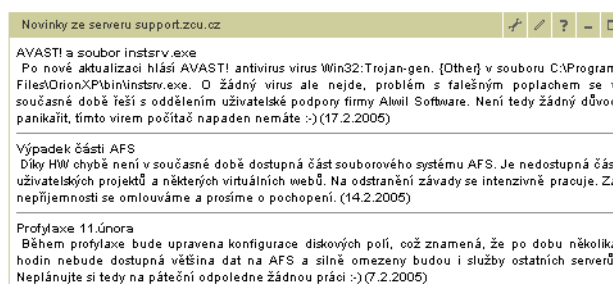
²www.htmlarea.com



Obrázek 6.2: Rozhraní portletu Static Text (edit mode)

6.2.5. HTMLcontent portlet

Tento portlet vyvinutý na Novgorodské univerzitě³ umožňuje zobrazit v portletu sadu HTML stránek lokálně uložených v definovaném adresáři na portálu. Portlet pracuje na principu propojení fyzického a virtuálního adresáře (kontextu) WAS.



Obrázek 6.3: Rozhraní portletu HTML Content (view mode)

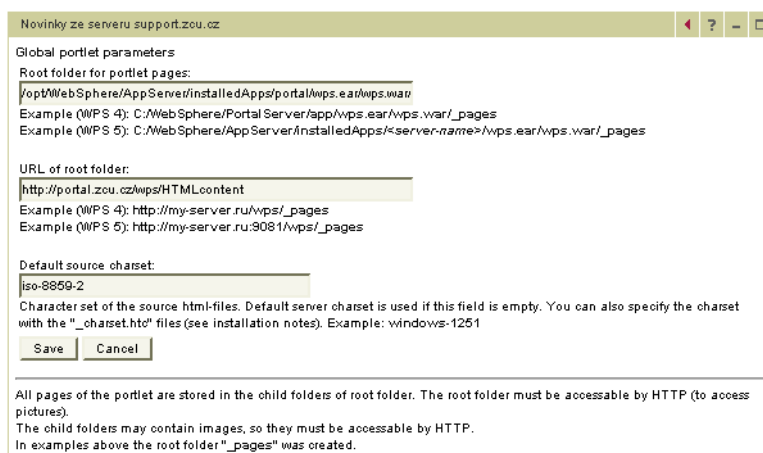
V konfiguračním módu portletu (obrázek 6.4) je zvolen kořenový adresář, kde jsou webové stránky uloženy a odpovídající virtuální adresář v URL, kvůli obrázkům a odkazům, například `<WAS_ROOT>/installedApps/<NODE>/wps.ear/wps.war/~/_pages/sample/image.jpeg` odpovídá `http://localhost:9081/wps/sample/~image.jpg`. Lokální linky je nutné označit složenými závorkami, aby k nim mohl portlet připojit svůj kontext.

V editačním módu (obr. 6.5) se určí titulek portletu a úvodní stránka, kterou lze nastavit zvlášť pro minimalizovaný a maximalizovaný portlet.

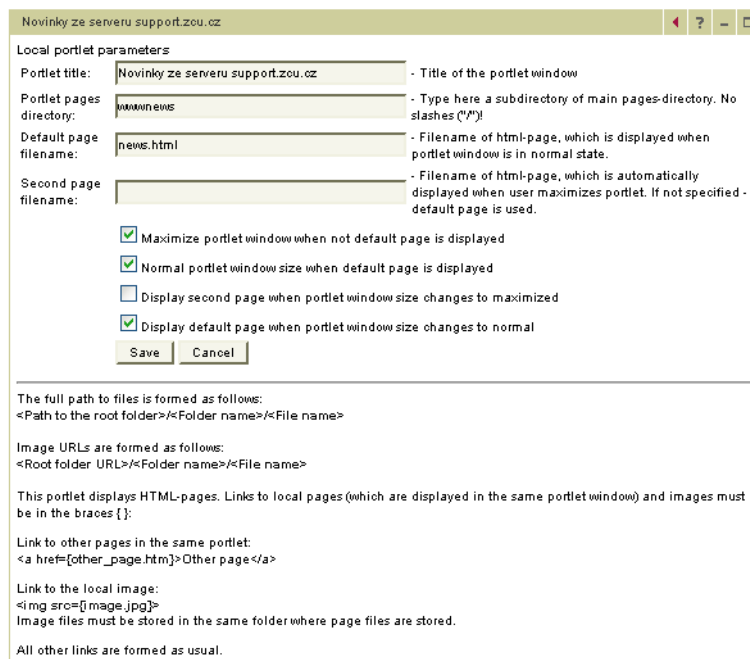
6.2.6. CSV portlet

Tento portlet je opět součástí WebSphere Portal a usnadňuje rychlé publikování tabelárních dat v portálu. Data ve formátu CSV (*Comma Separated Values*) zpřístupníme přes webový server a v editačním režimu portletu nastavíme jeho URL, volitelně uvedeme oddělovač jednotlivých hodnot, pokud jím není čárka. První řádek souboru obsahuje popisky sloupců tabulky.

³websphere.novsu.ac.ru



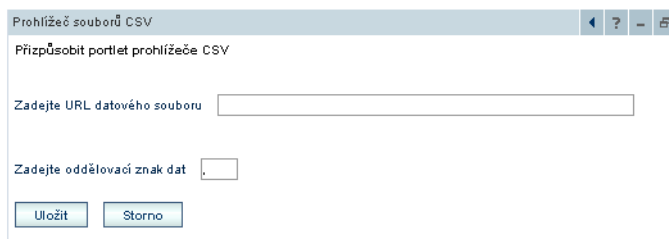
Obrázek 6.4: Rozhraní portletu HTML Content (configure mode)



Obrázek 6.5: Rozhraní portletu HTML Content (edit mode)

6.3. Publikujeme dynamický obsah

Cílem této kapitoly je ukázat možnosti publikace „dynamických dat“ v portálu. Pod dynamickými daty si můžeme představit data měnící se s časem, která jsou zpřístupněna v databázovém systému nebo přes *HTTP* v podobě XML či jiných souborů, ale i texty spravované pomocí systému pro správu obsahu, jakým je *WebSphere Portal Content Publisher – WPCP*. V *WebSphere Portal* jsou pro tento účel k dispozici již hotové portlety, které jsou postupně představeny v dalších odstavcích.



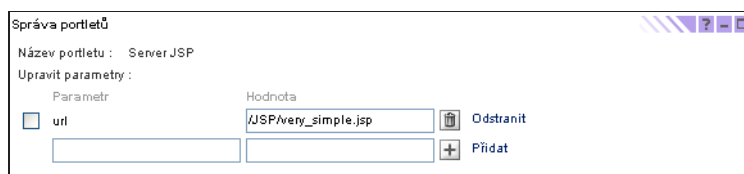
Obrázek 6.6: Rozhraní portletu Prohlížeč CSV (edit mode)

6.3.1. JSP Server portlet

Portlet *JSP Server* lze využít v případě, že chceme stávající *JSP* stránky převést do portálu. *JSP* stránku musíme ovšem „nakopirovat na portál do portletu“, stejně jako u *Fileserver* portletu.

Při instalaci portletu nastavíme v parametru *URL* cestu na *JSP* stránku uvnitř portletu, resp. i další volitelné parametry, které mají být *JSP* stránce předány.

Konfigurace přes *portlet.xml* (viz obr. 6.7) je nepraktická stejně jako u ostatních podobných portletů (*Fileserver* portlet a další). Výhodou je naopak možnost jednoduše využívat *JSP* tady portálu.



Obrázek 6.7: IBM JSP server – konfigurace portlet.xml

Firma *Sowre Consulting* opět nabízí lepší verzi tohoto portletu jako *JSP Server Multilingual* portlet, viz obr. 7.4 a 7.5.

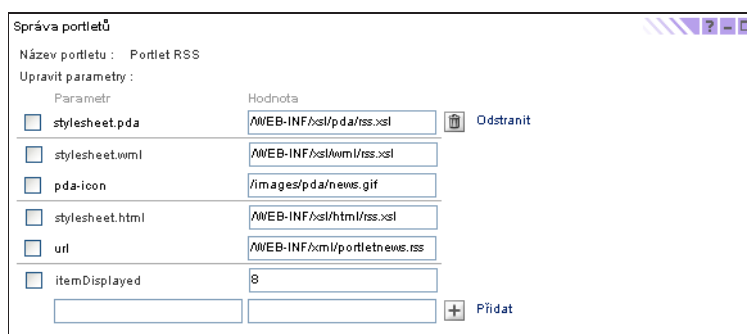
6.3.2. RSS portlet

Tento portlet je sice pojmenován *RSS (Rich Site Summary) Portlet*, ale protože je možné při jeho konfiguraci (obrázek 6.8) použít libovolný *XSL Stylesheet*, je možné ho využít pro tzv. *content feed* libovolných *XML* dat. Zároveň podporuje volbu *XSL* v závislosti na zařízení, z něhož si portlet prohlížíme (parametry *stylesheet.pda*, *stylesheet.wml* a *stylesheet.html*).

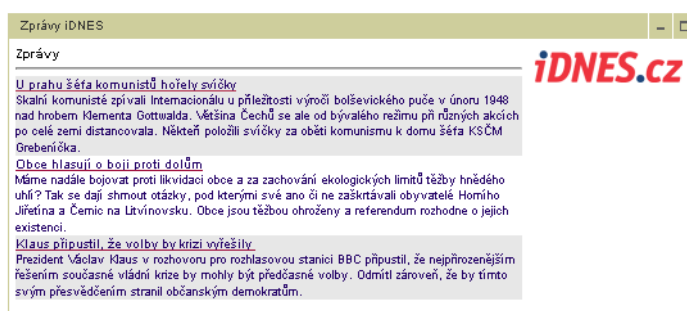
V portálu *ZČU* je tento portlet použit například k zobrazování novinek ze serveru *iDnes* (obrázek 6.9).

6.4. Správa dokumentů

Správa dokumentů je jednou z nepostradatelných komponent portálu. Umožňuje uživatelům sdílet důležité dokumenty, zařazovat je do definovaných kategorií, řídit přístup k těmto dokumentům, a zpravidla i nějaké *workflow*.



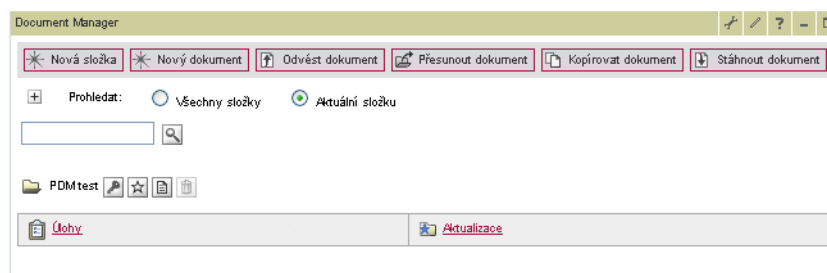
Obrázek 6.8: IBM RSS server – konfigurace portlet.xml



Obrázek 6.9: Rozhraní portletu RSS (view mode)

6.4.1. Portal Document Manager (PDM)

Pro správu a publikaci obsahu je ve WebSphere Portal 5.0 určena komponenta *Portal Document Manager – PDM* reprezentovaná stejnojmenným portletem (obrázek 6.10):

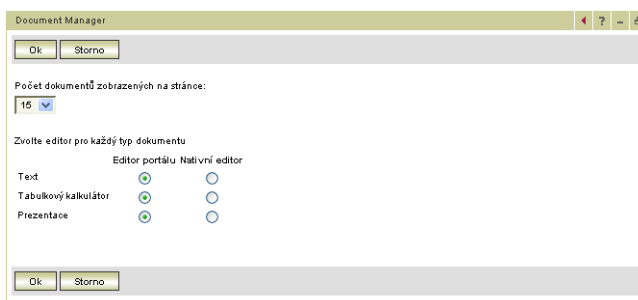


Obrázek 6.10: Portal Document Manager (view mode)

PDM poskytuje tyto funkce:

- procházení hierarchie adresářů
- rušení a vytváření nových adresářů
- vytváření, aktualizace a rušení dokumentů
- přesouvání a kopírování dokumentů
- vytváření verzí dokumentů
- schvalování, odmítání a komentování dokumentů

V editačním režimu *PDM* portletu (obr. 6.11) je možné nastavit počet zobrazených dokumentů na jedné stránce, a zda se má pro zvolený typ dokumentu použít vestavěný nebo nativní editor. *WYSIWYG* editor lze bohužel použít jen v prohlížeči *Microsoft Explorer*. Implementované workflow je příliš primitivní, *PDM* nabízí pouze varianty *no workflow* a *little workflow*.



Obrázek 6.11: Portal Document Manager (edit mode)

Podrobný popis funkce a ovládání *PDM* lze najít v osmé a deváté kapitole knihy [10] a curricula [12] doplněném o soubor cvičení v [13]. Uvedená literatura platí pro verzi portálu 5.0, ve verzi 5.1 jsou zahrnuty komponenty Lotus Workplace, které přinášejí mnohem lepší možnosti, zejména v oblasti správy dokumentů.

Na Ostravské univerzitě narazili spoluřešitelé na výkonové problémy při větším počtu uživatelů, na Západočeské univerzitě v Plzni pro změnu nefunguje *PDM* pod *HTTPS*. Sama firma *IBM* nepovažuje tento produkt za příliš zdařilý a ve verzi 5.1 ho nahradila novou verzí, které místo *WPCP* používá technologii akvizované firmy *Aprix*.

6.4.2. Prohlížeče souborů

Dalšími standardními portlety WebSphere Portal je skupina portletů určená pro zobrazování obsahu office dokumentů několika běžných formátů. Jsou to:

- *Excel Document Viewer* – pro zobrazení dokumentu typu Microsoft *XLS*
- *PDF Document Viewer* – pro formát Adobe *PDF*
- *PowerPoint Document Viewer* – pro formát Microsoft *PPT*
- *Rich Text Document Viewer* – pro formát *RTF*
- *Word Document Viewer* – pro formát Microsoft *DOC*

Podstatnou nevýhodou je, že na straně prohlížeče je třeba příslušný *plugin*, bez kterého není možné si daný typ dokumentu v portletu prohlédnout (obrázek 6.12), protože dokument je do okna portletu vložen použitím *HTML* značek `<OBJECT>` a `<EMBED>`.



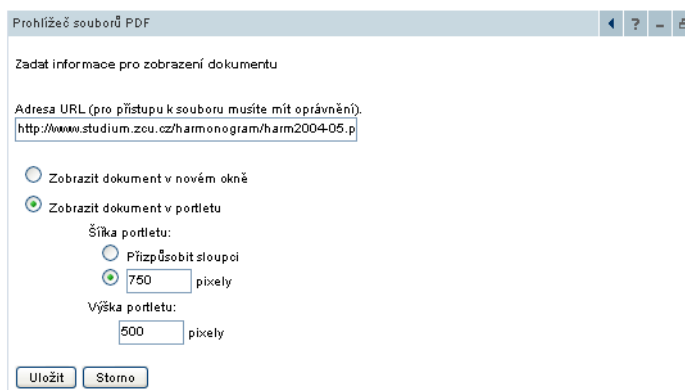
Obrázek 6.12: Prohlížeč souborů PDF (view mode)

Chceme-li zabránit rozbití vzhledu stránky, můžeme nastavit zobrazování dokumentu v novém okně, ale stejného (ne-li lepšího) efektu dosáhneme i portletem *Záložky*.



Obrázek 6.13: Prohlížeč souborů PDF (view mode)

V editačním režimu (6.14) můžeme nastavit, zda se má dokument zobrazit přímo v portletu nebo v nově otevřeném okně.



Obrázek 6.14: Prohlížeč souborů PDF (edit mode)

Je záhadou, proč tyto portlety nevyužívají stejnou technologii jako PDM, když už je jednou v portálu stejně nainstalovaná – viz přepínání editoru na obrázku 6.11.

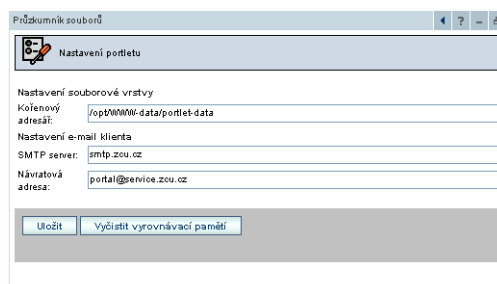
6.4.3. Cleverbee FileExplorer portlet

FileExplorer Portlet (obr. 6.15) byl zakoupen od firmy *Cleverbee*, jako náhrada za nefungující *Portal Document Manager – PDM*. Jeho výhodou je rychlé převedení již stávajících kolekcí dokumentů roztroušených po webových serverech univerzity do portálu, možnost odeslání dokumentu e-mailem a jednoduché náhledy dokumentů typů *MS Word*, *Excel*, *PDF* a *ZIP*.

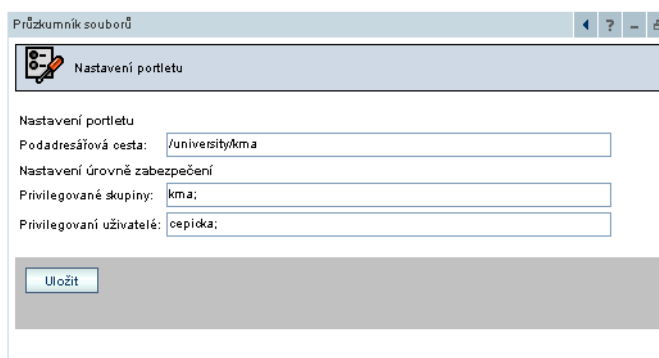


Obrázek 6.15: Rozhraní portletu File Explorer (view mode)

V konfiguračním režimu (obr. 6.16) je administrátorem portálu nastaven kořenový adresář a adresa SMTP serveru, v editačním režimu (obr. 6.17) je nastaven podadresář, jehož struktura má být prostřednictvím portletu dostupná. Dále jsou zde uvedeny skupiny a uživatelé, kteří mají právo rušit a vytvářet nové adresáře a provádět *upload* souborů.



Obrázek 6.16: Rozhraní portletu File Explorer (configure mode)



Obrázek 6.17: Rozhraní portletu File Explorer (edit mode)

Tento portlet je využíván v našich portálech především k zveřejňování studijních materiálů a pohodlnému sdílení dokumentů pro různé skupiny uživatelů (zápisy z porad a podobně).

6.5. Personalization

Personalizace představuje směřování obsahu a aplikací na specifické uživatele na základě určitých informací, které jsou o nich získávány a ukládány. Tyto informace jsou analyzovány a uživatelé jsou podle nich zařazeni do různých cílových skupin, kterým je pak „podsouván“ obsah spojený s jejich cílovou skupinou. V praxi to představuje nabízení informací a aplikací uživatelům, které by mohly zajímat. Tato problematika je velmi přehledně popsána v [10] v šesté kapitole.

Personalizaci umožňuje komponenta *WebSphere Personalization*, která je součástí WebSphere Portal a používá tzv. *rule-based personalization*, kde na základě definovaných pravidel jsou uživatelé zařazováni do různých cílových skupin a stejně tak dokumenty a aplikace. Pomocí průvodců ve *WSAD* je možné vytvořit runtime prostředí personalizační komponenty, jejíž součástmi jsou:

- *Resource Engine* – získává od WebSphere Portal informace o uživateli a dokumentech.
- *Rules Engine* – interpret pravidel.
- *Recommendation Engine* – vybírá obsah pro cílové skupiny.

V akademickém portálu nepřikládáme této technologii klíčový význam, protože uživatele máme „rozškátkované“ v LDAP registru a vystačíme si s deklarativní personalizací na bázi řízení přístupu ke stránkám a portletům.

Kdybychom naopak dělali internetový portál, kde jsou všichni uživatelé „stejní“ a zařazení „v jednom pytli“, pak by pro nás byla personalizace důležitá.

6.6. Customization

Customizace umožňuje uživatelům, aby si individuálně přizpůsobili portál svým potřebám: měnili obsah a vzhled stránek, přidávali a nastavovali portlety.

Je to žádaná, ale nebezpečná funkce, protože vznikají privátní prostředky, které nejsou dále říditelné... Při customizaci si uživatel vytvoří privátní kopii veřejného prostředku, která je potom odtržená od veřejné předlohy. Privátní kopie nereaguje na změny veřejné, nelze ji smazat jinak než smazáním veřejného prostředku.

Hlavním problémem je, že po customizaci již nijak nemůžeme upravovat stránku pro všechny uživatele – hlavně na ni umisťovat nové portlety tak, aby je viděli i ti uživatelé, kteří si stránku již přizpůsobili.

System tedy funguje způsobem „jednou si upravíš a potom se navždy starej sám“. Proto považujeme za lepší používat customizaci jen sporadicky.

Customizace se týká jen stránek; v případě portletů lze customizovat pouze tzv. konkrétní portlety, tj. instance portletů „sedící“ nebo „přilepené“ na stránce. Aby uživatel mohl používat editační režim portletu, musí být v roli privileged user k portletu i ke stránce (viz vztahy mezi rolemi na straně 5.3.2.).

Naše doporučení tedy zní: uživatelům klidně dejte právo privileged user na portlety, u kterých vznikne požadavek na individuální nastavení. Uživatelé však budou moci toto provést jen na těch stránkách, kde jim to dovolíte. Je ovšem potřeba ohlídat, aby portlet v editačním režimu neobsahoval nějakou bezpečnostní díru – například změnu adresáře mimo vymezený prostor. Na veřejné prostředky je tedy nutné dávat právo editor, přestože portlet má jen edit mode, do kterého by teoreticky stačilo právo privileged user.

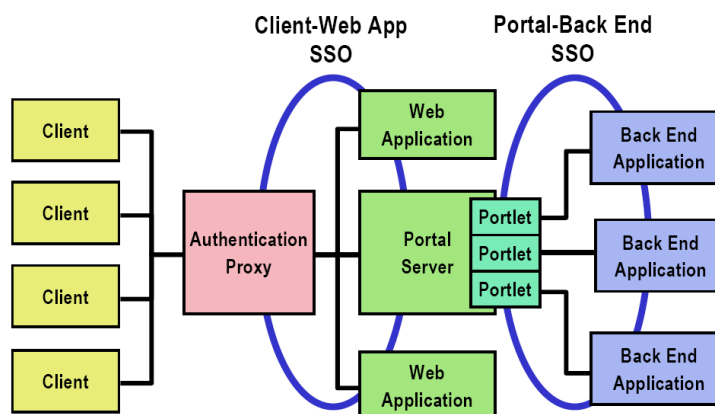
Několik dalších poznámek:

- je nutné dobře si vše předem dobře promyslet
- neexistuje jednotný úzus, co se nastavuje v editačním režimu a co v konfiguračním režimu portletů; je třeba otestovat chování každého portletu, u kterého chceme povolit customizaci
- běžní uživatelé by neměli mít právo editor na portlety (až na výjimky, kde to sémantika portletu bezpečně dovoluje)
- je nutné používat zámky kontejnerů tam, kde uživateli chceme zabránit odstranění portletu z customizovatelné stránky
- stránka může mít přiřazen seznam povolených portletů
- editor, manager a administrátor vždy editují veřejnou (primární) kopii stránky
- k odtržení privátní kopie stačí drobnost, třeba jen zaměnit pořadí portletů na stránce

6.7. Single-Sign-On

Single-Sign-On (SSO) je jednou ze základních funkcí portálů. Dovoluje, aby byl uživatel ověřen pouze jednou při přihlášení do portálu a přecházel mezi různými portálovými aplikacemi bez nutnosti opětovného ověřování uživatelským jménem a heslem.

SSO je zpravidla realizováno i mezi portálem a přidruženými webovými aplikacemi nebo portály jiných organizací. V tomto případě se mluví o tzv. *frontend SSO*.



Obrázek 6.18: Frontend a backend domény SSO

WebSphere Portal je v otázkách frontend SSO plně odkázán na podporu aplikačního server, který bohužel podporuje pouze proprietární metodu *LTPA token*, který se používá i při integraci s Tivoli Access Managerem nebo WebSeal. IBM bohužel strukturu *LTPA tokenu* neuvolní, takže možnost integrace například se systémem WebKDC (webové SSO pro Kerberos vyvinuté na MIT) používaným na ZČU není možná. Alternativní cestou by mohlo být SSO založené na PKI certifikátech.

Jednotného přihlašování lze docílit i na aplikační úrovni což znamená, že uživatel nemusí prokazovat svojí identitu pokud přistupuje z portálu k databázi podnikového informačního systému. Pak hovoříme o tzv. *backend SSO*. Z technického pohledu lze snadno předávat (nebo přemapovat) *principal* webového uživatele do backend systémů, ale naopak velmi obtížné je předat jeho *credential*. Obvykle nezbývá než nějakým způsobem uložit v portálu heslo pro backend systém.

Backend SSO lze v WebSphere Portal realizovat pomocí komponenty *Credential Vault* popisované v dalším odstavci. Celá situace je schematicky znázorněna na obrázku 6.18.

6.8. Credential Vault

Při integraci stávajících informačních systémů do portálu narazíme na problém, že tyto informační systémy není možné napojit na portálovou infrastrukturu tak, aby se staly součástí portálové SSO domény. Uživatelé k nim mají zpravidla separátní uživatelská jména a hesla, která by museli znovu zadávat i po přihlášení do portálu v každém portletu, který k takové aplikaci přistupuje.

V rámci WebSphere Portal toto řeší komponenta nazvaná *Credential Vault*, v překladu *trezor mandátů*. Jedná se o úložiště uživatelských jmen a hesel do stávajících informačních systémů, kam je uživatel zadá pouze jednou při prvním přihlášení a portál je pak již automaticky používá k další autentizaci – funguje to podobně jako správce hesel (password manager) známý třeba z webových prohlížečů.

Credential Vault se dělí na tzv. *segmenty* a *sloty*. Jeden segment představuje konkrétní autentizační doménu (nebo také informační systém), ve které zpravidla bývá několik slotů. V těchto slotech jsou uložena uživatelská jména a hesla. Pokud aplikace používá pouze jediného uživatele, je v segmentu slot pouze jeden, který sdílí všichni oprávnění uživatelé. Slot může obsahovat pouze jedno heslo (resp. *credential*).

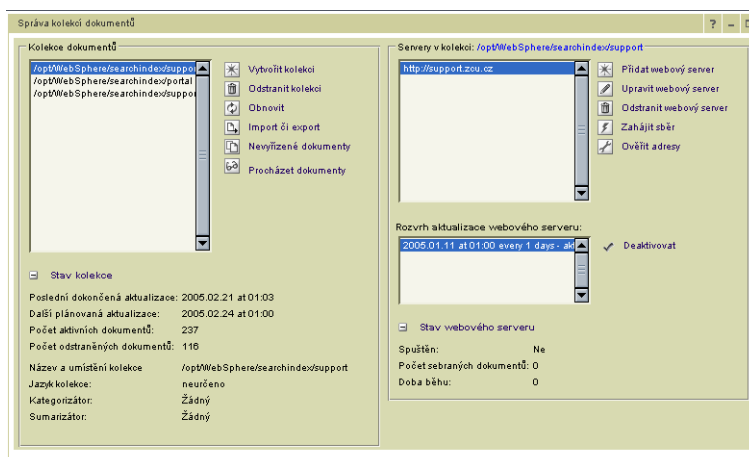
6.9. Hledání

Portál disponuje už v základní verzi nástrojem pro prohledávání HTML stránek (web crawler), který je reprezentován dvěma portlety: Document Search (obr. 6.19) a Search Administration (obr. 6.20). Oba mají nedostatečnou dokumentaci i nápo- vědu, proto je nejlepší postupovat dle [1].



Obrázek 6.19: Rozhraní portletu Hledání dokumentů (view mode)

Princip konfigurace (viz obr. 6.20) spočívá ve vytvoření kolekce, do které se zařadí servery, jež se mají prohledávat. Každý server může mít nastaveno několik filtra- cích pravidel a další parametry. Využití katalogizátorů a sumarizátorů je nejasné, protože není dokumentováno. Zřejmě je nutné nastavit alespoň jedno „kladné“ fil- trovací pravidlo pro dokumenty, jinak nejsou žádné dokumenty zaindexovány.



Obrázek 6.20: Správa kolekcí dokumentů (view mode)

Dlouho nás trápil fakt, že hledání nefunguje. Nakonec nám pomohla informace označená jako *tip* v [1]. Portlet Document Search je třeba nakonfigurovat na jednu konkrétní kolekci, a to pomocí administračního rozhraní *Správa portletů*. Portlet totiž nemá konfigurační režim.

Od té chvíle je hledání použitelné, ale detailnímu zkoumání funkce jednotlivých parametrů jsme se nevěnovali.

Dalším problémem je prohledávání samotného obsahu a dokumentů v portálu. V network news konferenci proběhlo na toto téma několik dotazů, ale bez jasného závěru. Pro prohledávání veřejného obsahu lze použít zmíněný integrovaný crawler, pro prohledávání dokumentů snad Portal Document Manager (PDM), který je ale na jiném místě (strana 71) klasifikován jako nepoužitelný.

Vývoj a integrace aplikací

Bez aplikací (portletů) je portál jen prázdná infrastruktura, která uživateli nenabízí prakticky žádnou užitečnou funkcionalitu. WebSphere Portal sice obsahuje několik desítek portletů ihned po instalaci, nicméně mnoho portletů je potřeba naprogramovat na míru organizace.

Obsah

7.1. J2EE standardy a API	80
7.2. Zdroje informací aneb J2EE není legrace	81
7.3. Metodika integrace aplikací do portálu	81
7.3.1. Možnosti přístupu k IS v portálu	82
7.3.2. Integrace na databázové vrstvě	82
7.3.3. Integrace přes nativní aplikační rozhraní	84
7.3.4. Integrace na prezentační vrstvě	85
7.4. Integrace SAP do WebSphere Portal	87
7.4.1. SAP HTML GUI Integration	88
7.4.2. SAP Business Warehouse Integration portlet	88
7.4.3. SAP Enterprise Integration Portlets	88
7.4.4. IBM SAP MSS My Staff Portlets	88
7.5. WebSphere Portal Application Integration	88
7.5.1. JDBC Portlet Builder	89
7.5.2. SAP Portlet Builder	92
7.5.3. Instalace a konfigurace portletů	95
7.6. Vývoj vícevrstevných aplikací pro portál	95
7.6.1. Úvod do EJB	96
7.6.2. Vývoj EJB pomocí IBM WebSphere Studia	97
7.6.3. Metodiky použité při vývoji EJB	97
7.6.4. Přehled vytvořených EJB	99
7.6.5. Řešení autentizace ve vícevrstvé architektuře	99
7.6.6. Prezentační vrstva – úvod do portletů	100
7.6.7. Přehled vytvořených portletů	100
7.7. IBM WebSphere Studio	100
7.7.1. Vývoj portletů ve WebSphere Studiu	101
7.7.2. Instalace WSAD	101
7.8. IBM Rational SDP	102
7.9. Nastavení JDBC Datasource	103

Schopnost vlastního vývoje portletů považujeme za jeden z klíčových faktorů úspěchu portálu. V každém portálovém projektu je potřeba vyrobit portlety s požadovanou (chybějící) funkcionalitou a integrovat stávající aplikace, což obvykle znamená mnoho programování.¹

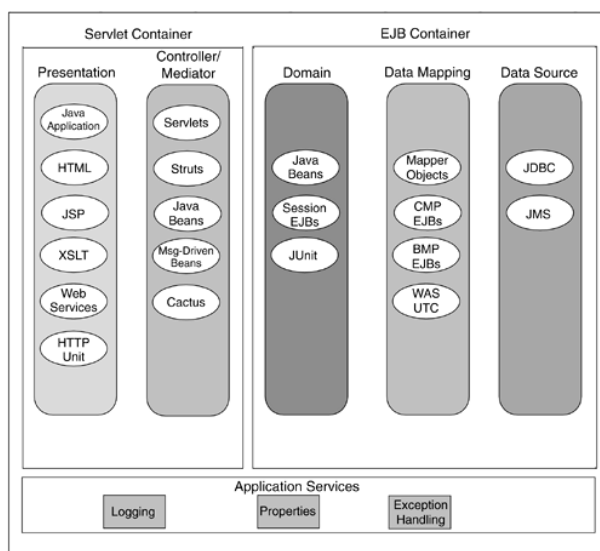
Většina portálů na trhu je založena na jazyce Java, a ani IBM WebSphere Portal není výjimkou. Portlety se programují buď v proprietárním *IBM Portlet API* nebo lze využít otevřený standard JSR168, který ale v současné době ještě nemá takové schopnosti a podporu jako nativní API produktu.

V první části této kapitoly se zabýváme vývojem portletů (prezentační vrstvy) a EJB (aplikační vrstvy) třívrstvého softwarového modelu. Druhá část je věnována integračním nástrojům pro stávající neportálové aplikace.

Problematiku vlastního vývoje portletů považujeme za tak důležitou, že jsme použili část diplomové práce Zdeňka Gallistla [ZCU-2] jako přílohy B a C.

7.1. J2EE standardy a API

Standardů a API je v prostředí *J2EE* celá řada a jejich specifikace jsou velmi rozsáhlé. Definují vždy vlastnosti komponent v určité vrstvě *J2EE* aplikace. Obrázek 7.1 ukazuje uplatnění konkrétních *J2EE* API, standardů a technologií v jednotlivých vrstvách.



Obrázek 7.1: Vrstvy J2EE aplikace

V dokumentaci k *J2EE* platformě se lze dočíst, že „*J2EE* je množina koordinovaných specifikací a metodik, které společně dávají možnost snadného vývoje, nasazení a správy vícevrstevných aplikací založených na architektuře klient-server“. Nové specifikace nebo jejich nové verze procházejí schvalovacím řízením sdružení *Java Community Process – JCP (jcp.org)* v podobě *Java Standardization Request – JSR*. *J2EE 1.3* lze najít pod *JSR-058*, současnou verzi 1.4 pod *JSR-151*. Ty však nedefinují detailně celé prostředí *J2EE*, ale jen jeho rámcovou architekturu a použité technologie, které jsou specifikovány v dalších *JSR*. Například specifikaci *EJB 3.0* najdeme jako *JSR-220*.

¹Existují samozřejmě i cesty bez programování (web services, různé integrátory), ale ty nejsou vždy k dispozici nebo poskytují jen částečné řešení.

Součástí *J2EE* jsou i specifikace *API* k mnoha primárním technologiím implementovaných ve formě Java balíků. Jsou jimi zejména:

- *Java API for XML-Based RPC (JAX-RPC)*
- *JavaServer Pages (JSP)*
- *Java Servlets*
- *Enterprise JavaBeans Components (EJB)*
- *J2EE Connector Architecture (J2CA)*
- *J2EE Management Model*
- *J2EE Deployment API*
- *Java Management Extensions (JMX)*
- *J2EE Authorization Contract for Containers*
- *Java API for XML Registries (JAXR)*
- *Java Message Service (JMS)*
- *Java Naming and Directory Interface (JNDI)*
- *Java Transaction API (JTA)*
- *Common Object Request Broker Architecture (CORBA)*
- *Java DataBase Connectivity (JDBC) Data Access API*

V závěru tohoto odstavce je nutně říci, že prostředí *J2EE* je velice komplexní a jemu věnovaná literatura je zpravidla velice rozsáhlá. Proto je další odstavec věnován přehledu důležitých informačních zdrojů.

7.2. Zdroje informací aneb *J2EE* není legrace

Informačních zdrojů o *J2EE* architektuře je celá řada, důležité je jen najít a zaměřit se na ty opravdu kvalitní. Na prvním místě jsou bezpochyby odpovídající specifikace dostupné na webových stránkách firmy Sun – java.sun.com, které však nejsou dobrým materiálem pro vývojáře začínající s *J2EE*. Dále je pro začátečníka velmi důležitá možnost obrátit se na někoho pro radu. V tomto směru můžeme jednoznačně doporučit on-line *Enterprise Java* komunitu TheServerSide.com, jejíž zakladateli jsou *Ed Roman* a *Floyd Marinescu*, autoři knih *Mastering Enterprise JavaBeans* a *EJB Design Patterns*, které jsou dobrou úvodní literaturou a jsou volně dostupné na TheServerSide.com.

K tomu, abychom implementovali kvalitní *J2EE* aplikaci, však nestačí pouze dodržovat předepsané specifikace či metodiky. Vyplatí se opětovně použít již osvědčených řešení dílčích problémů, to znamená využití tzv. *návrhových vzorů*². Stěžejní publikací v této oblasti je kniha *Design Patterns: Elements of Reusable Object-Oriented Software* čtveřice autorů *E. Gamma*, *R. Helm*, *R. Johnson* a *J. Vlissides*, kteří si vysloužili přezdívku *The Gang of Four*. Při vývoji *J2EE* aplikací však bude užitečnější kniha *Core J2EE Patterns: Best Practices and Design Strategies*, která je i volně dostupná na java.sun.com.

7.3. Metodika integrace aplikací do portálu

Důvodem, proč se zabývat integrací aplikací je, že každá organizace dříve či později dospěje k potřebě mezi sebou propojit klíčové aplikace, které již nelze dále provozovat jako „izolované ostrůvky dat“. Může k tomu dojít i cestou přes portál.

V zásadě lze integraci aplikací rozdělit do následujících tří úrovní:

²Viz heslo *návrhový vzor* na adrese cs.wikipedia.org

1. *Integrace dat* – je nejstarší způsob integrace aplikací. Zpravidla je realizován ad-hoc mechanismy, např. pravidelný import/export dat na úrovni databázového systému nebo tzv. *datovou pumpou*. Současné databázové systémy již většinou podporují různé typy replikace.
2. *Integrace aplikací* – je zaměřena na vzájemnou konektivitu aplikací, které je dosaženo využitím různých komunikačních subsystémů. Zpravidla je použit *Message Oriented Middleware (MOM)*, jako jsou *JMS*, *IBM MQSeries*, *Tibco Rendezvous*, *BEA Tuxedo* nebo vzdálené volání procedur – *Remote Procedure Call (RPC)*, kde lze například využít *Java RMI*, *CORBA IIOP* nebo *SOAP*.
3. *Integrace podnikových procesů* – již nezahrnuje pouze data, ale i procesy a sahá až za hranice samotné organizace. Příkladem může být spojení informačních systémů firmy se systémy dodavatelů a odběratel, kde průběh objednávky a dodávky zboží včetně zaplacení faktury je řízen automaticky. Tento druh integrace lze rozdělit na další dvě vrstvy, kde spodní vrstva představuje implementaci procesů, naproti tomu horní vrstva je odpovědná za vzájemnou koordinaci těchto procesů.

J2EE prostředí nabízí celkem tři možnosti (technologie) pro integraci aplikací:

- *JMS a Message-Driven Beans* – podpora komunikace *EJB* ↔ *MOM*
- *Java Web Services* – podpora komunikace *EJB* přes webové služby (např. s *Microsoft .NET*).
- *J2EE Connector Architecture* – rámec pro tvorbu adaptérů nebo konektorů umožňující komunikaci *EJB* a specifického podnikového informačního systému.

Dobrym úvodem do problematiky integrace podnikových aplikací je kniha [37].

7.3.1. Možnosti přístupu k IS v portálu

Řešení, která vyžadují programování jsou náročná na čas, ladění a vyžadují práci programátora obeznámeného s programováním v prostředí portálu. Proto jsme se (na UPOL) zaměřili na využití „neprogramátorských“ prostředků pro obě cesty k informacím v IS.

Neprogramátorskými prostředky jsou v prostředí WebSphere Portal portlety, které se dokáží napojit na nějaký databázový zdroj nebo na nějaké nativní rozhraní IS a zobrazit případně i modifikovat takto získaná data. Která data, odkud a jakým způsobem se mají zobrazit, to by mělo být předmětem konfigurace takovýchto portletů.

Portlety splňující výše uvedené vlastnosti jsme hledali jednak v sadě portletů přímo instalované při instalaci portálu, jednak mezi volně použitelnými portlety z *katalogu WebSphere Portal a Lotus* firmy IBM, který je umístěn na stránce

<http://catalog.lotus.com/wps/portal/portalworkplace>.

7.3.2. Integrace na databázové vrstvě

Přímý přístup k databázi má své výhody i nevýhody. Výhodou je jednoduchost a standardizace přístupu, kdy se využívá jednotné databázové rozhraní prostředí Java. Nevýhodou je pak nutnost omezit se ve většině případů pouze na čtení dat a změny ponechat procedurám IS, které zajišťují integritu dat. Pro tento přístup je také nutná znalost struktury databáze IS.

Přímou práci s daty v databázi lze v praxi často použít v případech, kdy uvedený IS obsahuje nějaký způsob exportu vybraných informací do tabulek v jiné databázi vně prostředí IS. Takovouto funkčnost v nějaké podobě poskytuje většina IS. Postup je pak takový, že se prostředky IS v pravidelných intervalech provádí export informací

do externí databáze („pumpování“) a tyto informace jsou pak „online“ zobrazovány v prostředí portálu. Většinou bývá velký okruh informací, které jsou určeny pouze k prohlížení a které není nutno mít k dispozici opravdu online.

Pro přístup k databázi existuje v prostředí Java standardizovaný postup – JDBC API.

JDBC API poskytuje základní rozhraní pro unifikovaný přístup k databázím. Základem konceptu JDBC je využití funkčnosti poskytované JDBC ovladačem, který je následně překládá do nativních volání dané databáze. Díky tomu je aplikační programátor odstíněn od specifického API databáze a může se naučit jednotné rozhraní JDBC, které pak použije pro přístup do libovolné databáze, která poskytuje JDBC ovladač. V současné době to jsou prakticky všechny hlavní systémy a ovladače jsou optimalizované a vyvíjené samotnými výrobci databázových serverů.

Aplikační server WebSphere implementuje serverové prostředí podle specifikace *J2EE (Java2 Platform Enterprise Edition)*. Součástí tohoto prostředí je sada služeb – mezi nimi *JNDI (Java Naming and Directory Interface)* a kontejner pro EJB (Enterprise Java Beans) poskytující službu sdílení zdrojů (resource pooling).

Toto prostředí umožňuje oddělit konfiguraci přístupů k databázi od vlastního výkonného kódu a podporuje sdílení připojení.

Doporučený postup pro přístup k databázi v aplikačních portletech je využít konfiguraci datového zdroje v aplikačním serveru WebSphere a tento datový zdroj zpřístupnit přes JNDI pro portlety; není tedy žádoucí, aby v portletu bylo „zadržováno“ konkrétní spojení na databázi.

Pro integraci na datové vrstvě jsou s IBM WebSphere Portal standardně dodávány dva portlety: *Sestavy dotazů (SQL portlet)* a *Tvůrce podnikových objektů (JDBC portlet Builder)*. Oba portlety se blíže popíšeme.

7.3.2.1. Portlet Sestavy dotazů

Pro zobrazení výsledků databázových dotazů existuje v základní sadě portletů instalace WebSphere Portal portlet s názvem *Sestavy dotazů (SQL Query portlet)*.

Obrázek 7.2: SQL portlet (view mode)

V tomto portletu se v editačním režimu konfiguruje pohledy na data. Seznam těchto

pohledů a jejich vyvolání má pak k dispozici každý uživatel stránky.

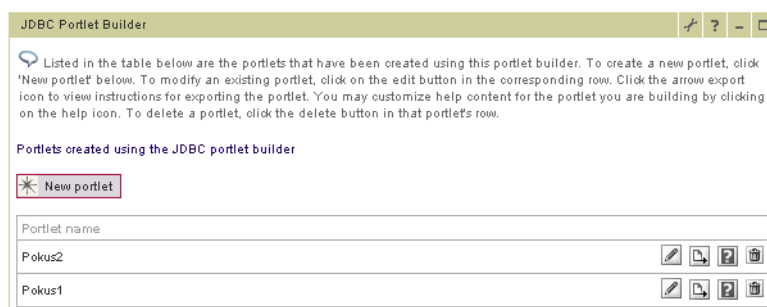
Pohled je nakonfigurován zadáním JDBC ovladače, jména a hesla uživatele, databázového URL a SQL příkazu pro výběr dat. Portlet nepoužívá datové zdroje.

Po vyvolání dotazu je výsledek zobrazen ve tvaru tabulky s definovaným počtem sloupců. Sloupce je možno pojmenovat. Pro větší počet datových řádků provádí portlet stránkování – umožňuje procházení výsledků dotazu po stránkách s menším počtem řádků.

Výhodou portletu je jeho jednoduchost. Nevýhodou je, že portlet neumožňuje používat JNDI datové zdroje.

7.3.2.2. JDBC Portlet Builder

Jednoduché a rychlé zpřístupnění informací z databázových systémů v portálu umožňuje *JDBC Portlet Builder*. Bez nutnosti programování lze pomocí této aplikace, mající charakter průvodce, snadno vytvořit portlety zobrazující data z tabulek databáze ve zvolené formě. Protože ve skutečnosti nevytváříme fyzický portlet, ale pouze konfigurační XML soubor, který je interpretován vestvěným *Runtime*, je možné snadno provést export vytvořeného „portletu“ pomocí nástroje *XML Access* a nainstalovat jej v jiném portálu. Podrobněji je *JDBC Portlet Builder* popsán v odstavci 7.5.1.



Obrázek 7.3: JDBC Portlet Builder

7.3.3. Integrace přes nativní aplikační rozhraní

Velké informační systémy většinou nabízejí nějakou programovou možnost přístupu k informacím nebo i jejich modifikaci. Mimo vlastní uživatelské rozhraní to může být:

Konektor umožňující programátorský přístup k informacím v aplikaci vytvořený na základě různých programátorských technologií, například:

- simulace JDBC/ODBC ovladače
- SOAP webová služba
- sada COM objektů
- assembly v prostředí .NET
- knihovna přístupových objektů v prostředí JAVA.

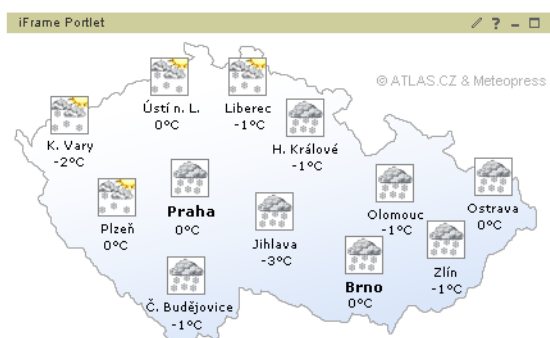
Pro integraci do prostředí portálu je nejvhodnější webová služba nebo knihovna Java.

Při zkoumání přístupu k IS jsme se zaměřili na spolupráci portálu a systému SAP R/3 a dále popíšeme možnosti této integrace.

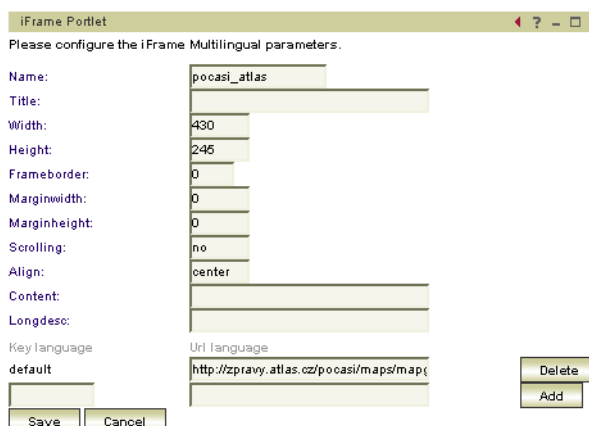
7.3.4. Integrace na prezentační vrstvě

Integraci na prezentační vrstvě zpravidla představuje využití obsahu stávajících webových stránek v portálu. Zde lze použít tzv. *web-clipping*, kde na straně serveru (portálu) je na určité místo (do portletu) vložen výstřížek specifikované webové stránky. Nevýhodou tohoto přístupu je, že změna obsahu originálu může mít nežádoucí vliv na fragmentu portletu.

Jiné řešení, na straně klienta, nabízí tzv. *iFrame*, kdy je na základě *HTML* značky `<IFRAME>` prohlížečem vložen obsah specifikované webové stránky do jiné. Na tomto principu je založen *iFrame Portlet*, u kterého v *Edit* módu (viz obrázek 7.5) specifikujeme URL webové stránky, která má být zobrazena ve *View* módu (obrázek 7.4). Nevýhodou je však, že ne všechny webové prohlížeče tuto novou značku podporují a je možné s její pomocí vložit pouze celou stránku.



Obrázek 7.4: iFrame portlet (view mode)



Obrázek 7.5: iFrame portlet (edit mode)

Pro *web-clipping* je ve WebSphere Portal k dispozici *editor webových výstřížků*. Při jeho konfiguraci (obrázky 7.6 a 7.7) je nastaven jeho titulek, popis, URL vkládané stránky, pokud stránka vyžaduje autentizaci uživatelské jméno a heslo a další volby. K dispozici jsou následující tři způsoby specifikace výstřížku:

1. Převzetí celého obsahu stránky,
2. převzetí fragmentu mezi specifikovaným textem,
3. manuálním výběrem obsahu ve zvláštním WYSIWYG editoru.

Takto nakonfigurovaný portlet je pak možné zařadit na zvolenou stránku portálu. Ukázka, jak lze pomocí *WebClip* portletu zařadit například vyhledávač *Google*, je na obrázku 7.8.



Obrázek 7.6: Editor webových výstřížků (Web Clipping)



Obrázek 7.7: Vytváření klipu

V určitých situacích postačuje pouhé začlenění grafických výstupů (obrázků) generovaných jinými aplikacemi.



Obrázek 7.8: Ukázka webového klipu

K tomu lze využít portlet *Záložky (Bookmarks)* nebo portlet *Proužková reklama (BannerAdd)* – viz strana 66.

Principiálně se tyto vizuální metody integrace na úrovni prohlížeče liší jen tím, jaká značka je vložena do HTML kódu stránky a jak komfortně nebo flexibilně lze portlet konfigurovat.

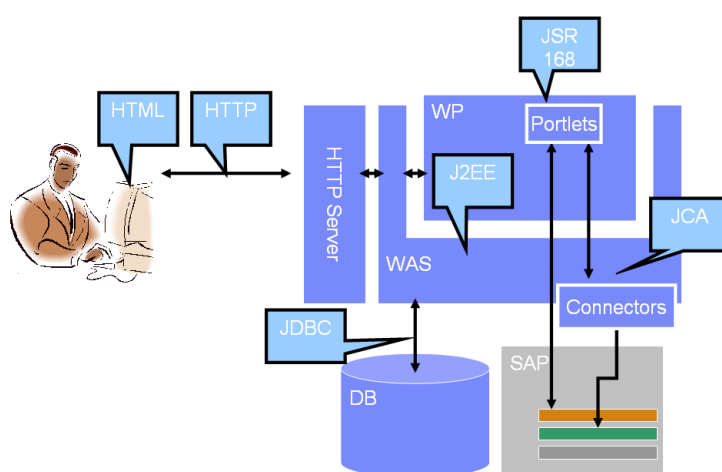
Obecně lepší je asi používat web clipping pro jeho lepší říditelnost a kešování obsahu, naopak proti mluví vyšší procesorová zátěž portálového serveru, proměnný clipping JavaScriptu (obvykle nefunguje) a CSS (kolidují se styly portálu). Navíc se tato komponenta WebSphere Portal chová trochu nevyzpytatelně – zřejmě má

chyby v logice editoru portletů. Jestliže váš výstřížek nefunguje správně ačkoli jej různě konfiguruje, zkuste jej snazat a vytvořit znovu na jeden zátah. Obvykle to pomůže.

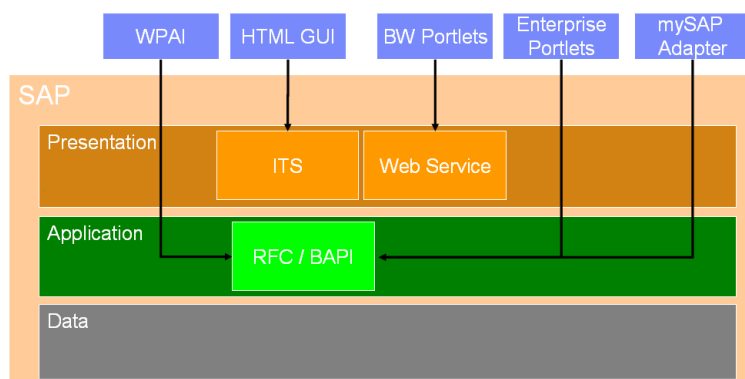
7.4. Integrace SAP do WebSphere Portal

V případě integrace systému SAP se nabízí hned několik možností, jak (na jakém místě) propojení realizovat (obr. 7.10).

Systém SAP je rozsáhlá sada programových komponent poskytující externí rozhraní na svoji prezentační a aplikační vrstvě (viz obrázek 7.10). Portlety pro kooperaci mezi Websphere Portal a SAP využívají přístup k některé z těchto dvou vrstev.



Obrázek 7.9: Propojení WPS a SAP



Obrázek 7.10: Možnosti napojení se na SAP

- **Integrace na prezentační vrstvě**

Zobrazení některé SAP prezentační vrstvy v prostředí portálu. Sem spadají portlety:

- SAP HTML GUI Integration
- SAP Business Warehouse Integration portlet

- **Integrace na aplikační vrstvě**

Portlety se připojují přes Java SAP konektor přímo do aplikační vrstvy SAP na BAPI funkce zapouzdřující jednotlivé funkčnosti aplikační vrstvy. Sem spadají portlety:

- *WebSphere Portal Application Integrator (WPAI)*
- *IBM SAP Enterprise application*
- *SAP Enterprise Integration Portlets*
- *IBM SAP MSS My Staff Portlets*

- **Programátorská tvorba portletů pro SAP:**

Nelze-li použít některé z existujících portletů, je možno na programovat nové. Při tvorbě se může použít přímo rozhraní *SAP JCO – SAP Java Connector*. V rámci tohoto rozhraní lze naprogramovat přihlášení k serveru SAP a následně vyvolání BAPI funkce se vstupními parametry a poté zpracovat a zobrazit výstupní parametry a výstupní tabulky volané funkce.

Pro účinnější práci programátora existuje *mySAP.com JCA adaptér* (JCA je standard pro připojování EIS systémů do J2EE aplikací). Tento adaptér generuje proxy třídy pro BAPI nebo RFC SAP funkce a EJB (Enterprise Java Beans). Pro jeho použití je nutná instalace *WebSphere Studio Application Developer Integration Edition*.

7.4.1. SAP HTML GUI Integration

Integrace SAP HTML rozhraní z produktu SAP Internet Transaction Server (SAP ITS).

Používá HTML tag `<IFRAME>` pro vložení stránky z ITS do portletu. Pro uložení identifikace SAP uživatele je použito SSO a Credential Valut.

7.4.2. SAP Business Warehouse Integration portlet

Sada portletů, která se připojuje na BW server – *SAP Business Warehouse Queries*. Jedná se o nádstavbu systému SAP pro dotazování a datový sklad. Portlet využívá IFRAME pro zobrazení prezentační stránky BW. Sada obsahuje 28 portletů.

7.4.3. SAP Enterprise Integration Portlets

Portlety pro vazbu na SAP HR zaměstnanci. Používají *mySAP.com* konektor pro připojení. Lze je upravovat a rozšiřovat na úrovni JSP stránek, které jsou jejich součástí. Předpokládají instalaci *Connection Factory for mySAP.com* adaptér a její zpřístupnění přes JNDI jmennou konvenci.

7.4.4. IBM SAP MSS My Staff Portlets

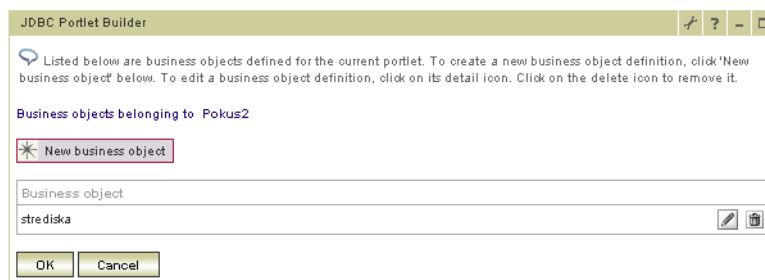
Sada několika portletů pro manažery zaměřená na zobrazování personálních informací.

7.5. WebSphere Portal Application Integration

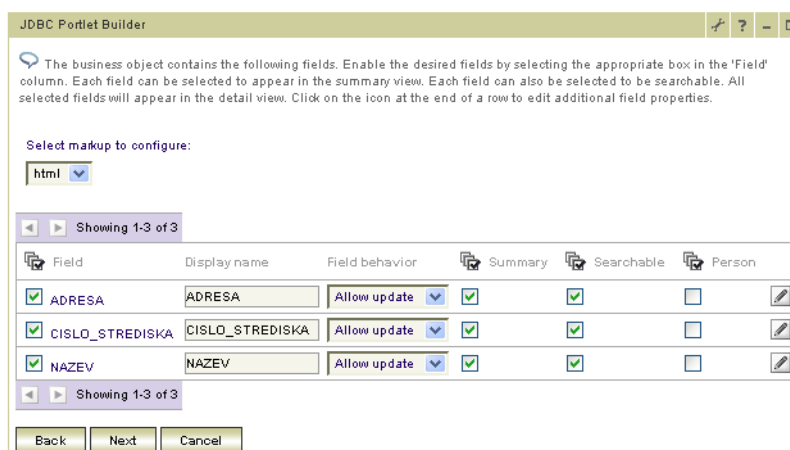
Sada technologií a portletů WPAI obsahuje část pro spojení portálu s databázemi a s prostředím SAP: *JDBC Portlet Builder*, *SAP Portlet Builder* a *SAP Business Object Editor Portlet*.

7.5.1. JDBC Portlet Builder

Tento portlet umožňuje vytvářet další portlety, které mohou přistupovat a manipulovat s daty různých databází, bez znalosti programování. Informace o databázi jsou dostupné pomocí „podnikových objektů“ (*business objects*), což jsou kolekce relačních dat, se kterými můžeme manipulovat pomocí standardních operací, jako je *create* nebo *update*.



Obrázek 7.11: JDBC Portlet Builder – editace portletu



Obrázek 7.12: JDBC Portlet Builder – editace business objektu

Portlety vytvořené pomocí JDBC Portlet Builderu jsou ve skutečnosti jen XML soubory vytvořené podle jisté šablony, které jsou potom prováděny runtime částí JDBC Portlet Builderu. Nový název celé technologie je *WPAI - WebSphere Portal Application Integrator*

Vytvořené portlety lze přenést do jiného portálu pomocí programu `xmlaccess` – viz strana 62.

7.5.1.1. Konfigurace JDBC portlet builderu

Před tím, než poprvé použijete JDBC Portlet Builder k vytvoření portletů, musíte jej nakonfigurovat tak, aby komunikoval s příslušnou aplikací (databází) a to v následujících krocích:

1. Na uvítací obrazovce klikněte na konfigurační ikonu
2. Vyplňte požadované informace
Configuration Settings – zde zadáte JNDI jméno datového zdroje, tak jak je nadefinované na WebSphere Aplikačním Serveru.

User ID – jméno uživatele, pod kterým chcete přistupovat k aplikaci.

Password – Heslo uživatele, pod kterým chcete přistupovat k aplikaci.

Confirm password – Potvrzení hesla

3. Uložte nastavení pomocí *Save settings*.

7.5.1.2. Vytvoření portletu pomocí JDBC portlet builderu

Nejprve je potřeba provést konfiguraci portletu (portlet nemusí používat stejné datábazové schéma jako portlet builder), a to v následujících krocích:

1. Zmáčkněte *New portlet*
2. Napište jméno portletu do pole *Portlet Name*
3. Zkontrolujte nastavení JNDI
4. Vyberte požadovanou volbu autentizace:
 - *Prompt users for user ID and password* – Tato volba vyzve uživatele k autentizaci, když se poprvé přistupuje k portletu.
 - *Specify slot name to create* – Zde vyplňte název slotu, který se má vytvořit.
 - *Use this user ID and password* – Tuto volbu použijte v případě, že nechcete uživatele autentizovat.
 - *Specify slot name to create* – Zde vyplňte název slotu, který se má vytvořit.
 - *User ID* – Zde vepište uživatelské jméno.
 - *Password* a *Confirm password* – Napište a potvrďte heslo.
 - *Use existing credential vault slot* – Vyberte existující slot ze seznamu a použijte uživatelské ID a heslo, které už je nastavené ve slotu.
5. *Select a template* – Zvolte šablonu.
6. Zmáčkněte *Next* pro pokračování nebo *Cancel* pro opuštění zadávání.

Pak vytvoříte nový Business object v rámci vytvářeného portletu:

1. Stiskněte *New business object*
2. Vyplňte následující parametry: *Schema name* – Vyberte schéma ze seznamu. *Table name* – Vyberte název tabulky ze seznamu za použití vyhledávacího řetězce a zmáčkněte ikonu *Get list*, např. když jako vyhledávací řetězec použijete %, získáte všechny tabulky zvoleného schématu.
3. Klikněte na požadovanou tabulku.
4. Zmáčkněte *OK* pro pokračování nebo *Cancel* pro opuštění zadávání.
5. Do pole *Business object name* vepište název vašeho Business Objectu.
6. Vyberte akci, kterou chcete nad daty povolit a k tomu můžete doplnit odpovídající zprávy (potvrzovací, po úspěšném provedení, po chybě).
7. Pokud chcete, aby portlet komunikoval a zobrazil korespondující informace v jiných portletech, tzv. *Click-to-Action* (C2A), klikněte na tlačítko *Add*. Vyplňte pole *Type* a *Namespace*, která musí být shodně vyplněna ve zdrojovém i cílovém portletu. Vyberte pole, přes které se mají portlety svázat a klikněte na *Add*. Takto můžete vybrat více polí.

Vyberte akci (*Create*, *Search*, *Update*), která má být povolena s touto *Click-to-Action*. Např. pokud zvolíte *Update*, provede se změna ve zdrojovém portletu a následně bude provedena i v provázaném portletu. Zmáčkněte *OK* pro pokračování nebo *Cancel*, pro opuštění zadávání.

8. Vyplňte tabulku složenou z následujících polí:

Field – Datová pole a k nim check box, zda se má pole zobrazit v detailním pohledu portletu.

Field behavior – Definiuje vlastnosti datového pole (*Read only*, *Allow update*, *Creation time*).

Summary – Zda se pole objeví v souhrnném pohledu.

Searchable – Zda se bude podle tohoto datového pole vyhledávat.

Person – Tento check box indikuje, zda datové pole obsahuje jméno nebo e-mail osoby. Pokud je osoba online, je toto pole vysvíceno a přichystáno na chatování.

Ikona tužky – Slouží k editaci vlastností pole.

- *Field Controls*: Tato obrazovka umožňuje nastavit vlastnosti datového pole. Vyberte požadovaný typ:
 - *Text box: Width* – Zde zadejte požadovanou šířku textového pole.
 - *Set default create value* – Defaultní hodnota, při vytvoření nové věty.
 - *Prefill with specified value* – Naplní se konstantní hodnotou.
 - *Prefill with user profile attribute* – Naplní se aktuální hodnotou atributu uživatele.
 - *Prefill with current date/time* – Naplní se aktuálním datem, napište formát, ve kterém datum požadujete (formáty jsou v Java třídě *java.text.SimpleDateFormat*).
 - *Set default search value* – Tato volba umožní nastavit defaultní hodnotu pro vyhledávání.
 - *Allow users to modify default search value in edit mode* – Pole zaškrtněte v případě, že chcete povolit uživateli modifikovat defaultní hodnoty pro vyhledávání.
 - *Text box with data retrieval* – Umožňuje naplnit textové pole hodnotou z jiného business objectu.
 - *Text area* – Umožňuje nastavit velikost textového pole a defaultní hodnotu pole.
 - *Drop down* – Zde zadáváte dvojici název (*Dropdown options*) a hodnota (*Options values*), které se pak zobrazí v seznamu k danému poli při vyhledávání.
 - *Business Object dropdown* – Vyberete si ze seznamu business object a pak vyberete pole *Dropdown options* a k němu odpovídající pole hodnot *Dropdown*.
 - *Check box* – Vyplní se hodnoty zaškrtnutého a nezaškrtnutého stavu a vybere se defaultní stav.
 - *Radio button* – Vyplní se dvojice voleb hodnota - název a určí se, která z voleb bude defaultní.
- *Click-to-Action* – Vyberte pole, přes které chcete business objekty spojit a vyplňte pole *Type* a *Namespace* shodně s tím, jak jste je vyplnili v bodě 7.
- *Image and Link* – Zde můžete nastavit obrázek nebo link k poli.
- *Provide link* – Zde nastavíte URL cestu, musí být kompletní (např. <http://www.ibm.com>). Můžete takto předat i parametr, když do tohoto políčka přidáte konstantu %%hodnota%%
- *Display image* – Do pole *Image URL* zadejte kompletní URL cestu k obrázku.

9. Po ukočení zadávání vlastností polí přejděte pomocí tlačítka *Next* na další obrazovku, kde si můžete pomocí šipek nastavit pořadí zobrazovaných polí.

10. *Rows per page* – Zadejte počet řádků na stránce, které chcete zobrazit (defaultně je nastaveno 20 řádků).

11. *Columns per page* – Uvedte počet sloupců, které se mají zobrazit v detailním pohledu.
12. Tlačítkem *Finish* uložíte nastavení.

Vytvořené portlety jsou vlastně jen XML soubory, které provádí runtime WPAI. Díky tomu lze portlety snadno přenášet mezi portály pomocí exportu-importu skriptem `xmlaccess`.

7.5.2. SAP Portlet Builder

Tento portlet umožňuje vytvářet portlety, které mohou přistupovat a manipulovat s daty databáze mySAP bez znalosti programování. Informace o databázi jsou dostupné pomocí business objektů, což jsou kolekce relačních dat, se kterými můžeme manipulovat pomocí standardních operací, jako je *create* nebo *update*. Tento portlet funguje podobným způsobem jako portlet *JDBC Portlet Builder*.

Obrázek 7.13: SAP Portlet Builder

7.5.2.1. Konfigurace SAP Portlet Builderu

Hostname – jméno nebo číslo serveru, kde je nainstalovaný SAP.

Client number – číslo klienta.

System number – číslo systému.

Language – dvouznakový kód země (např. cs)

Connection per user pool – maximální počet sdílených konekcí na jednoho uživatele.

User ID – uživatelské jméno pro přístup do SAPu.

Password – heslo pro přístup do SAPu.

Confirm password – potvrzení hesla.

7.5.2.2. Konfigurace portletu pomocí SAP Portlet Builderu

1. Zmáčkněte *New portlet*.
2. Napište jméno portletu do pole *Portlet Name*.

3. Ověřte síťová nastavení.
4. Vyberte požadovanou volbu autentizace.
 - *Prompt users for user ID and password* – Tato volba vyzve uživatele k autentizaci, když se poprvé přistupuje k portletu.
 - *Specify slot name to create* – Napište název slotu, který se má vytvořit.
 - *Use this user ID and password* – Tuto volbu použijte v případě, že nechcete uživatele autentizovat.
 - *Specify slot name to create* – Napiš název slotu, který se má vytvořit.
 - *User ID* – Vepište uživatelské jméno.
 - *Password a Confirm Password* – Napište a potvrďte heslo.
 - *Use existing Credential Vault slot* – vyberte existující slot ze seznamu a použijte uživatelské ID a heslo které už je nastavené ve slotu.
5. *Select a template* – Zvolte šablonu.
6. Zmáčkněte *Next* pro pokračování nebo *Cancel* pro opuštění zadávání.

Pak vytvoříte nový *Business Object*:

1. Stiskněte *New business object*
2. Vyberte business object ze seznamu nebo napište jméno tabulky v SAPu (pouze pro čtení z této tabulky).
3. Dále už pokračujte shodně jako v JDBC Portlet Builderu.

Business objekty můžeme modifikovat nebo nové vytvářet pomocí portletu *SAP Business Object Editor*. Tento portlet umožňuje vytvářet nové XML mapující soubory bez nutnosti psaní XML a jeho znalosti. Tyto XML soubory namapují funkce SAPu do business objektů, které pak mohou být použity v SAP Portlet Builderu (jeden XML soubor může být využit ve více portletech).

7.5.2.3. Konfigurace SAP Business Object Editoru

Hostname – jméno nebo IP adresa serveru, kde je nainstalovaný SAP.

Client number – číslo klienta.

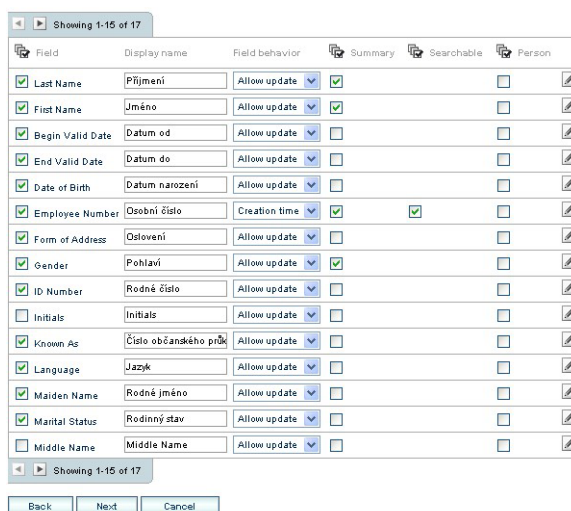
System number – číslo systému.

Language – dvouznakový kód země (např. cs)

Slot name – název slotu, který byl vytvořen pro SAP.

7.5.2.4. Vytvoření nového objektu v SAP Business Object Editoru

1. Stiskněte *New business object*.
2. Vepište název business objectu.
3. Klikněte na záložku *Retrieve*, *Create*, *Update* nebo *Delete* k definování funkcí, které jsou asociovány s každou akcí.
4. Vyberte typ funkce (*Pre-function*, *Main-function*, *Post function*). Pomocí *Pre-function* je specifikována funkce, která bude provedena před hlavní funkcí.
5. Klikněte na ikonu *Select SAP function*.
6. Vyberte název funkce jedním ze dvou způsobů:
 - (a) Vepište název BAPI funkce tak, jak je pojmenována v SAPu a stiskněte šipku. V SAPu seznam všech BAPI funkcí najdete v transakci BAPI.



Obrázek 7.14: SAP Business Object Editor

(b) Hledejte podle názvu nebo části názvu business objektu. Pomocí šipky získáte seznam všech business objektů definovaných v SAPu. Vyhledávání je case senzitivní. Pak klikněte na ID business objektu, který chcete použít.

7. Nadefinujte parametry v následujících 3 záložkách:

(a) *Import Parameters* – Objeví se seznam vstupních parametrů. Zmáčkněte ikonu *Set field mapping* a vyplňte následující údaje (jestliže pole obsahuje Y ve sloupci *Required*, je nutné toto pole namapovat).

Field name – Pod názvem, který zde zadáte, se bude pole zobrazovat.

Field location – Zadejte umístění pole.

Data type – Vyberte datový typ pro pole.

Padded length – Vepište délku pole.

Field flags:

Include from search – Vyhledávací parametr je včleněn do detailního a sumárního pohledu.

In summary – Pole bude zobrazeno ve výsledném pohledu.

Required – Při vkládání nebo updatu bude toto pole zobrazeno s červenou hvězdičkou a musí být vyplněno.

Searchable – Toto pole bude zobrazeno na vyhledávací obrazovce.

Editable – Toto pole bude zobrazeno při updatu.

Visible – Toto pole se nebude ukazovat.

Primary Key – Pole je primárním klíčem.

Computed – Pokud je pole zaškrtnuté, slouží jen pro čtení. Jestliže toto pole není zaškrtnuté a není možné jej editovat, pak se toto pole zobrazí při vkládání nové věty a nezobrazí se při updatu.

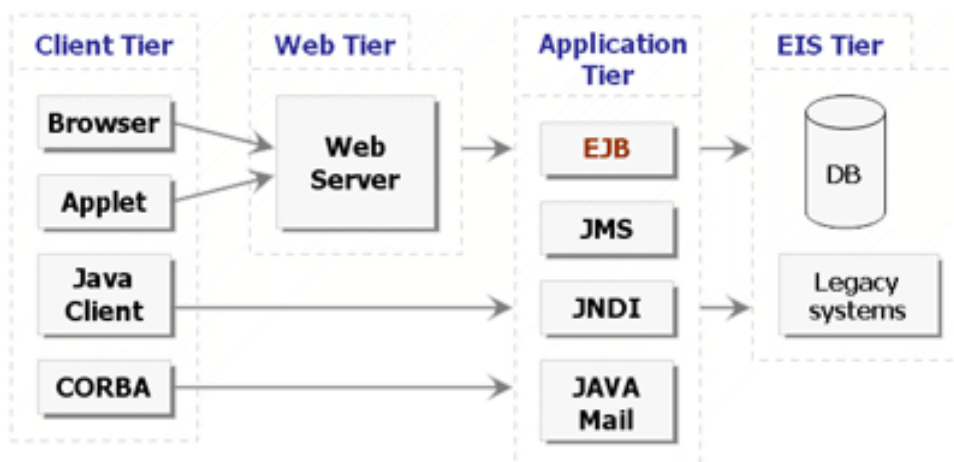
Default value – Zde můžete vložit defaultní hodnotu pole.

Namapovaná pole potom svítí modře ve sloupci *Mapping* pod názvem, který jsme zvolili v poli *Field name* a je před nimi ikona, která informuje, zda se jedná o parametr vstupní nebo výstupní.

(b) *Export Parameters* – Vyplňte exportní parametry.

(c) *Table Parameters* – Vyplňte parametry tabulky.

8. Pokud máme parametry nadefinované, stiskneme *Done*.



Obrázek 7.15: Vícevrstvá architektura v prostředí J2EE

9. Záložka *XML Preview* zobrazí XML soubor, který je výstupem SAP Business Object Editoru.
10. Tento XML soubor potom nahrajte na portálový server do adresářů `sap_config`.

7.5.3. Instalace a konfigurace portletů

Pro plnou funkčnost WPAI portletů pro přístup k JDBC a k SAP se musí nainstalovat následující portletové aplikace z balíčku *IBM Application Portlet Builder*:

- `WPAIPortletBuilders50.war`
- `WPAISAPEditor50.war`
- `WPAIDefaultTemplate50.war`

Instalaci (aktualizaci) těchto portletů proveďte v administrátorském rozhraní portálu.

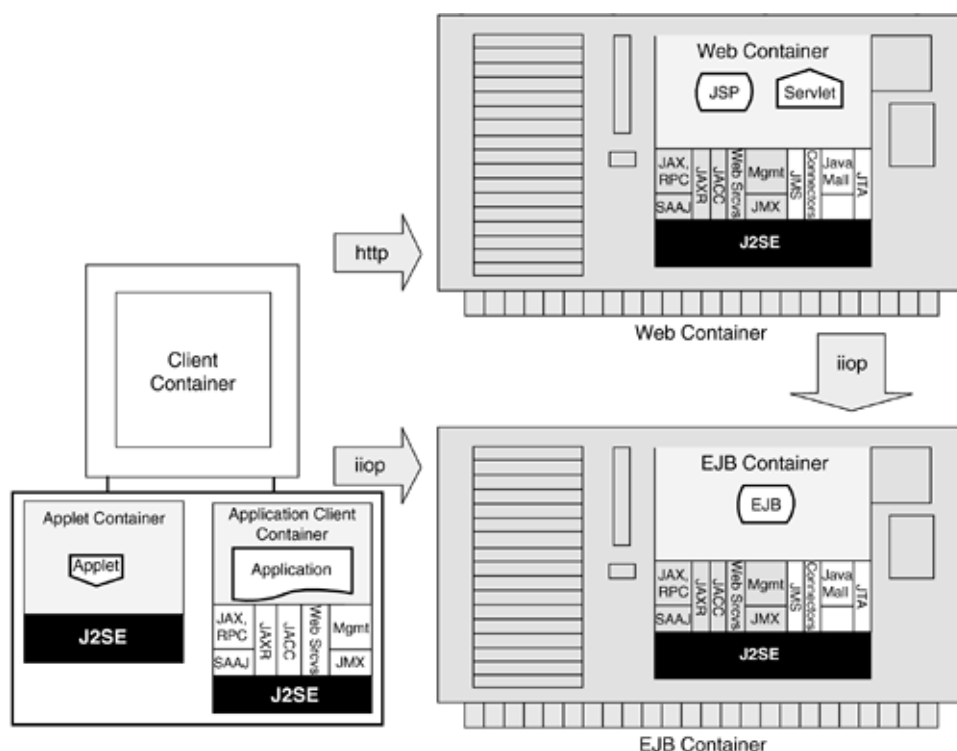
7.6. Vývoj vícevrstevných aplikací pro portál

Aplikace vyvíjené pro portál mají zpravidla třívrstvou architekturu. Spodní vrstvu představuje databázový systém, mezilehlá vrstva představuje aplikační logiku, horní – prezentační – vrstva tvoří uživatelské rozhraní. Výhodou aplikace této architektury je, že tím dosáhneme modulárnosti aplikace, možnost opětovného použití komponent, vyšší produktivity programátorů díky specializaci na konkrétní vrstvu aplikace nebo modul a jednodušší správy aplikace.

Vícevrstvou architekturu využívá i prostředí *J2EE – Java 2 Enterprise Edition*, jednotlivé vrstvy a vazby mezi nimi ukazuje obrázek 7.15.

Hlavní myšlenkou prostředí *J2EE* je komponentová technologie. Prostředím, ve kterém komponenty pracují, je *J2EE Container*, přičemž vlastnosti kontejneru a komponent jsou specifikovány v klíčových *J2EE* specifikacích. Pro každou vrstvu je v podstatě definován určitý typ komponent s odpovídajícím kontejnerem.

Nutno ještě dodat, že *J2EE* kontejnery umožňují snadnou a pro vývojáře transparentní škálovatelnost aplikace. Jednou ze služeb kontejnerů je *Component Pooling*, některé modernější aplikační servery (*IBM WebSphere Application Server*, *JBoss*) podporují i spojování do tzv. *klusterů*.



Obrázek 7.16: Typy J2EE kontejnerů a vazby mezi nimi

7.6.1. Úvod do EJB

Enterprise Java Beans – EJB jsou komponenty pracující v prostředí EJB kontejneru, které implementují části aplikační logiky. Protože součástí aplikace mohou být různé typy klientů (webový prohlížeč nebo jiná aplikace komunikující přes rozhraní webových služeb), umožňují tedy *EJB* i snadné opětovné využití aplikační logiky na více místech. Současná verze je *EJB 2.1*, jejíž specifikaci je dostupná jako *JSR-019*. Specifikace definuje vlastnosti a rozhraní *EJB* kontejneru, jím poskytované služby a rozhraní a definici tří typů *EJB* komponent:

- *Session Bean* – typicky obsahuje tzv. *Business Logic*, zpravidla vykonává požadavky klienta.
- *Entity Bean* – je persistentní objekt, který reprezentuje určitá data v databázi, zpravidla záznam v tabulce.
- *Message-Driven Bean* – má stejnou úlohu jako *Session Bean*, je však narozdíl od něj asynchronní a lze s ním komunikovat výhradně přes rozhraní *JMS*.

Z praktického pohledu je *EJB* komponenta soubor ve formátu *JAR – Java Archive*, který obsahuje binární kód komponenty a *XML* popisovače definující obsah archivu (může obsahovat více *EJB* komponent), nastavení případných parametrů a informace o napojení *EJB* na okolí. Tento popisovač se nazývá *Deployment Descriptor* a má podobu jednoho nebo více *XML* souborů. Archiv může obsahovat i potřebné knihovny v podobě *JAR* archivů a další souboru. Přesný obsah archivu a formát *XML* popisovačů je definován v odpovídající verzi specifikaci *EJB*.

Instalace komponenty do kontejneru na aplikační server (*deployment*) se provede nástrojem specifickým pro daného výrobce, kde je zaregistrována do jmenné a adresářové služby *JNDI*. Ostatní komponenty *J2EE* aplikace ji tak snadno najdou podle definovaného jména a mohou s ní pak komunikovat prostřednictvím *RMI*, *JMS* nebo přímo.

Dobrým úvodem do *EJB* je kniha *Mastering Enterprise JavaBeans*, která je i volně dostupná na *The Server Side*.

7.6.2. Vývoj EJB pomocí IBM WebSphere Studia

Tvorba *EJB* je ve *WebSphere Application Developer* – *WSAD* studiu velice snadná. Vytvoříme nový *EJB Project* dle specifikace *EJB 2.0* tak, že v menu postupně zvolíme *File* → *New Project* → *EJB*. *WSAD* automaticky vytvoří rámec pro *EJB* včetně deskriptorů (zatím prázdných). V tomto projektu můžeme vytvářet *EJB* komponenty definovaných typů tak, že zvolíme *File* → *New* → *Other...* → *EJB* → *Enterprise Bean*. Pomocí tohoto průvodce vytvoříme kostru *EJB* zvoleného typu a vlastností, automaticky jsou odpovídajícím způsobem aktualizovány i příslušné deskriptory. Zbývá jen definovat rozhraní komponenty a implementace funkcí definovaných v tomto rozhraní.

Máme-li implementaci komponent hotovou, zabalíme je snadno do archivu odpovídajícího specifikaci *EJB* tak, že provedeme export projektu do archivu *JAR*. Je-li *EJB* projekt součástí projektu *J2EE* aplikace, pak stačí pouze export aplikace, ostatní podprojekty jsou přibaleny automaticky.

Posledním krokem je instalace aplikace na aplikačním serveru. Tu provedeme pomocí administrační konzole *WebSphere Portal* tak, že v levém menu vybereme *Applications* → *Install New Application*. Ve formuláři v pravé části administrační konzole použijeme volbu *Local path* a provedeme stiskem tlačítka *Next* upload archivu na server. V dalším formuláři ponecháme nabídnuté nastavení a tlačítkem *Next* se dostaneme na průvodce instalací aplikace, s jehož pomocí nainstalujeme aplikaci (v našem případě pouze *EJB* komponenty) v následujících sedmi krocích:

1. Nastavení základních voleb instalace aplikace,
2. namapování instalovaných *EJB* do *JNDI*,
3. namapování vazeb na další využívané jiné *EJB*,
4. namapování vazeb na využívané (datové) zdroje,
5. specifikování aplikačních serverů v klusteru kam mají být *EJB* nainstalovány,
6. definice bezpečnostních politik pro nechráněné metody *EJB* komponent
7. a potvrzení nastavení parametrů instalace.

Stiskem tlačítka *Finish* potvrdíme instalaci a provedené změny uložíme volbou *Save to Master Configuration*. Uložení změn je nutné ještě jednou potvrdit tlačítkem *Save*. Poté aplikaci spustíme tak, že v levém menu zvolíme *Applications* → *Enterprise Applications*, v pravém okně najdeme novou aplikaci, kterou označíme a stiskneme tlačítko *Start*.

7.6.3. Metodiky použité při vývoji EJB

Při tvorbě větších aplikací je velkou investicí do budoucna správný návrh architektury a dobré programovací praktiky. Proto je nutné již od samého začátku vývoje dodržovat určité zásady, i když se zprvu zdá, že brzdí rychlý vývoj aplikace. Praxe ukázala, že nejdelsí životnost mají aplikace, u nichž se od začátku předpokládalo, že se v nich budou provádět změny.

V tomto duchu byla navržena i aplikace pro zobrazování údajů z *IS/STAG* ve *WebSphere Portal*. Při jejím vývoji byly dodržovány následující zásady, některé z nich vyšly teprve najevo až během implementace:

- Veškeré zdrojové kódy jsou spravovány systémem *CVS* – *Concurrent Versions System*.

- Vrstva databázového systému:
 - *SQL* dotazy jsou z valné části uloženy v databázi jako pohledy. Přínosem je jednoduchost kódu *EJB*, nezávislost dat a možnost optimalizace dotazů bez zásahu do kódu aplikace.
 - Parametrizované *SQL* dotazy nejsou zásadně vytvářeny skládáním částí řetězců, ale jsou důsledně využívány patřičné funkce *JDBC* rozhraní. Tím se zamezí tzv. *SQL Injection*.
- Vrstva aplikační logiky – *EJB*:
 - Rozhraní komponent jsou důsledně komentována *JavaDoc komentáři*, je tedy snadné kdykoli vygenerovat přehlednou *HTML* dokumentaci.
- Prezentáční vrstva – Portlety:
 - Je důsledně používán model *MVC* – *Model-View-Controller*.
 - Pro chybové stavy je vyhrazen samostatný *View*, který je použit i při neobsložených výjimkách.
- Vazby mezi vrstvami:
 - Přenos dat mezi aplikační a prezentáční vrstvou je realizován univerzálním objektem *DTO* – *Data Transfer Object*, který usnadňuje zapouzdření obecných tabulárních dat. Při změně struktury dat v databázi je tedy nutné provést zásah pouze ve *View* portletu.
 - Ošetření výjimek je realizováno tak, že jsou zapouzdřeny do obecné výjimky aplikace (*Exception Nesting*), které je přiřazen jednoznačný identifikátor. Ten, mimo jiné, umožňuje snadné přiřazení chybového hlášení výjimce a jeho lokalizaci.
- Lokalizace:
 - Na prezentáční vrstvě je realizována tak, že každá portletová aplikace obsahuje tzv. *Resource Bundle* obsahující překlady všech textových informací. Odpovídající *Bundle* je vybrán podle zvoleného národního prostředí – *Locale*.
 - Na aplikační vrstvě je prováděna pouze lokalizace výjimek databáze a *IS/STAG*. Tím je zajištěno, že je nutné udržovat pouze jeden *Resource Bundle*, neboť *EJB* komponenty mohou být využívány více aplikacemi.
 - Na úrovni databáze *IS/STAG* jsou lokalizovány domény a některé důležité položky (např. názvy předmětů, studijních programů). Proto je nutná propagace aktuálního *Locale* až na úroveň *SQL* dotazů.
- Ladění aplikace:
 - Na aplikační a prezentáční vrstvě jsou veškeré výjimky a chybové stavy logovány pomocí *Log4j* – logging.apache.org.
 - Každá výjimka má jednoznačný identifikátor, což umožňuje rychlou lokalizaci vzniklé chyby.
- Přenositelnost:
 - Na aplikační úrovni je zajištěna specifikací *EJB*. Ke každému modulu *EJB* komponent je připojen i deployment deskriptor pro open-source aplikační server *JBoss* – www.jboss.org. Vytvořené komponenty byly v prostředí *JBoss* plně funkční a předpokládá se jejich využití i v dalších organizacích provozujících *IS/STAG*.
 - Na prezentáční vrstvě je zajištěna standardem *JSR-168* známým také jako *Java Portlet API*. Bohužel je toto teprve prvotní verze standardu, která neřeší některé důležité aspekty, jako je například komunikace mezi portlety. Proto byly vytvořené aplikace implementovány nad *IBM Portlet*

API, které komunikaci mezi portlety podporuje. V portletové aplikaci dle JSR-168 lze sice určitým způsobem zrealizovat komunikaci mezi portlety (přes *Application Scope Portlet Session*), je však z programového hlediska těžkopádná a neefektivní. Nicméně byl učiněn úspěšný pokus o provoz JSR-168 portletu vytvořeném v IBM WSAD prostředí i v open-source portálu *uPortal* – *www.uportal.org*. *uPortal* byl spolu s výše zmíněnými EJB komponentami nainstalován do aplikačního serveru *JBoss*.

7.6.4. Přehled vytvořených EJB

V rámci integrace portálu do univerzitního prostředí byla vytvořena sada EJB komponent (a portletů, které je využívají) pro přístup k informacím uložených v informačním systému studijní agendy *IS/STAG*³.

V současné době jsou k dispozici následující EJB komponenty pro *IS/STAG*:

- *Student* – základní informace o studentovi. Zapsané předměty, zkoušky, známky, rozvrh, průběh studia.
- *Ucitel* – základní informace o učiteli. Garantované předměty, vypsané termíny zkoušek, rozvrh, diplomové práce.
- *Predmet* – základní informace o předmětu. Syllabus, vyučující, rozvrh, doporučená literatura, podmiňující a vylučující předměty, počty studentů.
- *RozvrhovaAkce* – základní informace o rozvrhové akci. Předmět, vyučující, zapsaní studenti.
- *Zkousky* – základní informace o zkušebním termínu. Předmět, zkoušející, zapsaní studenti.
- *Katedra* – základní informace o katedře. Vedoucí, Učitelé, předměty, zkušební termíny, rozvrhové akce
- *Mistnost* – základní informace o místnosti. Umístění, provozní doba, kapacita, rozvrh.
- *StudijniProgram* – základní informace o studijním programu. Název, kód, fakulta, forma, obory.
- *StudijniObor* – základní informace o studijním oboru. Název, anotace, platnost, kreditní limity, fakulta, studenti, studijní plány.
- *StudijniPlan* – základní informace o studijním plánu. Název, etapa, kreditní limity, bloky předmětů.
- *BlokPredmetu* – základní informace o bloku předmětů. Název, statut, kreditní limity, předměty.
- *Ciselniky* – přístup k číselníkům/doménám aplikace *IS/STAG*. Státy, kraje, obce, fakulty, pracoviště, časové řady, typy rozvrhových akcí, atd.
- *Kalendar* – přístup ke kalendáři aktuálního akademického roku. Začátek/konec akademického roku/semestru, jednotlivé rozvrhové dny.

7.6.5. Řešení autentizace ve vícevrstvé architektuře

V *J2EE* aplikacích přistupujících do databázových systémů je velmi často využíván *Connection Pool*, neboli datový zdroj nakonfigurovaný zpravidla administrátorem serveru. Mezi jeho výhody patří efektivní sdílení svazku permanentních spojení do databáze komponentami různých aplikací, které tímto nemusí znát žádný login a heslo k databázi. Zároveň je tím umožněna i nezávislost aplikace na konkrétním databázovém produktu.

³IS/STAG je vyvíjen na ZČU v Plzni a využívá jej dalších devět vysokých škol

Nevýhodou ovšem je, že pokud aplikace provádí změnu dat v databázi, musí být datový zdroj nakonfigurován na uživatele s poměrně vysokými právy. Řízení přístupu k datům pak není řešen na úrovni databázového serveru (který disponuje odpovídajícími autorizačními prostředky, jako je *Role-Based* a *Fine-Grained Access*), ale na úrovni aplikační logiky. Dále je tím znemožněn i audit na straně databáze, neboť všechny změny provádí jeden uživatel.

Datový zdroj lze sice nakonfigurovat tak, že lze získat spojení do databáze pro zvoleného uživatele, je však nutné předat i jeho heslo do databáze. To ale bohužel vyžaduje, aby autentizaci prováděla místo portálu aplikace. Tím se vzdáme možnosti *SSO – Single Sign On* a snadného přechodu na jinou autentizační technologii.

Databázový systém *Oracle*, nad kterým je postaven *IS/STAG* nabízí pro řešení těchto problémů tzv. *Middle Tier Proxy Authentication*.

7.6.6. Prezentační vrstva – úvod do portletů

Cesta k portletu vede přes takzvaný *servlet*, který vznikl jako protějšek starého *CGI* rozhraní v jazyce *Java*. Postupně se objevila technologie *JSP – Java Server Pages* umožňující *Java Scripting* v *HTML* stránkách, kde *JSP* stránka je na straně serveru „přeložena“ do *servletu*. Dalším vývojovým stupněm je takzvaný *Portlet* a jeho *Portlet API*, které je podobné dosud používanému *Servlet API*, avšak již přebírá vlastnosti *API* pro vizuální komponenty okenních systémů.

V současné době je *Portlet API* specifikováno jako standardní *Java Portlet API (JSR-168)*. Paralelně s ním však existují *API*, která jsou doplněna o některé vlastnosti, které v *JSR-168* dosud chybí (např. komunikace mezi portlety, jak již bylo zmíněno dříve). Je to především *IBM Portlet API* a některé *open-source API*, které se vesměs snaží napodobit *API IBM*.

Portlet API poskytuje i možnost volání *JSP* stránky a tím podporuje návrh portletu podle modelu *MVC*. *JSP* stránka tak implementuje komponentu *View*.

Podrobný popis *JSP*, *Servlet API* a *Portlet API* je možné najít v příloze B, vývoji portletových aplikací v *IBM WebSphere Application Developer Studio* je věnován tutorial v příloze C.

7.6.7. Přehled vytvořených portletů

Portlety, které byly v rámci tohoto portálového projektu vytvořeny, využívají *IBM Portlet API* pro jeho bohatší funkcionalitu. Podrobný výčet a popis portletů lze najít v katalogu vytvořených portletů v kapitole 10

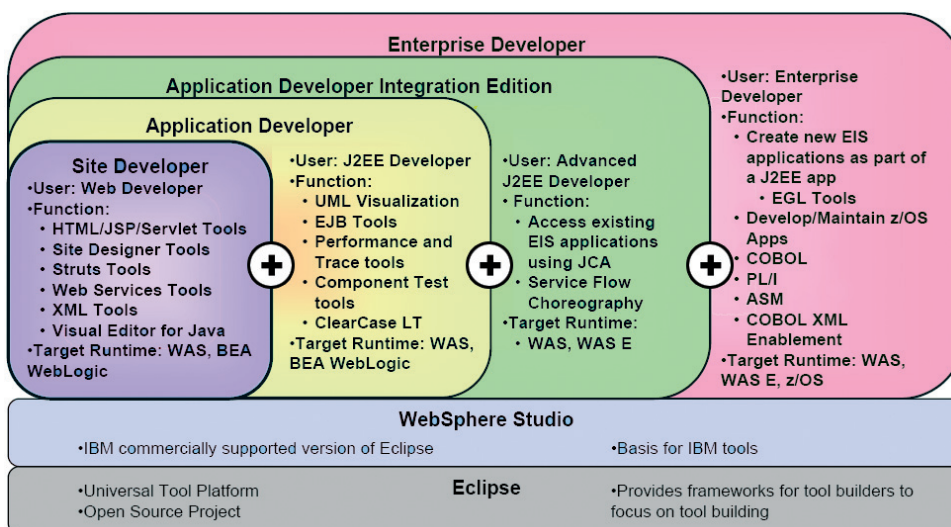
7.7. IBM WebSphere Studio

Rodina produktů *IBM WebSphere studio* je typickou ukázkou komerční varianty *Eclipse*. Dá se říci, že *IBM* upravila a doplnila *Eclipse*, aby jej mohla dodávat jako vlastní vývojové prostředí pro své produkty (zejména *WebSphere Application Server*, *Lotus*, *DB2* a další). Následující obrázek ukazuje rodinu *WebSphere studio* a její vztah k *Eclipse*:

Jednotlivé produkty jsou odstupňované podle zaměřené nástroje a množství zahrnutých funkcí.

- **Site Developer – WSSD**

Nejjednodušší nástroj z rodiny je zamečen na *webdesignera*. Předpokládá se tvorba statických *HTML* stránek nebo *JSP*. *WSSD* umožňuje správu celých *website* (nebo chcete-li projektů), použití šablon a další nástroje podobně jako



Obrázek 7.17: Rodina WebSphere Studio

například Macromedia Dreamweaver. WSSD lze použít i k běžnému programování v jazyce Java (J2SE – Java2 standard edition).

• **Application Developer – WSAD**

S tímto balením WebSphere studia se setkáte nejčastěji. Obsahuje v sobě WSSD plus nástroje pro J2EE (Java2 Enterprise Edition). Nástroj je zaměřen na návrh a vývoj webových nebo obecně klient/server aplikací, vývoj EJB a testování. V rámci WSAD lze spustit několik různých aplikačních serverů a v nich běžící aplikace přímo ladit.

• **Application Developer Integration Edition**

Jedná se o ještě větší verzi WSAD obsahující nástroje pro EAI (enterprise application integration), zejména technologii JCA (Java connector architecture).

• **Enterprise Developer**

Největší balení WebSphere studia kromě jazyka Java podporuje celé portfolio programovacích jazyků (i starších) používaných IBM mj. i na mainframech.

Na ZČU a OSU používáme WSAD s rozšířením Portal Toolkit pro vývoj nové generace IS/STAG a portletů pro univerzitní portál.

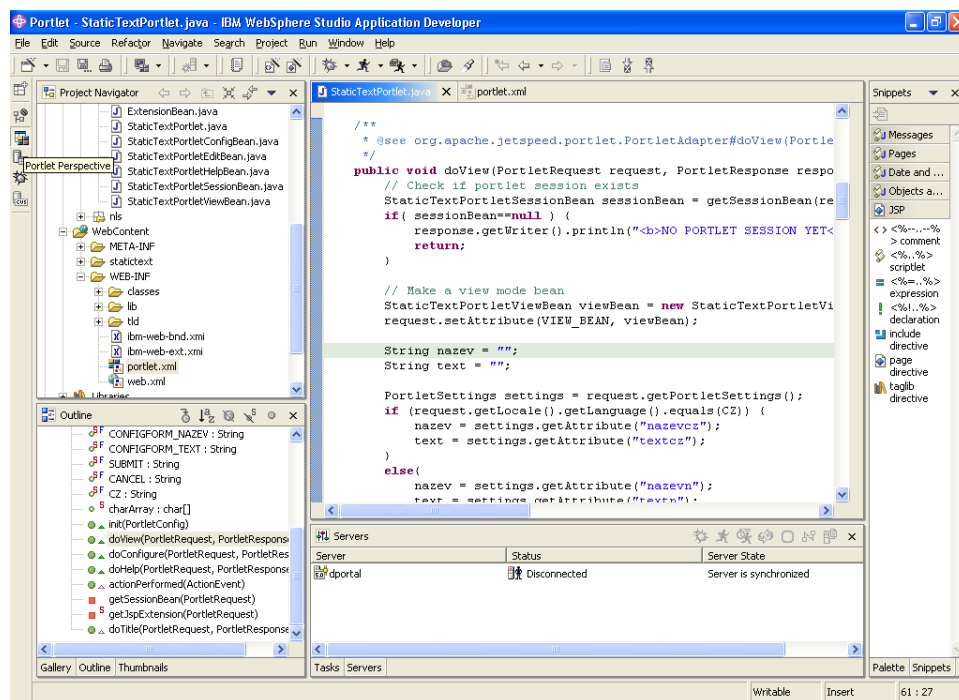
7.7.1. Vývoj portletů ve WebSphere Studiu

Samotné IBM WebSphere Application Developer Studio – WSAD neobsahuje podporu vývoje portletů, proto je nutné do něj nainstalovat *Portal Toolkit*, který výrazně usnadňuje tvorbu portletových aplikací. Pomocí odpovídajících průvodců lze snadno vytvořit portlety odpovídající *IBM Portlet API* i *JSR-168 Portlet API*.

K vývoji prezentovaných aplikací bylo použito WSAD verze 5.1 a *Portal Toolkit* verze 5.0.2. Podrobněji je prostředí WSAD popsáno v tutorialu v příloze C.

7.7.2. Instalace WSAD

WSAD resp. WSSD je součástí instalační sady IBM WebSphere Portal for Multiplatforms Express+. Lze jej nainstalovat pod OS Windows i Linux, přičemž verze pro Windows jsou uvolňovány obvykle dříve.



Obrázek 7.18: Portletová perspektiva WSAD

Po instalaci samotného WSAD je potřeba doinstalovat Portal Toolkit, který volitelně vyžaduje WebSphere Test Environment, do kterého je potřeba zase doinstalovat WAS5 PTF. Principiálně se jedná o speciální verzi aplikačního serveru, který se instaluje přímo do WebSphere Studia. Do tohoto aplikačního serveru se doinstaluje speciální verze portálu.

Jestliže instalace samotného portálu je obtížná, potom instalace Portal Toolkitu je horor: produkty (WSAD) mají naprosto chaotické číslování verzí a díky tomu, že všechny dotčené produkty (WAS, WPS, WSAD, PT) mají mezi verzemi 5.0 a 5.1.x několik variant, je potřeba další desítky záplat, které obvykle nejdou aplikovat.

Celé to zabere asi měsíc času. Sice jsme dosáhli jedné funkční instalace, ale nakonec jsme WSAD používali bez debugování – Portal toolkit umožňuje tzv. *portlet preview*, což je sice pomalé a nepraktické, ale takovou konfiguraci lze poměrně snadno nainstalovat, má-li člověk správná instalační CD.

Problémy s WSAD řešíme přechodem na Rational Application Developer:

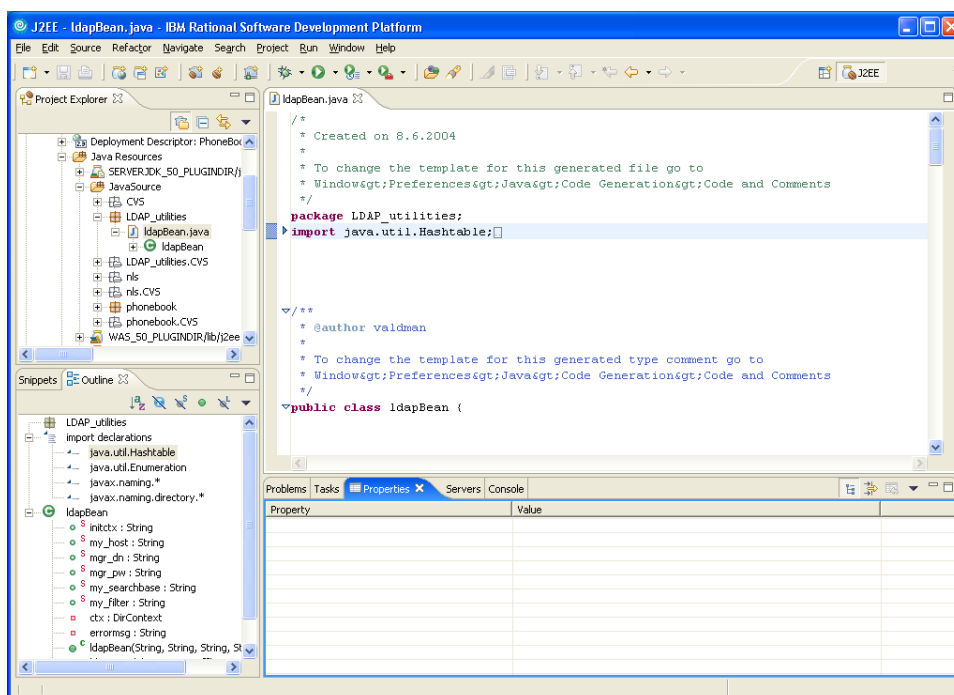
Poznámka

Aby ve WSAD fungovalo `team→share peoject`, musí CVS odpovídat řetzcem `cvs server...` a ne `cvs update...` V debianu to vyřešil downgrade na verzi 1.11.1p1debian, která se chová tímto způsobem.

V Rational Application Developer 6.0 je již toto konfigurovatelné v předvolbách.

7.8. IBM Rational SDP

IBM Rational Software Development Platform (SDP) je nástupcem WebSphere studia; víceméně se jedná o přeskupení produktů v rámci IBM, kde vývojové nástroje nyní patří do divize Rational, která pokrývá celou škálu produktů pro oblast analýzy, návrhu, vývoje, testování a instalace software.



Obrázek 7.19: J2EE perspektiva v Rational Application Developer

RAD 6.0 je součástí Rational SDP, je založen na Eclipse verze 3 a na rozdíl od WSAD v něm mnohem lépe funguje debugování. Lze jej také bez problémů nainstalovat, a proto jsme jej zvolili jako lepší platformu pro další vývoj.

V únoru 2005 jsme zakoupili dvě licence na RAD6.0, nejen abychom získali lepší vývojový nástroj, ale i proto, abychom licenčně pokryli rostoucí počet portálových vývojářů.

Tento nástroj plánujeme použít i pro výuku informatiky, kde bychom jej v rámci IBM Scholars Agreement měli získat zdarma.

7.9. Nastavení JDBC Datasource

Je velmi vhodné vytvářet portlety tak, aby se k databázi připojovaly pomocí JDBC datasource, a bylo tak možné řídit počet spojení, které portál otevírá nad databází.

Připojení portálových komponent k databázi je realizováno přes rozhraní *JDBC – Java DataBase Connectivity*. Toto rozhraní umožňuje i vytvoření tzv. datových zdrojů, které jsou J2EE kontejnerem poskytovány jako služba, kterou je možné vyhledat prostřednictvím *JNDI – Java Naming and Directory Interface*. Výhodou tohoto přístupu je, že komponenty neznají jméno a heslo uživatele, pod kterým se připojují k databázi a kontejner provádí tzv. *pooling* spojení. To znamená, že je předem připravena množina permanentních spojení, která jsou sdílena komponentami. Cílem této kapitoly je popsat, jakým způsobem lze v *IBM WebSphere Application Server* nakonfigurovat takovýto datový zdroj.

Konfiguraci datového zdroje provedeme pomocí *WebSphere Application Server Administrative Console* ve dvou základních krocích – vytvoříme nejprve tzv. *JDBC Provider*, který představuje driver ke konkrétní databázové platformě a potom vytvoříme datový zdroj, využívající tento *JDBC Provider*. Chceme-li tedy vytvořit datový zdroj do *Oracle* databáze, postupujeme podle následujícího návodu:

1. Přihlásíme se jako administrátor a v levém menu vebereme *Resources* → *JDBC Providers*
2. Přidáme nový *JDBC Provider* stiskem tlačítka *New*, kde dále zvolíme typ *User-defined JDBC Provider*, který nakonfigurujeme následovně:

- *Name* – Oracle JDBC Provider
- *Description* – Oracle 10g Instant Client JDBC Driver
- *Classpath* – /oracle/lib/ojdbc14.jar
- *NativeLibraryPath* – není nutno v případě použití JDBC Thin klienta
- *ImplementationClassName* – oracle.jdbc.pool.~>
OracleConnectionPoolDataSource

a stiskneme tlačítko *Apply*

3. Nyní můžeme již vytvořit vlastní datový zdroj využívající tento *JDBC Provider*:

(a) V sekci *Additional Properties* zvolíme *Data Sources* (Pozor – ne *Data Source (version 4)*), stiskneme tlačítko *New* a vybereme *Configuration*, kde nastavíme tyto parametry:

- *Name* – IS_STAG
- *JNDI Name* – jdbc/stag
- *CMP* – No
- *Description* – komentář k datovému zdroji
- *Category* – datový zdroj je možno zařadit do určité kategorie
- *Statement Cache Size* – můžeme ponechat nabídnutou hodnotu
- *Data Source Helper Classname* – com.ibm.websphere.~>
rsadapter.OracleDataStoreHelper
- *Component-managed Authentication Alias* – zatím nevybereme nic – viz dále bod (c)
- *Container-managed Authentication Alias* – zatím nevybereme nic – viz dále bod (c)

a stiskneme tlačítko *New*.

(b) Ve spodní části administrační konzole v *Additional Properties* vybereme *Custom Properties*, parametr *URL* nastavíme na hodnotu `jdbc:oracle:thin:@server.domena.cz:1526:databaze` a stiskneme tlačítko *Apply*

(c) Vrátime k předchozímu formuláři, v sekci *Related Items* zvolíme *J2C Authentication Data Entries*, stiskneme tlačítko *New*. Zadáme následující data:

- *Alias* – IS_STAG
- *User ID* – ora_schema
- *Password* – heslo
- *Description* – nějaký komentář

a stiskneme tlačítko *Apply*. Tyto autentizační údaje je možné později změnit v oddíle *Security* → *JAAS Configuration* → *J2C Authentication Data*. Nyní můžeme nastavit parametry *Component-managed Authentication Alias* a *Container-managed Authentication Alias* na hodnotu `dportal/IS_STAG`, které jsme v bodě (a) nenastavili a stiskneme *Apply*.

4. Změnu konfigurace uložíme stiskem tlačítka *Save* a restartujeme portál.
5. Nakonec můžeme nově vytvořený datový zdroj otestovat. V administrační konzoli postupně zvolíme: *Resources* → *JDBC Providers* → *Oracle JDBC Provider* → *Datasources* → *IS-STAG* a stiskneme tlačítko *Test Connection*.

Část II

Řešení projektu PORT@L

Řešení projektu

Tato kapitola popisuje postup řešení projektu, čerpání finančních prostředků a další náležitosti vyžadované zadáním projektu.

Obsah

8.1. Postup řešení	107
8.1.1. Pořízení a instalace WPS	108
8.1.2. Vývoj aplikací	108
8.1.3. Získávání know-how	109
8.1.4. Pilotní projekty	109
8.2. Adresář řešitelů	110
8.3. Změny v projektu	110
8.4. Čerpání finančních prostředků	110
8.4.1. Investiční prostředky	110
8.4.2. Neinvestiční prostředky	111
8.5. Spolupráce s IBM	111
8.6. Hledání pomoci u partnerů IBM	112
8.7. Diskuse splnění cílů projektu	113
8.8. Shrnutí	115
8.8.1. Udělali jsme	115
8.8.2. Nestihli jsme	115
8.8.3. Podařilo se	116
8.8.4. Nepovedlo se	116
8.8.5. „Lessons learned“	116
8.9. Další využití výsledků projektu	116

Tato kapitola je věnována postupu řešení projektu PORT@L. V úvodní části popisuje jak byl získán WebSphere Portal, jeho instalaci, postupné řešení technických problémů a získávání potřebných znalostí a současný stav pilotních projektů. Dále popisuje podporu ze strany firmy IBM a jejích partnerů, změny v projektu, ke kterým během jeho realizaci došlo a v neposlední řadě i finanční stránku projektu. V závěru je diskutováno splnění vytyčených cílů.

8.1. Postup řešení

Při řešení projektu jsme vycházeli z našich předchozích zkušeností a doporučené metodiky. V mezích personálních možností jsme se snažili práce maximálně paralelizovat jejich rozdělením do víceméně disjunktních oblastí:

1. instalace produktu a napojení na IT infrastrukturu
2. vývoj portletů
3. nastavení a správa portálu

Bylo nám jasné, že velké množství času spolyká napojení portálu na IT infrastrukturu a stejně tak i vývoj portletů. Jelikož tyto akce na sobě nejsou zcela závislé (vývoj programového vybavení probíhal v laboratorním prostředí na testovací instalaci portálu), snažili jsme se již od začátku specializovat takto:

- Jan Valdman – instalace, správa portálu, webdesign¹
- Maxmilán Otta – DBA, vývoj EBJ a integrace IS/STAG
- Jiří Šimonek – instalace, vývoj portletů
- Kateřina Rašková – integrace SAP a obecně EAI

8.1.1. Pořízení a instalace WPS

Vzhledem k tomu, že jsme již měli zapůjčené licence na WPS od IBM, nebylo jejich zakoupení klíčové pro odstartování projektu; podobně to bylo i se servery – pro testování stačily výkonnější laboratorní PC. Licence na WPS i servery jsme tedy zakoupili až v průběhu léta 2004.

Mnoho času nás stála vlastní instalace WPS a jeho napojení na IT infrastrukturu školy. Nejlépe na tom byla OSU, jejich *Novell eDirectory* byl přímo podporován a navíc tato škola má centrální registr osob. V případě ZČU jsme narazili na problém s *OpenLDAP* serverem, které se nám podařilo definitivně vyřešit až v lednu 2005. Do té doby byl portál přístupný pouze několika stovkám uživatelů. S instalací portálu na UPOL musela kvůli výměně tamního spoliřešitele pomoci OSU, nicméně se již nepodařilo jej napojit na místní infrastrukturu. UPOL používá starou verzi *Novell* bez *LDAP Directory* a jejich autentizační infrastrukturu na bázi *TACACS* serverů se nepodařilo propojit s portálem.

Při řešení našich infrastrukturálních problémů jsem hledali pomoc u IBM a jejich partnerů (viz strana 8.6.), ale víceméně marně: Support k produktu od IBM slouží vlastně jen k hlášení chyb v produktu a získání aktualizací. IBM neoficiálně říká, že její starostí není, aby zákazník uměl produkt používat, k tomu se má obracet na její partnery. V nejtěžších chvílích nám pomohl pan Dušan Smolej z *IBM Software Services*, který nám poskytl množství neveřejných materiálů a cenné rady.

8.1.2. Vývoj aplikací

Vývoj *Enterprise Java* aplikací a portletů byla pro nás úplně nová věc. Problematika je poměrně obsáhlá a bez předchozí znalosti programování webových Java aplikací se do ní těžko dostává. Díky jedné diplomové práci obhájené v létě 2004 na ZČU jsme však překonali prvotní obtíže a navíc získali i dobrý návod v českém jazyce, který využíváme pro zapojení dalších vývojářů.

EJB vyvíjíme pomocí open-source aplikačním serveru *JBoss* a *IBM WSAD5.0* dodávaném s portálem, v únoru 2005 jsme zakoupili jeho nástupce *IBM RAD6.0*, ve kterém je podstatně lepší podpora ladění. Použití aplikačního serveru *JBoss* totiž značně urychlilo vývojový cyklus díky své velmi pohodlné *hot deployment* technologii, narozdíl od *WebSphere Portal*, kde je nutné provést *EJB deployment* v několika časově náročných krocích.

Vývoj portletů se ukázal jako velmi náročný na čas a po celou dobu projektu zápolíme s nedostatkem vývojářů. Personální situace na VŠ rovněž nepřispívá k ochotě

¹krom toho administrace projektu, průzkum problematiky do šířky, komunikace s IBM a jejími partnery

vedoucích pracovníků investovat do dlouhodobých projektů typu portál, když mají problémy personálně pokrýt běžné činnosti.

Vývojem aplikací se zabývá celá kapitola 7, v rámci diplomové práce vytvořený tutorial vývoje portletů je v příloze C.

8.1.3. Získávání know-how

Po celou dobu projektu jsme se snažili zvyšovat svoji kvalifikaci a znalosti WPS. Studovali jsme samozřejmě dokumentaci k produktu (*IBM InfoCenter*, která je sice nadměru rozsáhlá (*IBM RedBooks* mají obvykle 500 až 1000 stránek), ale na příliš mnoho otázek v ní chybí odpovědi.

Problémy v portálu obvykle nelze řešit bez podrobné znalosti a pochopení jeho struktury a funkce jednotlivých komponent. Z předchozí zkušenosti jsme věděli, že nejlepší zdroj informací jsou portálové konference IBM, a proto jsme se zúčastnili technické konference ve Stuttgartu.

Jelikož jsou konference IBM poměrně nákladné, a také proto, že se omezují pouze na technická témata, snažili jsme se navštívit další portálové konference, pokud možno se zaměřením na akademické prostředí, abychom se inspirovali a vytěžili z portálu maximální užitek. Do této kategorie spadají konference PEPC, Euro-pen.CZ, UNINFOS a setkání portálové skupiny CESNET.

Dále jsme se snažili získat školení v oblasti správy portálu a vývoje portletů. Kupodivu se to ukázalo jako velký problém. IBM nám slibovala nabídnout nějaké školení od února 2004. Nakonec jsme až v lednu 2005 sehnali první firmu splňující naše očekávání – *Cleverbee*.

Informace jsme hledali i v odborné literatuře. Spoléhalí jsme na americké vydavatele a *IBM Press*, protože česká portálová literatura neexistuje. Problémem všech knih (kromě *IBM RedBooks*) je minimálně roční zpoždění za uváděnými produkty; na začátku roku 2004 se dostupné knihy zabývají z našeho pohledu starou verzí portálu 4.2.

Z dřívějšíka jsme věděli o knize [10], ke které jsme ještě objednali [11] a *Red-Book* [6]. Po obdržení těchto knih a vyhodnocení jejich informačního přínosu jsme objednali další výtisky [10] (každá univerzita jeden výtisk) a za zbylou částku cca 3 tisíce Kč zakoupili 9 českých knih zabývajících se související tematikou. Průběžně jsme sledovali informace v diskusní skupině (newsgroup) *com.ibm.software.websphere.portal-server* a e-ziny *PortalsMonthly*, *BitPipe*, *TheServerSide.com* a jiné.

Až 26. ledna 2005, tedy ke konci grantu, jsme při jednání s IBM opět nadhodili otázku školení,² a odpovědí nám bylo tentokrát doporučení podpsat *IBM Scholars Agreement*. To ZČU neprodleně učinila a 15. února byla naše žádost schválena. Díky tomu se nám otevřela cesta k mnoha materiálům, mj. [12, 13, 14] a [15]. Mít tyto materiály o rok dříve, mohlo být mnoho věcí jinak.

8.1.4. Pilotní projekty

Na konci roku 2004 na ZČU odstartoval poměrně rozsáhlý pilotní projekt, do kterého byly pod vedením dvou prorektorů zapojeny dvě testovací katedry (KIV a KMA, viz příloha A). Jeho cílem je plný provoz portálu od akademického roku 2005/2006.

Na OSU se portál začal postupně využívat již od podzimu 2004 tak, jak do něj postupně byly integrovány aplikace, resp. nové aplikace jsou vytvářené pouze pro portál, takže uživatelé nemají možnost portálu uniknout. Podrobnější informace o vytvořených portletech jsou v kapitole 9.

² které pro nás již skoro rok slibovali zařadit

8.2. Adresář řešitelů

Vybudování portálu je náročný projekt, který nelze řešit ve dvou lidech, nýbrž je potřeba vytvořit v každé organizaci tzv. horizontální tým technických specialistů (operační systémy, databáze, adresářové služby, bezpečnost, Java programátoři, webdesignerů a webových programátorů) i manažerů (prorektorů, vedoucích IT oddělení, projektových manažerů).

Toto se nám do jisté míry podařilo (viz příloha A.1.), nicméně mnoho z uvedených osob sice portál podporuje, ale „nepřiloží ruku k dílu“. Jinými slovy, i v České Republice platí britská zkušenost, že na průměrné univerzitě dělají portál dva až tři lidé, kteří za dva až tři roky dovedou projekt do pilotního provozu. Teprve potom je šance, že se pohnou ledy.

Z pohledu tohoto grantu lze tedy na jeho řešitele nahlížet jako na koordinátory širších týmů na jednotlivých univerzitách.

8.3. Změny v projektu

V projektu nastaly celkem tři změny, z toho o dvě jsme sami požádali:

1. 5.května 2005

Bylo požádáno o změnu projektu: místo tří licencí na WebSphere portál budou pořízeny pouze dvě, protože ZČU si již licenci pořídila v době schvalování grantu a navíc mezitím došlo k výraznému zdražení produktu (v důsledku změny kursu Eura), takže by na třetí licenci chybělo cca 410 tisíc Kč.

Dále došlo z důvodu personálních změn na UPOL k výměně spoluřešitele projektu - místo Mgr. Stěpána Stěpanova bude od května 2004 na projektu pracovat Mgr. Kateřina Rašková.

2. 9.června 2004

Rada Fondu rozvoje v dopise z 9. června umožnila použít dodatečnou částku 32 tisíc Kč (původně určenou na sociální a zdravotní pojištění) na „jiný účel v rámci projektu“.

3. 4.února 2005

Bylo požádáno o možnost použít finanční prostředky z kapitoly *cestovné zahraniční* na pokrytí nákladů v kapitole *cestovné domácí* a to do výše 40 tisíc Kč. Dále bylo navrženo použít částku 32 tisíc Kč na odměny řešitelům. K okamžiku uzavírání projektu jsme zatím nedostali k této žádosti od Rady FR CESNET žádné oficiální vyjádření.

8.4. Čerpání finančních prostředků

Na tomto místě podáváme stručný komentář čerpání finančních prostředků dle jednotlivých kapitol grantu.

8.4.1. Investiční prostředky

Byly zakoupeny dvě procesorové licence *IBM WebSphere Portal Express+* verze 5.0 v celkové výši 1221 tis. Kč. V souladu se změnou č. 1 vložila ZČU do projektu vlastní licenci na WPS v pořizovací ceně 446 tisíc Kč.

Nad rámec původního plánu byl zakoupen portlet *Cleverbee FileExplorer*³ za cenu

³nezbytná náhrada za nepoužitelný PDM

76 tis. Kč a dvě licence *IBM Rational Application Developer*⁴ za 102 tis. Kč.

Pro potřeby provozu portálu a související infrastruktury byly pořízeny dostatečně dimenzované PC servery:

- 1 × *Dell PowerEdge 1750* za celkem 148 tis Kč (ZČU),
- 2 × *IBM xSeries 345* za celkem 300 tis Kč a
- 1 × *IBM xSeries 365* za 250 tis Kč (OSU) a
- 1 × *IBM xSeries 255R* za 391 tis Kč (UPOL).

Jako naši spoluúčast (v investicích) deklaruujeme jednu licenci WPS a tři PC servery v úhrnné výši 1,137 mil. Kč.

8.4.2. Neinvestiční prostředky

Náklady na účast na domácích konferencích (setkání CPG, European) a na vzájemné pracovní schůzky v Plzni (3×), Ostravě (2×) a Olomouci (2×) se vyšplhaly do výše 46 tis. Kč. Náklady na zahraniční konference (IBM Stuttgart, PEPC2004, Strassburg) dosáhly výše 250 tis. Kč.⁵

Zprávy z konferencí přikládáme v příloze E.

Na odborná školení byla použita částka 48 tis. Kč – jednalo se o jednodenní školení v oblasti WPAI a dvoudenní školení v oblasti pokročilého vývoje Java aplikací. Obě školení byla objednána od firmy *Cleverbee* a zúčastnili se jich kromě řešitelů další osoby z portálových týmů univerzit.

Za částku 10 tisíc Kč bylo zakoupeno 14 odborných knih, z toho 3 zahraniční tituly [10, 11, 6].

Na projektu řešitelé podíleli v přepočtu cca následujícími úvazky: Valdman – 0.8, Otta – 0.6, Šimonek – 0.75, Rašková – 0.3; v součtu tedy cca 2.45 úvazku měsíčně. Mzdové náklady lze tedy při průměrném platu 20 tis. Kč vyčísřit na cca 588 tis. Kč. Práci dalších osob podílejících se na projektu neuvažujeme.

Částka 25 tisíc Kč na osttání služby a dodatečně přidělená částka 32 tisíc Kč nebyla čerpána. V žádosti o změnu projektu č.2 navrhujeme použít částku 32 tis. Kč jako odměny pro řešitele.

8.5. Spolupráce s IBM

IBM je v mnohém zvláštní firma, která se chová nepředvídatelně. Na jednu stranu vám umožní zdarma přístup k obrovskému množství informací a software, na druhou stranu je neřešitelný problém řádně využívat (zaplacený) support. České zastoupení IBM často hovoří o firmách ve firmě, což je vzhledem k celosvětovému počtu divizí a zaměstnanců asi pravda. Každopádně je potřeba počítat s tím, že lidé od produktu si myslí a říkají něco jiného než obchodníci, ti zase něco jiného než lidé ze software services a každý vývojář portálu nebo expert z laboratoří navíc do problému vnese svůj naprosto originální názor.

Během prací na portálu jsme přišli do styku s mnoha lidmi od IBM – obchodníky, produktovými specialisty, konzultanty, lidmi z IBM laboratoří, vývojáři portálu z USA i Německa. Častým jevem bohužel bylo, že jsme od různých lidí dostávali naprosto protichůdné informace (namátkou například licenční politika, podpora

⁴funkčního debugování je nezbytné pro efektivní vývoj portletů; doplněk WSAD

⁵V okamžiku ukončení grantu ještě není potvrzeno schválení žádosti o změnu č. 3; v případě jejího neschválení navrhujeme schodek domácího cestovného dorovnat z kapitoly *Ostatní služby*.

Kerberos, podpora JSR-168, Content Management, SSO/LTPA Token, implementace vlastního User Registry, implementace TAI Modulů).

Obchodně IBM funguje podle modelu *Partner Business*. To znamená, že kromě vlastní divize *IBM Global Services - IGS* a laboratoří ve firmě prakticky (z našeho pohledu, alespoň v ČR) nejsou lidé, kteří mají detailní know-how a praktické zkušenosti s provozováním produktu. Tato oblast je z velké části ponechána na partnerech, kteří ale bohužel v oblasti WebSphere buď sami také nemají potřebné know-how nebo jsou jejich kapacity alokovány na několik měsíců dopředu. Prý je to způsobeno tím, že v ČR tento model fungoval zejména pro Lotus produkty, kde bylo více obchodních případů kratšího trvání. Naopak WebSphere vyžaduje dlouhodobé kontrakty.

V rámci projektu se ZČU pokoušela přihlásit do programu *IBM Partner World*, zatím bezúspěšně. Prý jsme se nekvalifikovali jako *Independent Software Vendor - ISV*. V současné době (únor 2005) probíhá zpracování našeho odvolání.

V souvislosti s tímto projektem ZČU podepsala smlouvu *IBM Scholars Agreement*, která dává možnost zdarma přistupovat k většině software IBM a dalším studijním materiálům, ovšem pouze pro studijní účely. V této věci rozvíjíme spolupráci s katedrou informatiky (KIV ZČU).

8.6. Hledání pomoci u partnerů IBM

Prakticky celou dobu řešení projektu jsme se snažili najít obchodního partnera IBM, který by měl dostatečné know-how a kapacitu, aby nám pomohl vyřešit některé problémy.⁶

Přišli jsme do styku s těmito společnostmi:

- **Corpus Solution**

První firma, se kterou jsme jednali o možné spolupráci. Přestože to z počátku vypadalo slibně, nakonec z nejasných důvodů ke spolupráci nedošlo; část „viny“ je asi i na naší straně.

- **Ability Development**, nyní **ADT**

Spolupráce s firmou ADT vyústila v nabídku poradenských služeb jejich HelpDesku – služba Spektrum. V polovině listopadu jsme dostali na vyzkoušení 2 hodiny jejich služeb, nějaký čas jsme hledali problém vhodné velikosti na vyzkoušení, a když jsme konečně v polovině ledna odeslali dva dotazy, nedostali jsme žádnou odpověď.

Přesto nadále uvažujeme o využití služeb této firmy, protože nám jako jediná nabídla možnost obracet se na ně s drobnými dotazy.

- **Groupsoft**

S firmou Groupsoft jsme jednali o řešení konkrétních našich problémů v oblasti napojení portálu na infrastrukturu ZČU – viz nabídka na straně 219.

Domluvili jsme se na řešení problému týkajícího se *memberOf* atributu v *OpenLDAP*. Od konce listopadu do poloviny ledna firma ani po urgencích nevyvíjela žádnou aktivitu, a tak jsme po uvedení opravného balíčku *PQ980004* spolupráci ukončili.

- **Cleverbee**

Firmu Cleverbee nám doporučila firma Groupsoft jako svého subdodavatele v otázkách vývoje aplikací. Jedná se o česko-švýcarský softwarehouse, který má s vývojem Java aplikací a IBM portálem poměrně velké zkušenosti.

Od této firmy jsme koupili několik dnů školení a portlet *FileExplorer* (viz strana 73) a nadále hodláme pokračovat ve spolupráci, přestože jejich služby

⁶výhodami partnerských firem jsou: znalost komunikace s IBM, osobní kontakty, možnost zprostředkovaně komunikovat s laboratořemi IBM, informace dostupné z *PartnerWorld*.

se týkají pouze oblasti vývoje aplikací. V administraci portálu máme asi větší zkušenosti sami.

- **Your System**

Jedná se o českou firmu, která je partnerem IBM v Ostravě. Je zástupcem pro Moravskoslezský kraj v oblasti distribuce software i hardware.

Ostravská univerzita od této firmy zakoupila několik serverů (2 ks *IBM Server xSeries 345* a 2 ks *IBM Server xSeries 365*), na kterých provozujeme portál.

Dále je firma Your System kontaktem pro Ostravskou univerzitu v záležitostech poskytování produktů firmy IBM k testování. Z hlediska konzultací je ovšem firma Your System nepoužitelná, naše zkušenosti a znalosti v oblasti portálu vysoce převyšují znalosti pracovníků této firmy.

8.7. Diskuse splnění cílů projektu

Projekt PORT@L měl několik konkrétních cílů, jejichž splnění diskutujeme v následujícím textu.

- *Shromáždit finanční prostředky na zakoupení licencí aplikačního serveru a portálu IBM WebSphere Portal Express+ i potřebného hardware.*

Tohoto cíle bylo bezpochyby dosaženo, všechny tři univerzity mají licenci na WebSphere Portal a potřebný HW. Nad rámec grantu bylo potřeba samozřejmě vytvořit i testovací portálové prostředí a investovat do výkonných pracovních PC stanic pro vývojáře.

- *Instalovat a rutinně zprovoznit AS/portál na všech univerzitách zapojených do tohoto projektu a zpřístupnit několik pilotních aplikací skrze portál.*

Tohoto cíle bylo dosaženo jen částečně:

Na Západočeské univerzitě je portál provozován v rámci pilotního provozu, který skončí v červnu 2005. Opravdu ostrý provoz plánujeme až od příštího akademického roku (tj. od září), nicméně i v současné pilotní fázi portál používá několik stovek uživatelů. Z provozního pohledu je portál zprovozněn rutinně, z uživatelského pohledu je toto označení diskutabilní.

Na Ostravské univerzitě je portál (ve smyslu aplikačního frameworku) používán opravdu rutinně, avšak nejsou plně využity jeho "portálové" schopnosti. Z provozního pohledu je portál v rutinním provozu.

Na Univerzitě Palackého je portál zatím pouze v testovacím provozu, hlavními problémy je neexistence infrastruktury LDAP a nevyjasněná situace ohledně tzv. internetových licencí SAP.

- *Každá z jmenovaných univerzit vytvoří několik pilotních aplikací, jejichž smyslem je ověření vývojového cyklu i funkčnosti nově instalovaných technologií, resp. jejich napojení na stávající infrastrukturu. Technicky se bude jednat o aplikace menšího rozsahu s poměrně jednoduchou aplikační logikou, ale které se budou napojovat např. do databází Oracle, univerzitních adresářů s vazbou na jednotnou autentizaci, zpřístupňovat stávající PHP aplikace skrze portál apod. Tyto pilotní projekty jsou již rozpracovány, na ZČU se jedná například o nový telefonní seznam.*

Na ZČU a OSU bylo vytvořeno množství portletů (viz kapitola 10) programátorskou cestou (za využití IBM API), které jsou napojeny na enterprise databáze a univerzitní adresářové služby. Jedná se o drobné aplikace (tel. seznamy, ISIC karty) i velké projekty (namátkou IS/STAG, KNOS, Akreditace OSU)

UPOL se soustředila na integraci SAP pomocí několika různých technologií (viz kapitola 7.4.), bohužel ale jen s částečnými úspěchy.

- *Propojit IBM WebSphere Portal (resp. aplikační server IBM WebSphere) se stávající autentizační infrastrukturou (Kerberos nebo PKI) a výpočetním prostředím*

zapojených univerzit jako takovým (např. vazby na registry uživatelů, databázové a adresářové služby apod.)

Na ZČU jsme původně zamýšleli napojit WebSphere Portal přímo na Kerberos, nicméně během řešení projektu se ukázalo (v diskusích s lidmi z IBM laboratoří), že to není jednoduše realizovatelné. Asi bychom dokázali napojit na Kerberos WAS, ale již nikoli WPS. Každopádně se jedná o prakticky nedokumentovanou záležitost „out of support“, kterou z pohledu IBM nemá dělat zákazník ale buď partner nebo přímo *IBM Global Service – IGS*. Nakonec jsem se tedy spokojil s řešením, kdy je použit *OpenLDAP* server s autentizačním backendem do systému *Kerberos*. Řešení si sice není ideově čisté, ale je plně funkční.

Na OSU je WebSphere Portal napojen přímo na výpočetní prostředí Novell prostřednictvím *Novell eDirectory*, což je asi ideální stav.

Napojení na databáze Oracle funguje bez problémů na všech třech univerzitách pomocí JDBC ovladače od Oracle.

K testování PKI jsme se z časových důvodů nedostali, ale tento cíl vidíme v horizontu několika měsíců jako pravděpodobně dosažitelný.⁷

- *Získat teoretické i praktické zkušenosti s vývojem portletů a enterprise Java aplikací (zejména standardy J2EE a Web Services); navrhnout metodiku pro převod dalších aplikací do portálu.*

Tohoto cíle bylo na ZČU i OSU plně splněno (vývoj portálových aplikací v jazyce Java i pomocí *JDBC Portlet Builderu*). Jako velmi obtížné se na univerzitách ukázalo shánění Java vývojářů. Převod aplikační logiky několika IS a vytváření portletů je náročné na časové a lidské zdroje. Univerzity naší velikosti by bez problémů vytižily 10–20 vývojářů, ale k dispozici jich je obvykle pouze 1–5.

Možnosti integrace stávajících aplikací (*Legacy Applications*) do portálu jsou diskutovány v kapitole 7.3.

- *Využití prezentačních možností portálu a vestavěné správy obsahu pro zlepšení kvality intranetových WWW prezentací.*

Na tento aspekt se soustředila hlavně ZČU. Ukázalo se, že skládání informačních webů kateder z prefabrikovaných stavebních kamenů je dobrý princip. Uživatelé (editoři) jsou spokojeni, že si hravým způsobem své stránky rychle „naklikají“ bez velké námahy. Z estetického pohledu portál pomáhá formovat corporate identity webových stránek a zabraňuje amatérismu, základním chybám v návrhu a přehnaným grafickým kreacím.

Ve věci správy obsahu jsme byli schopnostmi WebSphere Portal zklamáni. Správce dokumentů PDM (*Portal Document Manager*) se ukázal jako nepoužitelný (funkčně a výkonově), ale podařilo se nám jej částečně nahradit zakoupením portletu *FileSystemExplorer* od firmy *Cleverbee*. Portlet popisujeme na straně 73.

Portál nemá integrovaný CMS systém s uživatelským rozhraním, je potřeba použít poměrně složitou technologii WPCP, která se skládá z runtime části (v portálu) a authoring části (samostatná webová aplikace instalovaná společně s portálem, ale jen pod Windows). Celé to pak funguje na bázi transformace obsahu (*Content*) do Java tříd, jenž jsou potom volány portlety. K využití této technologie jsme se z časových důvodů nedostali, což je možná nakonec dobře. WPCP nebylo příliš povedené a ve verzi portálu 5.1 IBM po akvizici firmy *Aprix* celý content management v portálu přepracovala – jsou použity komponenty z *Lotus Workplace*.

Ani jedna ze zúčastněných škol zatím neplánuje převést své oficiální weby do portálu, tj. zatím držíme informace (intranet) oddělené od prezentace školy.

⁷Technicky by se mělo jednat o změnu autentizačního modelu webové aplikace WebSphere Portal z “form based” na “certificate” a doladění tzv. login procedury v portálu, což nemusí proběhnout bez komplikací.

8.8. Shrnutí

Celkově hodnotíme dosažené výsledky projektu PORT@L kladně v tom směru, že se podařilo zprovoznit poměrně komplexního software a integrovat jej do stávajícího prostředí univerzit, zvládnout nové portálové technologie a technologie *J2EE* prostředí a vyškolit v nich několik zaměstnanců i studentů, kteří se tak mohli podílet na implementaci vytvořených portálových komponent. Velikým přínosem je i velmi úzká spolupráce participujících univerzit (zejména mezi ZČU a OSU) a výměna vytvořených portálových komponent. Silně motivujícím prvkem byla i skutečnost, že komponenty vytvořené jednou univerzitou fungovaly takřikajíc „na první zapojení“ i v portálu univerzity druhé.

Na závěr ještě stručně shrneme dosažené výsledky projektu:

8.8.1. Udělali jsme

Na ZČU a OSU byly nainstalovány portály, napojeny na IT infrastrukturu a uvedeny do pilotního provozu.

Vytvořili jsme množství portletů (viz kapitola 10), které pokrývají širokou škálu funkcí – od drobností až po integraci velkých aplikací typu IS/STAG nebo KNOS.

Získali jsme značné know-how v oblasti *J2EE* a portálu IBM. Zejména v oblasti samotného produktu WebSphere Portal jsou naše zkušenosti minimálně srovnatelné se specializovanými komerčními firmami (ne-li lepší).

Podíleli jsme se na vytvoření českého návodu (tutorialu) jak vytvářet portlety. Dále jsme v této zprávě přehledně analyzovali problematiku integrace aplikací a správy portálu.

8.8.2. Nestihli jsme

Vymezení portálu v organizaci je obtížné, resp. těžko se hledá, kde končí portál a začínají jiné systémy. Projekt sice neměl přesně definované konkrétní cíle, ale z pohledu naší prvotní všeobjímající a velmi optimistické vize jsme se nestihli věnovat následujícím oblastem:

- integrace *Lotus Notes* (ukázala se netriviální, pokud organizace nemá správnou verzi *Domino LDAP Server*)
- využití *Lotus Collaborative Components* (viz LN, netriviální, problém SSO, nezbyl čas)
- začít používat *Web Services* (nezbyl čas, zvládnutí vývoje „obyčejných“ portletů bylo náročnější než původní odhad, integraci jsme zatím prováděli hlavně na datové vrstvě)
- implementace TAI modulů pro *Kerberos SSO* (ukázalo se jako velmi obtížné, řešili jsme naléhavější problémy)
- vyzkoušet front-side SSO přes PKI (dtto)
- integrace *MIS Verso* (odloženo do vyjasnění formy koexistence MIS a portálu)
- integrace stávajících PHP aplikací jinak než přes *web clipping* (vyžaduje dořešit front-side SSO, řešením by mohlo být JSR-223)
- vyzkoušet technologii WPCP (nezbyl čas, WPCP je neperspektivní a WPS 5.1 používá jinou technologii)
- vyzkoušet technologii *Personalization* (nezbyl čas, malý význam pro akademické prostředí)

8.8.3. Podařilo se

Povedlo se nám proniknout do složitého světa J2EE portálů a aplikačních serverů. Podařilo se nám překonat technické obtíže a dovést dva ze tří portálů do pilotního provozu a dosáhnout jejich akceptace uživateli.

Potvrdili jsme, že vysoké školy mohou spolupracovat na velkých projektech a že portletové aplikace lze opravdu snadno přenášet z jedné instituce do jiné. Díky J2EE architektuře je napojení na místní prostředí velmi jednoduché.

Získali jsme potřebné know-how v oblasti administrace portálu a vývoje portletů. Na ZČU a OSU do portálového projektu postupně zapojujeme další pracoviště (zejména katedru výpočetní techniky) včetně studentů.

ZČU podepsala z podnětu tohoto projektu *IBM Scholars Agreement*; takto získaný přístup k software a studijním materiálům plánujeme využít při výuce specializovaných předmětů.

8.8.4. Nepovedlo se

Nepodařilo se do konce dotáhnout portál na UPOL. Tamní pracovníci byli v roce 2004 zavaleni prací na implementaci SAP a na portál jim nezbýval čas. Situaci navíc zkomplikoval odchod spoluřešitele Stěpanova z CVT. Univerzita Palackého v Olomouci navíc nemá vyřešenou bázi uživatelů tak, aby šla propojit s portálem.

Dále se nám nepodařilo zcela uspokojivě vyřešit *content management* v portálu. Portlety *Statický text* a *FileExplorer* nejsou plnohodnotnou náhradou za WPCP a PDM. Zvládnutí této problematiky se budeme věnovat i po ukončení projektu PORT@L.

8.8.5. „Lessons learned“

Přestože jsme měli předem nastudována různá doporučení týkající se implementace portálu, i my jsme nakonec podcenili časovou a personální náročnost tohoto projektu. Zejména napojení na infrastrukturu zabralo opravdu mnoho času.

Podpora top managementu školy je velice důležitá; pokud portál nemá v jejich očích dostatečně vysokou prioritu, práce se zpožďují a zejména je obtížné „najmout“ další lidi.

Zpočátku jsme se domnívali, že portál zvládneme pouze vlastními silami; problematika je však natolik široká, že nezbývá než využít služby komerčních subjektů.

Zvládnutí vývoje portletů není jednoduché a trvá nějaký čas. Nový vývojář (zběhlý v programování Java aplikací) je schopný samostatné práce v přiměřené kvalitě teprve po cca dvou až třech měsících.

Celé problematice WebSphere Portal prakticky nikdo nerozumí úplně; každou informaci (včetně informací od personálu IBM) je potřeba ověřit.

8.9. Další využití výsledků projektu

Implementace portálu pro nás rozhodně nebyla jen akademickým cvičením. Na portálech budeme nadále pracovat a integrovat do nich další informace i aplikace. Počet uživatelů se po ukončení pilotního provozu zvýší z řádu stovek na tisíce a portál se opravdu stane (skoro) jediným místem, kde uživatel najde opravdu všechny podstatné informace, které potřebuje v každodenním styku s institucí.

Plný provoz portálu OSU již prakticky běží, na ZČU jej plánujeme od září 2005.

Postupně budeme rovněž upouštět od některých stávajících jednoúčelových aplikací, které mají ekvivalent v portálu, díky čemuž půjde i vyčíslit konkrétní úspory (provoz HW, SW, OS, aplikace, správa, energie a další).

Rádi bychom časem přešli na WebSphere Portal ve verzi 5.1, která je technologicky vyspělejší, má dobrý *content management* a širokému okruhu uživatelů přinese nové funkce. Všechny získané zkušenosti se nám samozřejmě budou hodit.

Získané znalosti se snažíme zpětně implementovat do výuky, až již formou specializovaných přednášek, seminářů s aktivními uživateli–studenty nebo jako témata diplomových prací.

Výstupy projektu

Tato kapitola informuje o dosažených výsledcích projektu, a to jak formou představení vybudovaných portálů tří univerzit, tak i jako seznam publikací a vystoupení na konferencích.

Obsah

9.1. Portál ZČU	120
9.1.1. Obecné informace	120
9.1.2. Struktura portálu	121
9.1.3. Zavedení decentralizovaných principů	122
9.1.4. Personalizace	122
9.1.5. Co se nepodařilo nebo jsme nestihli	122
9.2. Portál OSU	123
9.2.1. Struktura portálu	124
9.2.2. Personalizace	125
9.2.3. Přístupové role a profily	125
9.2.4. Co se nepodařilo nebo jsme nestihli	125
9.3. Portál UPOL (návrh)	125
9.3.1. Pravidla pro nasazení WSP v prostředí UPOL	126
9.3.2. Rozvrstvení portálu	127
9.3.3. Přihlašování uživatelů a přiřazení rolí	127
9.3.4. Integrace stávajících aplikací a SSO přístup	127
9.3.5. Další rozvoj portálu	128
9.4. Prezentace na konferencích	128
9.5. Publikované články	129
9.6. Vytvořené portlety	129

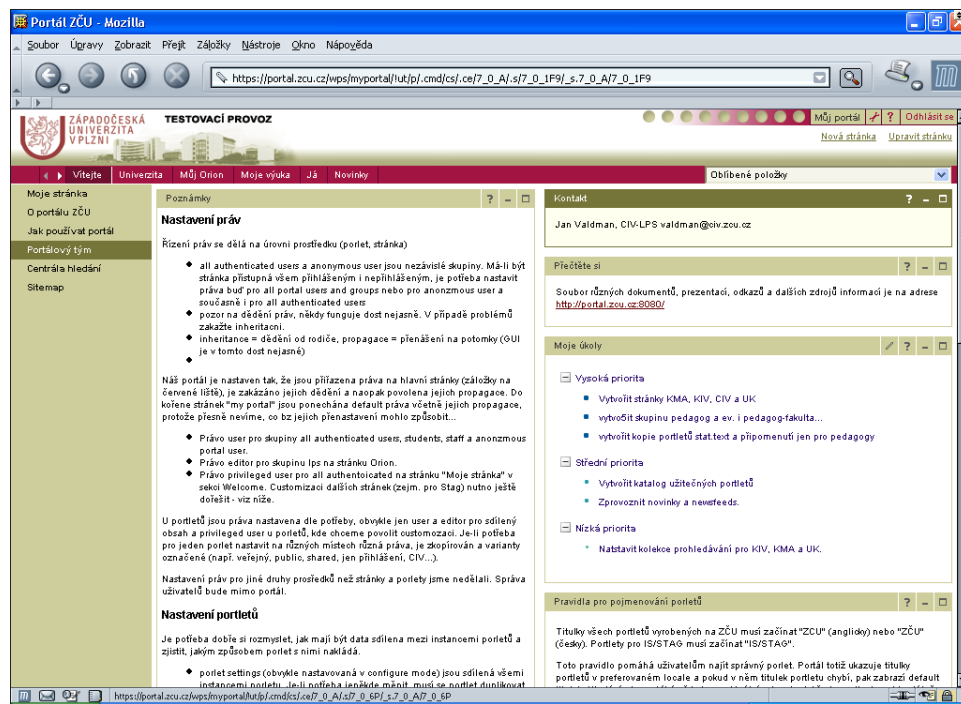
Hmatatelným výsledkem projektu je vytvoření tří naprosto samostatných portálů na třech školách. Každý portál je přitom odlišuje od svých „sourozenců“ a jde svojí vlastní cestou:

- Portál ZČU se soustředí na studenty, podporu výuky, transformaci studijní agenty IS/STAG a princip decentralizované správy obsahu.
- Portál OSU se vyprofiloval hlavně jako aplikační framework pro různé aplikace, zaměřené spíše na oblast zaměstnaneckého intranetu.
- Portál UPOL je zatím nejméně ambiciózní, vzhledem k nejrůznějšímu okolnostem (personální zdroje, stav IT infrastruktury, podpora vedení) prakticky jen kopíruje od ZČU a OSU. Jeho těžištěm je v integraci SAP do prostředí portálu.

Následuje stručné představení všech tří portálů, které informuje o jejich funkcionalitě množství integrovaných aplikací. Katalog vyrobených portletů je v závěru kapitoly.

9.1. Portál ZČU

Implementaci portálu do intranetového prostředí ZČU sledovala dva cíle: realizovat robustní aplikační framework pro třívrstvou sw architekturu a vytvořit konzistentní personalizovatelné informační prostředí s distribuovanou správou obsahu. Obou těchto cílů bylo z větší části dosaženo.



Obrázek 9.1: Ukázka portálové stránky

Portál ZČU je za zaměřen na vnitřní uživatele ZČU – studenty a zaměstnance. Obsahově se jedná o horizontální portál zaměřený na podporu studia a každodenní interakci uživatele s organizací. Portál neřeší oblast vědy a výzkumu (alespoň prozatím), technicko-administrativního provozu univerzity a ani není manažerským systémem (ale stávající MIS bude částečně integrován).

V této kapitole představíme portál z pohledu uživatele (což je asi nejzajímavější), ale je třeba si uvědomit, že se jedná jen o “špičku ledovce” – obrovské množství práce se skrývá ve změnách IT infrastruktury a “vnitropodnikových” procesů. Například obrovské komplikace nám způsobilo napojení portálu na adresář LDAP a autentizační infrastrukturu Kerberos, nebo několikaměsíční skluz projektu *Centrální registry ZČU*, jehož výstupy nutně potřebujeme.¹

9.1.1. Obecné informace

Portál plně podporuje dva jazyky: český a anglický. Další jazyky jsou podporovány pouze na úrovni portálového serveru, tj. lze je použít, ale uživatel vesměs uvidí anglický obsah. Portál nemá (alespoň ve střednědobé perspektivě) nahradit web školy nebo jejích částí (hlavně kateder), naše filozofie je “v portálu jsou jen informace, prezentace at’ jsou jinde”.

¹Centrální registr bohužel neexistuje ještě v době psaní tohoto textu – leden 2005.

9.1.2. Struktura portálu

Struktura portálu ZČU vychází ze zkušeností anglických univerzit (několik nezávislých hlavních tematických oblastí, ich-forma) a částečně též z výchozí struktury stránek WebSphere Portal.

Vítejte

Tato část portálu byla ponechána z původní instalační struktury IBM. Obsahuje přihlašovací obrazovku a všeobecné informace a pokyny vážící se k portálu. Nejvýznamnější je stránka *Moje stránka*, kterou jedinou si mohou uživatelé přizpůsobit tak, že si na ni zkopírují portlety umístěné na jiných stránkách portálu. Tím si z informační nabídky portálu uživatel vytvoří svůj osobní profil.

Univerzita

Tato stránka představuje rozcestník na stránky jednotlivých pracovišť, na která jsou delegována práva spravovat svoji část portálu. Každé pracoviště zde má (nebo výhledově bude mít) vytvořené své portálové stránky, jež si samo spravuje. K tomu účelu bylo vytvořeno několik různých portletů, které společně se standardně dodávanými portlety tvoří „stavební kameny“ pro tyto stránky. Zmínit můžeme například tel. seznam pracovišť, předměty vyučované katedrou, personální obsazení pracoviště, diskusní fórum, odevzdávání sem. prací atd.

V této části portálu vznikají v rámci pilotního projektu i portálové stránky jednotlivých pracovišť, na kterých ověřujeme jednotlivé modely vystavování a správy informací, ladíme funkce portletů a nastavení autorizace.

Můj Orion

Tato stránka představuje informační rozhraní uživatele k distribuovanému výpočetnímu prostředí ZČU s názvem Orion. Jsou zde informace o využití diskové a poštovní kvóty, důležité novinky a odkazy na další weby. V brzké době zde přibudou portlety pro informaci o blížící se expiraci hesla a jeho změnu, přístup k elektronické poště, who-is portlet a podobně.

Moje studium

Tato část portálu slouží pouze studentům. Stěžejní část tvoří portlety pro studijní agendu IS/STAG, jež byly vytvořeny přepsáním části původního STAGu z PL/SQL do třívrstvé architektury (portlety a EJB). Konkrétně se jedná o moduly student, předmět, rozvrh, známky... Začíná tak nová éra IS/STAG postavená na soudobé technologii. Vytvořené portlety jsou prozatím napsány v proprietární IBM API, nicméně jejich převod do JSR-168 API není složitý a již jej připravujeme. Na aplikační logice v EJB se samozřejmě nic měnit nemusí. Kolem této „stagové“ části se nabaluje informační část, do které mají přispívat prorektorka pro studium, děkani, proděkani a studijní oddělení fakult. K tomu účelu využíváme různé informační portlety, kterými se snažíme postupně nahrazovat statické texty, a zavést tak mechanismus vláken zpráv (content feeds), která chceme výhledově exportovat jako RSS.

Moje výuka

Tato část portálu je obdobou předchozí části (*Moje studium*), ale nyní zaměřená na pedagogii. Jsou zde portlety pro zadávání známek a vypisování termínu zkoušek,

fakultní diskusní fóra, a zpravodajská část pro pedagogy.

Já

Portál jako *digital dashboard* je mottem této části portálu, kde si uživatel může prohlížet údaje o něm vedené v různých univerzitních informačních systémech. V současné době je implementován pohled do PAM systému Magion zobrazující například zůstatek dovolené a přehled cestovních národ. (Obě tyto funkce jsou pro “normálního smrtelníka” jinak elektronicky nezjistitelné...)

Novinky

Od portálu si slibujeme i zavedení (víceméně) jednotné celouniverzitní zpravodajské platformy. Kromě toho, že zde agregujeme informace z různých zdrojů uvnitř i vně univerzity, jsme připravili i několik “novinkových” portletů. Některé portlety jsou jednoduché, pracující jen s jedním zdrojem dat (např. SQL tabulkou). Větší přínos očekáváme od zavedení systém vláken zpráv (newsfeeds), u kterých je individuálně vymezen okruh čtenářů i přispěvatelů. K dispozici je portlet zobrazující agregované zprávy z více vláken.

9.1.3. Zavedení decentralizovaných principů

Portál ZČU klade důraz na schéma *centralizovaná správa aplikací – decentralizovaná správa obsahu*. Portlety, které jsme vytvářeli pro tento portál, jsme od začátku koncipovali jako stavební kameny, které později použijí editoři jednotlivých stránek. K dispozici je celá řada portletů, univerzálních i specializovaných, která se bude v následujících letech dále rozrůstat.

Princip decentralizované správy obsahu jsme v rámci pilotního provozu zkoušeli na některých pracovištích (Katedra matematiky, Katedra informatiky, Univerzitní knihovna). Odezva od uživatelů (editorů stránek) je velmi pozitivní; ověřili jsme, že vytváření portálových stránek je velmi jednoduché (například matematici si pochvalují „programování bez programování“), a uživatelé k němu přistupují hravým způsobem.

9.1.4. Personalizace

Personalizace v portálu ZČU je řešena *deklarativním způsobem*, tj. nastavením přístupových práv k portletům nebo i celým stránkám.

Tento systém samozřejmě funguje, nicméně plně nevyužívá potenciál portálu. Technologie WPCP (WebSphere Content Publishing) umožňuje vytvářet portlety s dynamickým obsahem (content-spots) a pomocí WPCP engine do nich podle soustavy pravidel plnit dynamicky obsah.

Technologií WPCP jsme se zabývali až na přelomu let 2004 a 2005, kdy jsme zjistili, že authoring prostředí funguje v portálu jen pod OS Windows (máme linuxovou instalaci), a že tuto technologii IBM ve verzi portálu 5.1 opouští. Proto jsme se rozhodli zatím se problematikou *personalization* nezabývat a věnovat se raději vývoji dalších portletů a správě dokumentů.

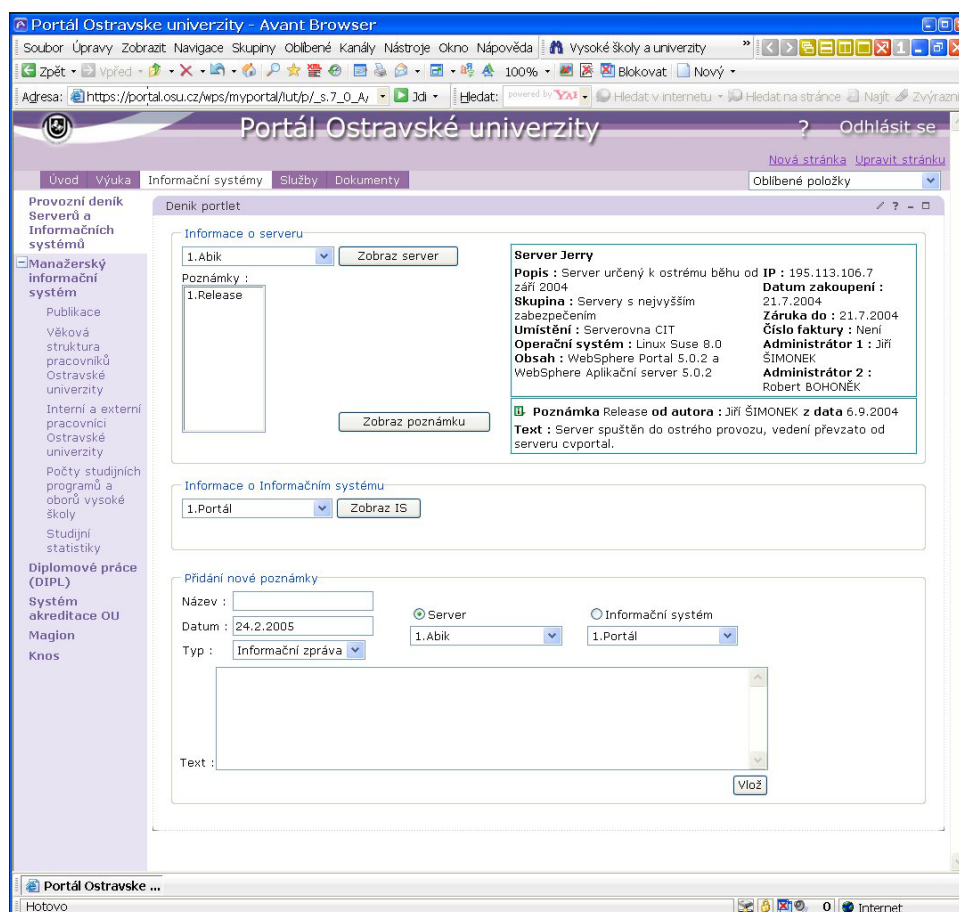
9.1.5. Co se nepodařilo nebo jsme nestihli

Obecně se dá říci, že náš portál zatím postrádá kolaborativní funkce. Po několika-měsíčních útrapách s napojením portálu na infrastrukturu jsme se rozhodli nein-

stalovat Lotus Quickplace a Sametime z důvodu nedostatku personálních zdrojů a malé „společenské poptávky“. Podobně i integrace Lotus Notes do portálu je zatím ve fázi příprav. Jak je poznamenáno na jiném místě, během 12 měsíců prostě nelze stihnout všechno. Zpoždění postihlo i integraci (rovná se přepisování) IS/STAG a dalších aplikací; vývoj zákaznických portletů je časově náročnější než jsme se původně domnívali. Dále jsem neřešili front-end SSO, protože informace od IBM byly rozporuplné a navíc jsme nezískali přístup k potřebné dokumentaci.

9.2. Portál OSU

Portál Ostravské univerzity se stal za dobu implementace nedílnou součástí běhu informačních systémů na Ostravské univerzitě. Hlavním cílem implementace portálu byla integrace informačních systémů na jedné platformě a plné využití třívrstvé architektury. Z hlediska koncového uživatele bylo hlavním cílem vytvoření personalizovaného informačního prostředí, které přebere funkci intranetu a nabídne webový přístup k naprosté většině informačních systémů.



Obrázek 9.2: Ukázka portálové stránky

Portál Ostravské univerzity se stává kompletním intranetem Ostravské univerzity, přebírá kompletní funkčnost Intranetu a umožňuje přístup k dalším těžko přístupným informacím. Jelikož je personalizován, tak je zaměřen přímo na osobu, která je přihlášená a dynamicky mění svůj obsah podle pravomocí příslušné osoby. Tudiž je často koncovými uživateli nazýván "osobní kartou zaměstnance, či studenta".

Portál obsahuje také Manažerský informační systém, který poskytuje statistické informace z jednotlivých oblastí. Dále je portál zaměřen na poskytování provozních služeb pro uživatele, přebírá stávající roztržitou funkčnost a zaplňuje její "bílé místa". V budoucnu chceme aby se stal dynamickým webem pro nepřihlášené uživatele.

Představme tedy portál tak jak ho vidí uživatel. To znamená čistě jeho prezentační vrstvu, pro uživatele jedinou možnost jak nahlížet na portál. Struktura se mění v závislosti na oprávnění přihlašované osoby, která se opírá o centrální registr osob Ostravské univerzity. Zde uvedeme celou možnou strukturu, kterou je na portálu možno využít.

9.2.1. Struktura portálu

V této kapitole je popsána pouze první úroveň struktury, nemá cenu zabíhat do druhých či třetích úrovní, kde se jedná už o specifické výstupy.

Úvod

Tato stránka uvítá uživatele na portálu. Úvodní informace nasměrují uživatele v případě jakéhokoli problému na help desk, či telefonický kontakt. Poskytne novinky, které se na univerzitě udály a podá úvodní informace o osobě.

Výuka

Sekce výuka pod sebou ukrývá informace pro učitele ve formě učitelských portletů k IS/STAG. Tyto portlety tvoří podstatnou část této sekce. Dále je zde možno nalézt nástroje k ověřování identity studenta, určeno především pro studijní oddělení a vyučující. Dále je zde implementováno sledování plateb za studium, kdy se v závislosti na identifikaci studenta pomocí centrálního registru osob sledují jeho platby za studium.

Poslední funkcí v této sekci je generování státnicových otázek v závislosti na vystudovaných předmětech studenta.

Studium

Tato část je určena pouze studentům a nachází se v ní portlety pro studijní agendu IS/STAG. Jež byly vytvořeny jako duplikace funkčnosti IS/STAG do třívrstvé architektury. Portlety budou tvořit takzvaný elektronický index a v návaznosti na něj se zruší klasický index v papírové podobě. Postupně se předpokládá kompletní migrace IS/STAG pod portál.

Informační systémy

Část Informační systémy je velice obsáhlou sekci a zahrnuje integrované informační systémy pod portál. Jedná se o dvojí integraci, jednak o převod stávajících informačních systémů, tak také o vývoj nových. Ukrývá se zde také Manažerský informační systém, poskytující statistické přehledy.

Služby

Tato sekce v sobě ukrývá funkce určené všem autentizovaným uživatelům portálu. Ať už se jedná o vstup do HelpDesku, či jiného systému, který se blíží svým smyslem

službě pro uživatele, nebo o službu faktickou, pak jsou informace zařazeny v této sekci. Dále jsou zde implementovány portlety pro podporu jiných informačních systémů.

Dokumenty

Tato stránka je vstupem do správy dokumentů, kterou jsme se snažili delší dobu implementovat pomocí standardních nástrojů WebSphere Portal, ale nepodařilo se. V současné budeme používat portlet firmy Cleverbee, který umožní ukládání dokumentů na Portál Ostravské univerzity.

9.2.2. Personalizace

Podobně jako plzeňští kolegové řešíme personalizaci „deklarativním charakterem“ a nevyužíváme dalších technologií implementovaných v portálu. Zjištění že vývoj v tomto odvětví se zastavuje a v další verzi portálu již nebude součástí nás v našem rozhodnutí jen utvrdil.

Nicméně personalizace řešena naším způsobem je dostatečně silná, pokud má za sebou dostatečně správně implementovaný ověřovací mechanismus s adresářovou strukturou. Z tohoto důvodu na Ostravské univerzitě dochází k restrukturalizaci LDAP.

9.2.3. Přístupové role a profily

V návaznosti řešení personalizace v portálu, byl spuštěn proces řešení kompletních přístupových práv ke všem informačním systémům na Ostravské univerzitě. Výsledný systém by měl sledovat osobu od jeho nástupu, až po ukončení vztahu s Ostravskou univerzitou. Celý systém přiřadí osobě dynamický profil, který se bude modifikovat podle potřeb.

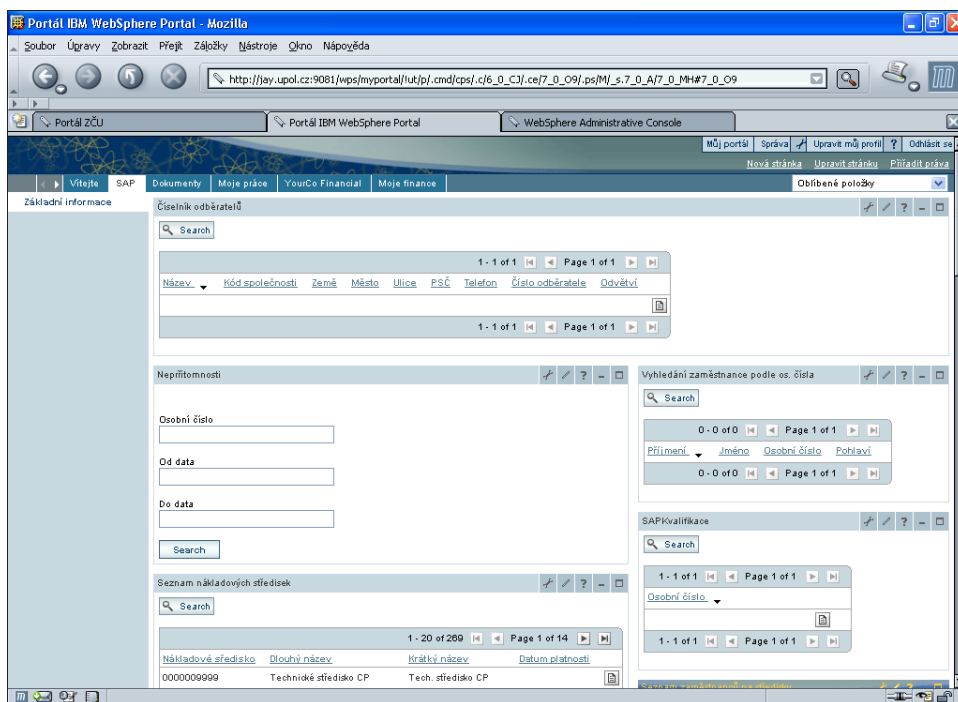
Systém je ve fázi analýzy, počítáme s jeho implementací v průběhu léta 2005.

9.2.4. Co se nepodařilo nebo jsme nestihli

Portál se ukázal jako vynikající integrační mechanismus, ale jak narůstají požadavky, tak je jasné, že jediné co jsme podcenili, tak je časový odhad implementace portálu. Koncoví uživatelé velmi vítají služby portálu a vznášejí své požadavky na tento systém, tudíž sami pomáhají tvořit správnou strukturu prezentační vrstvy. Tento způsob přináší spokojenost koncových uživatelů, ale na druhou stranu je velice náročný časově. Tudíž předpokládané ukončení první fáze implementace portálu na Ostravské univerzitě je konec roku 2005.

9.3. Portál UPOL (návrh)

Portálové řešení na UP v Olomouci je zaměřeno na integraci přístupu k mnoha používaným aplikacím do jednoho přístupového bodu. Na UP v Olomouci je nasazena celá řada aplikací pro různé cílové skupiny (zaměstnanci, studenti, vyučující, vědečtí pracovníci) a různé okruhy zpracování (průběh studia, ekonomika, vědecký růst...). Konečným cílem nasazení portálu se stalo zajištění přístupu ke všem používaným aplikacím prostřednictvím uživatelsky modifikovaných portletů nebo aspoň s využitím SSO – jednoho přihlášení pro všechny aplikace.



Obrázek 9.3: Ukázka portálové stránky

Vzhledem k tomu, že UP Olomouc využívá informační systém SAP pro řízení ekonomiky, personalistiky a mezd, zaměřili jsme se na integraci tohoto systému do prostředí portálu a obecně na integraci informačních systémů.

9.3.1. Pravidla pro nasazení WSP v prostředí UPOL

Statické HTML stránky univerzity slouží k prezentaci a vyvěšování veřejných informací a jsou umístěny na adrese www.upol.cz.

Portál UP Olomouc spojuje přístup ke všem aktivním (pracujícím s daty v databázi) aplikacím a je umístěn na adrese portal.upol.cz.

Portál obsahuje neautorizovanou, volně přístupnou část, ve které jsou soustředěny aplikace přístupné všem (např. vyhledávání osob) a chráněnou část přístupnou pod protokolem HTTPS, pouze po přihlášení uživatele.

Web www.upol.cz neobsahuje odkazy na jednotlivé univerzitou využívané aplikace vyžadující autorizovaný přístup. Přístup ke všem autorizovaným aplikacím se přesune do chráněné části portálu. Po přihlášení uživatele do portálu jsou mu nabídnuty přístupy k těm aplikacím, na které má právo.

Přímý přístup k původním samostatným webovým aplikacím bude zamezen a odkazy budou přeměrovány na portál.

Web www.upol.cz může obsahovat odkazy na aplikace, které nevyžadují autorizaci, přímo do místa jejich umístění v portálu.

Web www.upol.cz obsahuje odkaz na portál, přes který budou všichni zaměstnanci, učitelé a studenti přistupovat k aplikacím.

9.3.2. Rozvrstvení portálu

Portál se skládá z veřejné a chráněné části.

Ve veřejné části jsou umístěny informace o portálu, portlet pro přihlášení a samostatné stránky pro vstupy k veřejně přístupným aplikacím (vyhledávání osob, elektronická přihláška).

Chráněná část portálu představuje intranetové řešení s SSO přístupem k jednotlivým aplikacím a s rozvrstvením stránek a portletů podle rolí a práv, které má přihlášený uživatel.

Základní role jsou:

- Student
- Učitel
- Zaměstnanec
- Příslušnost k jednotlivým organizačním jednotkám (fakulty, katedry,...).

Práva jsou dána právy v jednotlivých aplikacích – například přímý přístup k informačnímu systému SAP se řídí vlastnostmi uživatele v tomto systému.

Každý uživatel portálu má přidělenou osobní stránku, do které si může umísťovat mu přístupné portlety podle toho, které nejčastěji využívá a jak je potřebuje mít k dispozici současně.

9.3.3. Přihlašování uživatelů a přiřazení rolí

Na UP v Olomouci neexistuje v současné době jednotný LDAP server jako adresářová služba. Informace o uživateli jsou roztrženy do několika systémů:

- Uživatelé se přihlašují do lokálních sítí Novell, jejich přihlašovací údaje jsou pro další systémy přístupny přes terminálové ověřovací servery Tacacs.
- Údaje o tom, kdo je student, zaměstnanec a učitel jsou dostupny v aplikaci Karty, která primárně slouží k evidenci a vydávání registračních karet studentů a zaměstnanců.
- Údaje o příslušnosti k jednotlivým organizačním jednotkám lze nalézt v aplikaci INIS.
- Studijní informace (co kdo studuje a co kdo učí) jsou umístěny v systému IS/STAG.
- Informace o zaměstnancích jsou evidovány v IS SAP.

Informace nutné pro přihlášení uživatele portálu se tedy získávají z těchto jednotlivých systémů a to jednak přímým přístupem (Tacacs), jednak replikací informací do pracovní databáze portálu.

Základním identifikačním údajem spojujícím jednotlivé informace v různých systémech je rodné číslo uživatele.

Po identifikaci uživatele při přihlášení jsou z informací z těchto systémů vytvořeny uživatelské role a identifikovány aplikace, na které má právo. Podle těchto rolí je pak personalizováno zobrazení portálu.

9.3.4. Integrace stávajících aplikací a SSO přístup

Integrace stávajících aplikací probíhá ve dvou krocích.

1. Prvním krokem je sjednocení přístupu k existujícím aplikacím přes jedno přihlášení do portálu. Portálu jsou známy postupy pro přihlášení původní aplikace, tato aplikace je vyvolána se svou původní prezentační vrstvou, ale uživatel je již přihlášen.

Příkladem tohoto přístupu je integrace stravovacího systému ANETTE – předávání autentizačních informací do stravovacího systému pomocí RC4 šifrovaného logínu uživatele portálu opatřeného časovým razítkem.

2. Druhým krokem je vytvoření portletů, které vytváří vlastní portálovou prezentační vrstvu nad informacemi původní aplikace.

Příkladem je soubor portletů pro práci se studijním systémem IS/STAG a pro zobrazování informací ze systému INIS.

Různé postupy použití SSO pro přístup k aplikacím:

- přístup k datovým zdrojům přes JNDI ODBC zdroje s uloženými autentizačními údaji.
- Přístup k aplikaci SAP s uložením přístupových informací do trezoru správy mandátů.
- Předávání dohodnutých autentizačních informací mezi portálem a aplikací při vyvolání této aplikace (systém ANETTE)

Hlavní integrované aplikace:

- IS/STAG – Informační systém pro vedení studijní agendy UP. Obsahuje kompletní evidenci studentů, sylaby předmětů, rozvrhy, termíny zkoušek atd.
- INIS – současný integrovaný informační systém UP. Umožňuje zaměstnancům UP získávat informace ze všech oblastí podle přidělených přístupových práv, doplňovat a aktualizovat své aktivity zejména v oblasti pedagogického a vědeckovýzkumného působení. Samosprávným orgánům poskytuje v Manažérském informačním systému v přehledné tabulkové podobě sumární údaje a pomocí poměrových ukazatelů nabízí analytické podklady k závěrům pro jejich rozhodnutí.
- SAP – ekonomický, personální a mzdový informační systém
- ANETTE/Kredit 7 – stravovací systém, objednávání jídel ve školních menzách.
- TinLib – knihovní systém.

9.3.5. Další rozvoj portálu

Integrace aplikací do portálového přístupového bodu je dlouhodobý proces, ve kterém budeme dále pokračovat. Naším cílem je postupná transformace systémů INIS a IS/STAG do prostředí portálu a postupná integrace portálu a statické HTML prezentace univerzity.

Pokusíme se dodržet pravidlo, aby všechna nová celouniverzitní řešení byla primárně realizovaná v prostředí portálu.

Podle vyhodnocení provozu portálu a odezvy uživatelů budeme upravovat sady dostupných portletů a strukturu portálu.

9.4. Prezentace na konferencích

Naše portálové snažení a jeho průběžné výsledky jsme prezentovali při různých příležitostech:

- setkání portálové skupiny CESNETu (duben 2004)

- konference Euroopen (květen 2004)
- konference UNINFOS 2004 (viz str. 214) pořádané EUNIS SK (prosinec 2004)

Dále jsme byli na 16.3.2005 společností IBM pozváni na její seminář "Superportál", abychom tam jako její zákazník prezentovali své zkušenosti s jejím produktem. Na semináři má být oficiálně uveden WebSphere Portal ve verzi 5.1 v České republice.

V květnu nebo červnu 2005 (čili po skončení tohoto grantu) budeme výsledky prezentovat na konferenci EUNIS CZ a dalším setkání portálové skupiny.

O Panevropské portálové konferenci (PEPC2004, viz str. 213) jsme se dozvěděli v okamžiku, kdy již byl program hotov, takže jsme nepřihlásili žádný příspěvek nehledě na to, že v létě 2004 toho ještě nebylo moc, co prezentovat. Svoji práci jsme tedy prezentovali pouze v diskusích, ve kterých jsme portálovou skupinu CESNETu potažmo Českou republiku dle našeho názoru dost zviditelnili.

V současné době se však domníváme, že portály ZČU a OSU jsou na zcela srovnatelné úrovni s anglickými školami. Minimálně ZČU plánuje vyslat své zástupce na další ročník. (Konference je putovní, v tuto chvíli není jasné, zda se bude konat v roce 2005 nebo až 2006).

Na konferencích IBM nejsou prezentace jejich zákazníků obvyklé, nicméně naši účastí jsme i tak budili jistý rozruch, protože účast lidí ze školství je zde minimální.

9.5. Publikované články

V rámci grantu bylo publikováno osm odborných článků/prací (viz str. 159), z toho dva byly recenzované, dva byly otištěny v univerzitních periodikách, jedním byla skripta a ve dvou případech se jednalo o diplomové práce vedené řešiteli grantu v přímé souvislosti s tímto projektem.

V souvislosti s postupným etablováním portálů na zmíněných školách budeme i po skončení grantu pokračovat v publikační činnosti převážně osvětového a vzdělávacího charakteru. Další články vzniknou na jaře 2005 po skončení grantu a budou shrnovat jeho výsledky a získané zkušenosti.

9.6. Vytvořené portlety

Není-li portál osídlen portlety s požadovanou funkcionalitou, zůstane jen sofistikovanou prázdnou infrastrukturou. Přestože jsem se snažili maximálně využít dostupné portlety, nezbývalo nám než několik desítek dalších portletů pracně vyrobit. Ukázalo se, že programování portletů je časově náročná záležitost (mají-li se udělat pořádně), a že problémem je i na školách sehnat dostatečně kvalifikované programátory, vyškolit je na portál, nechat je proniknout do problematiky a udržet je na projektu alespoň tak dlouho, aby dosáhli přiměřené produktivity.

Přehled vytvořených portletů je uveden v kapitole 10.

Katalog vytvořených portletů

V této kapitole prezentujeme portlety, které jsme vytvořili pro potřebu našich portálů.

Obsah

10.1 Univerzální portlety	132
10.1.1 Přihlašovací portlet	132
10.1.2 Sitemap	132
10.1.3 WhoAmI	133
10.1.4 Statický text	133
10.1.5 Diskusní portlet	134
10.2 IS/STAG portlety	134
10.2.1 Informace o studentovi	135
10.2.2 Plnění studijního programu studenta	135
10.2.3 Rozvrh studenta	136
10.2.4 Znamky studenta	136
10.2.5 Zkoušky studenta	136
10.2.6 Informace o učiteli	137
10.2.7 Rozvrh učitele	137
10.2.8 Diplomové práce učitele	137
10.2.9 Informace o předmětu	138
10.3 MAGION portlety	138
10.3.1 Personální informace	139
10.3.2 Cestovní náhrady	139
10.3.3 Kursovní lístek	139
10.3.4 Stravenky	140
10.3.5 Vývěska Magion	140
10.4 Aplikace ZČU	140
10.4.1 Telefonní seznam	141
10.4.2 Telefonní seznam pracoviště	141
10.5 Aplikace OSU	141
10.5.1 Statistiky OSU	141
10.5.2 Akreditace	142
10.5.3 Databáze závěrečných prací	142
10.5.4 Státnicové otázky	143
10.5.5 Informace o studentovi	143
10.5.6 Ověření platnosti ISIC karty	144

10.5.7 Jidelníček	144
10.5.8 KNOS	144
10.5.9 Peoplefinder	145
10.5.10 Testování síťových služeb	145
10.5.11 Hlášení poruch	146
10.5.12 Rezervace počítačových učeben	147
10.5.13 Webové statistiky	147
10.6 Aplikace UPOL	147
10.6.1 SAP Zaměstnanci	148
10.6.2 SAP Nepřítomnosti	148
10.6.3 SAP Kvalifikace	148

V následující části podáváme přehled vytvořených portletů, čímž jsou myšleny portlety od základu naprogramované v jazyce Java nebo vytvořené pomocí nástrojů na tvorbu portletů (portlet builders). Nejedná se o kompletní výčet, ale spíše o přehled řešené problematiky, resp. přehled oblastí a funkcí, které jsme pokryli portlety.

Snímky obrazovek byly pořízeny z různých instalací portálu (vývojové i provozní prostředí ZČU, OSU a UPOL) v lednu a únoru 2005, nicméně portlety neustále vylepšujeme, takže uvedené obrázky nemusí přesně odpovídat aktuálnímu stavu portletů. Snímky některých portletů (jmenovitě portlety pro IS/STAG) mohly být pořízeny z vývojového prostředí, a proto mohou zobrazovat „nereálná“ data.¹

Titulek portletu na obrázku (*Portlet Title* – určuje administrátor portálu) se běžně může lišit od skutečného jména portletu, které mu dal vývojář.

V některých případech byly portlety upraveny v grafickém editoru – jedná se však pouze o jejich přizpůsobení tisku, zejména vypuštění opakujících se řádek v tabulkách.

10.1. Univerzální portlety

Přestože WebSphere Portal je dodáván s mnoha portlety, většina z nich není pro univerzitní prostředí vhodná nebo použitelná. Překvapila nás i absence některých elementárních portletů jako je námi vytvořený Sitemap portlet nebo Statický text.

10.1.1. Přihlašovací portlet

WebSphere Portal standardně obsahuje přihlašovací formulář na samostatné stránce (tzv. screen), na kterou se uživatel dostane kliknutím na text “přihlásit se” v pravém horním rohu obyčejné stránky. Jelikož se nám to zdálo nepohodlné (text lze navíc na stránce snadno přehlédnout), předělali jsme přihlašovací stránku do podoby mnohem výraznějšího portletu.

Přístupová práva je potřeba nastavit tak, aby v roli uživatel byl pouze *anonymous portal user*.

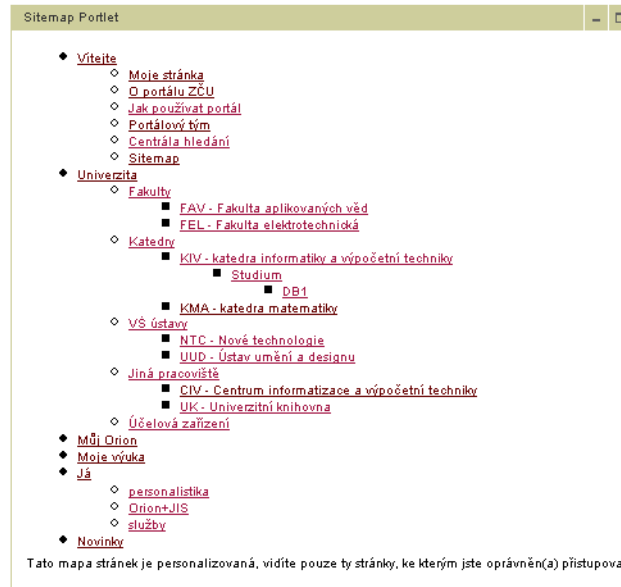
10.1.2. Sitemap

Podle standardů přístupnosti webů by každá stránka měla obsahovat odkaz na výchozí stránku webu (homepage) a také na seznam všech stránek (sitemap). Sitemap portlet v portálu kupodivu nebyl, ale podařilo se nám podle návodu v konferenci s využitím nedokumentovaného ModelAPI takový portlet vytvořit.

¹Například proto, že během testování nepoužíváme ostré databáze, ale jejich vývojářské kopie, ve kterých jsou pomíchána data.



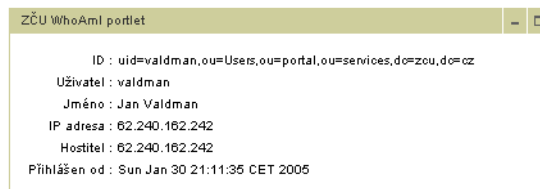
Obrázek 10.1: Přihlašovací portlet



Obrázek 10.2: Sitemap portlet

10.1.3. WhoAmI

Jedná se o velmi jednoduchý portlet, který zobrazuje základní údaje o přihlášeném uživateli. Výběr zobrazených informací je dán atributy použité Java třídy a není jistě ideální. Portlet využije hlavně administrátor, protože u běžného uživatele se nepředpokládá práce pod více identitami.



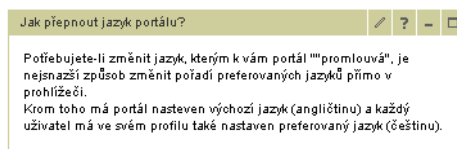
Obrázek 10.3: Who Am I portlet

10.1.4. Statický text

Asi největším nedostatkem WebSphere Portal je absence jednoduchého nástroje pro samoobslužné publikování statických informací. Dostupné portlety JSPserver (viz

strana 70) a FileServer (viz strana 66) nejsou vhodné pro řadové uživatele (nutnost vytvoření souboru na filesystému portálového serveru, zkopírování, přejmenování a konfigurace portletu), technologie WPCP je zase příliš složitá pro netechnické uživatele.

Tento nedostatek jsme řešili vytvořením portletu Statický text², jenž umožňuje zadat fragment HTML v editačním režimu. Portlet podporuje dva jazyky (český a anglický). Data jsou uchovávána přímo na portálové stránce v rámci objektu *PortletData*, čili tento portlet (resp. v něm uchovávaný text) nelze přesouvat mezi stránkami.



Obrázek 10.4: Statický text

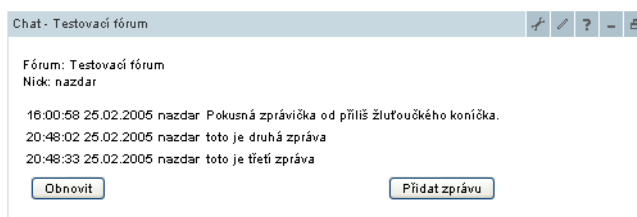
Portlet existuje i ve variantě, která data uchovávána v *PortletSettings*, tj. text zapsaný v portletu je na všech stránkách vždy stejný. Využitelnost této varianty je například pro různá upozornění nebo servisní zprávy.

Naším záměrem je tento portlet dále rozšiřovat: přidat podporu dalších jazyků a udělat pohodlnější jejich editaci (beze změny nastavení prohlížeče), změnit textarea na tzv. *HTMLArea*, umožnit jednoduché verzování obsahu a schvalování.

10.1.5. Diskusní portlet

Pro potřeby diskusí vyučujícího se studenty jsme hledali jednoduchý diskusní portlet. Nechtěli jsme zatím nasazovat IBM Sametime a žádný vhodný portlet nebyl k dispozici, a tak jsem se rozhodli vytvořit vlastní.

Chat portlet umožňuje vést jednoduchou diskusi (zatím bez podpory diskusních vláken), uživatel si může zvolit počet zobrazovaných zpráv a nick – ten může být povinný, volitelný nebo zakázaný.



Obrázek 10.5: Diskusní portlet

10.2. IS/STAG portlety

Následující portlety tvoří základ nové generace IS/STAG. Plně využívají výhody třívrstvé architektury, tj. veškerá aplikační logika je realizována pomocí EJB.

Všechny portlety podporují český a anglický jazyk, a z IS/STAG zobrazují data v příslušném jazyce (jsou-li k dispozici).

²podrobněji viz strana 67

Kapitola 10. Katalog vytvořených portletů

Portlety mezi sebou komunikují pomocí zaslání zpráv, například po kliknutí na zkratku předmětu v portletu Zkoušky se na tento předmět přepnou i další portlety.

V současné verzi umožňují portlety jen čtení dat, ale je již rozpracována další verze, která umožní některé aktivní operace, například přihlášení se na zkoušku. Vytvořené portlety nejsou složité, ale je za nimi skryto množství práce a úprav systému IS/STAG – například bylo potřeba přidat podporu více jazyků v některých tabulkách, protože v portálu chceme mít vše minimálně ve dvou jazycích.

10.2.1. Informace o studentovi

Portlet ukazuje jméno studenta, studijní číslo v IS/STAG, fakultu, obor a formu studia. Studuje-li student na více fakultách současně, má přiřazeno více studijních čísel, mezi kterými dovolí portlet přepínat. V takovém případě rozešle portlet zprávu i ostatním portletům, aby zobrazovali data ke správné identitě studenta.



IS/STAG - Údaje o studentovi

Marek KURKA

Aktivní	Osobní číslo	Fakulta	Obor	Typ	Forma
	A99223	FAV	Inženýrská informatika	Magisterský	Prezenční

Obrázek 10.6: Informace o studentovi

10.2.2. Plnění studijního programu studenta

Portlet funguje jako elektronická alternativa indexu nebo zápočtového listu, navíc ovšem studentovi dává možnost sledovat údaje již v průběhu semestru. Kliknutím na zkratku předmětu dojde k rozeslání zprávy ostatním portletům.

Zobrazeny jsou zkratky a názvy předmětů, jejich typ, počet kreditů, datum zakončení a některé další údaje.

Lze přepínat mezi zimním a letním semestrem.



IS/STAG - Student: přehled plnění předmětů

A99223 - Study Progress

2003/2004 Winter Semester

Type	Course Code	Course Title	Status	Credits	Try	Grade	Date	Room	New	Exams	Examiner	E-mail	Phone
1	KIV/ACS1	Architektury číslíkových systémů 1	A	6	1	2	30. 01. 2004	-	-	-	-	-	-
1	KAJ/AT3K	Angličtina pro techniky 3K	C	2	1	Splnil	17. 12. 2003	-	-	-	-	-	-
1	KIV/D5	Distribuované systémy	A	6	1	3	16. 02. 2004	-	-	-	-	-	-
1	KIV/FJP	Formální jazyky a překladače	A	6	1	3	21. 01. 2004	-	-	-	-	-	-
1	KTS/FO	Fotbal	C	2	1	Splnil	19. 12. 2003	-	-	-	-	-	-
1	KB/IMYEX	Mykologická exkurze	C	2	1	Splnil	08. 10. 2003	-	-	-	-	-	-
1	KIV/D5	Operační systémy	A	6	1	2	08. 01. 2004	-	-	-	-	-	-
1	KIV/PPI	Publikování a programování na Internetu	B	6	1	2	15. 01. 2004	-	-	-	-	-	-

Obrázek 10.7: Přehled plnění předmětů

10.2.3. Rozvrh studenta

Portlet umožňuje zobrazit rozvrh studenta buď ve formě řádkového výpisu nebo tabulky týdenního rozvrhu. Lze přepínat mezi zimním a letním semestrem.

Den	Od	Do	Místnost	Typ	Zkratka	Název	Vyučující
Úterý	07:30	11:00	UP115	Přednáška	KIV/VSP	Výkonnost a spolehlivost čísl. systémů	Doc. Ing. Stanislav Racek, CSc.
Úterý	11:10	12:50	UL208	Seminář	KAJ/AT5	Angličtina pro techniky 5	Mgr. Ivan Matta
Úterý	13:55	17:25	UU405	Přednáška	KIV/PPR	Paralelní programování	Doc. Ing. Stanislav Racek, CSc.
Středa	10:15	12:50	UL112	Přednáška	KIV/NMS	Navrhování mikro počítačových systémů	Dr. Ing. Karel Dudáček
Pátek	09:20	11:55	UP104	Přednáška	KIV/UPA	Úvod do počítačových architektur	Ing. Vlastimil Vavříčka, CSc.
Pátek	09:20	11:00	UL406	Cvičení	KIV/NMS	Navrhování mikro počítačových systémů	Dr. Ing. Karel Dudáček

Obrázek 10.8: Rozvrh studenta

10.2.4. Známky studenta

Portlet ukazuje studentovi seznam všech známek, které má v daném semestru zapsané v systému IS/STAG. Je zde zobrazena dosažená známka a počet kreditů, datum, jméno zkoušejícího a kontaktní údaje pro případ nejasností.

Zkratka	Název	Statut	Kredity	Pokus	Známka	Datum	Zkoušející	E-mail	Telefon
KIV/ACS1	Architektury čísl. systémů 1	A	6	1	2	30. 01. 2004	Ing. Vlastimil Vavříčka, CSc.	vavricka@kiv.zou.cz	2415
KAJ/AT3K	Angličtina pro techniky 3K	C	2	1	Splnil	17. 12. 2003	-	-	-
KIV/D5S	Distribované systémy	A	6	1	3	16. 02. 2004	Ing. Jiří Ledvína, CSc.	ledvina@kiv.zou.cz	2445
KIV/FJP	Formální jazyky a překladače	A	6	1	3	21. 01. 2004	Doc. Ing. Karel Ježek, CSc.	jezek_ka@kiv.zou.cz	2475
KTS/FD	Fotbal	C	2	1	Splnil	19. 12. 2003	-	-	-
KBI/MYEX	Mykologická exkurze	C	2	1	Splnil	08. 10. 2003	-	-	-
KIV/O5S	Operační systémy	A	6	1	2	08. 01. 2004	Prof. Ing. Jiří Šafařík, CSc.	safarikj@kiv.zou.cz	2439
KIV/PPI	Publikování a programování na Internetu	B	6	1	2	15. 01. 2004	Ing. Přemysl Brada, Ph.D.	brada@kiv.zou.cz	2435

Obrázek 10.9: Známky studenta

10.2.5. Zkoušky studenta

Portlet zobrazuje dva seznamy. V prvním jsou uvedeny termíny zkoušek, které si student již zapsal. Druhý seznam ukazuje další možné termíny, které si student může zapsat. Uvedeno je označení předmětu, datum, místo a čas zahájení zkoušky a kontakt na zkoušejícího.

Kapitola 10. Katalog vytvořených portletů

IS/STAG - Termíny zkoušek studenta						
Zapsané zkoušky						
Datum	Začátek	Předmět	Místnost	Zkoušející	E-mail	Telefon
03.02.2005	10:00	Paralelní programování	UU-407	Doc. Ing. Stanislav Raček, CSc.	straeok@kiv.zou.cz	2449
Nové zkoušky						
Datum	Začátek	Předmět	Místnost	Zkoušející	E-mail	Telefon
02.02.2005	10:00	Výkonnost a spol. systémů	UU-407	Doc. Ing. Stanislav Raček, CSc.	straeok@kiv.zou.cz	2449

Obrázek 10.10: Zkoušky studenta

10.2.6. Informace o učiteli

Portlet zobrazuje jméno, tituly a název kmenového pracoviště pedagoga tak, jak jsou tyto údaje zadané v IS/STAG. Pedagog má v IS/STAG vždy jen jedno osobní číslo, může ale pracovat na více pracovištích. V takovém případě portlet zobrazí přepínač, pomocí kterého lze zvolit pracoviště, ke kterému se vztahují údaje v ostatních portletech.

IS/STAG - Údaje o učiteli	
Ing. Maximilián Otta, Ph.D.	Oddělení: KIV - Katedra informatiky a výpočetní techniky

Obrázek 10.11: Informace o učiteli

10.2.7. Rozvrh učitele

Tento portlet zobrazuje rozvrh pedagoga. Svým vzhledem ani funkcí se příliš neliší od portletu Rozvrh studenta.

IS/STAG - Rozvrh učitele						
Můj rozvrh						
Zimní semestr <input type="button" value="v"/> Akademický rok: 2004/2005						
Den	Od	Do	Zkratka	Místnost	Název	Typ
Pondělí	07:30	09:10	KIV/DB1	UL411	Databázové systémy 1	Cvičení
Čtvrtek	07:30	09:10	KIV/DB1	UL409	Databázové systémy 1	Cvičení
Čtvrtek	09:20	11:00	KIV/DB1	UL409	Databázové systémy 1	Cvičení

Obrázek 10.12: Rozvrh učitele

10.2.8. Diplomové práce učitele

Portlet zobrazuje seznam diplomových prací vedených nebo oponovaných pedagogem. Je zobrazeno jméno diplomanta, název práce, pracoviště (katedra) a datum zadání.

Ve stávající verzi nelze vybrat předchozí akademické roky, zobrazeno je pouze aktuální období.

Příjmení	Jméno	Osobní číslo	Téma	Vztah k DP/BP	Pracoviště	Datum zadání
DRAŽKA	Miroslav	A00033	Evidence publikační činnosti v IBM Web Sphere portálu	Vedoucí	KIV	31. 08. 2004
HLAVATÝ	Jakub	A99071	Přístup k IS/STAG pomocí mobilních zařízení	Oponent	KIV	31. 08. 2004
JENÍČEK	Michal	A00078	Návrh a implementace základních EJB komponent pro IS/STAG	Vedoucí	KIV	31. 08. 2004

Obrázek 10.13: Diplomové práce učitele

10.2.9. Informace o předmětu

Portlet nabízí kompletní sadu informací o předmětu. Zobrazení se liší v závislosti na režimu zobrazení portletu: V normálním režimu portlet zobrazuje jen základní informace (název, počet kreditů, rozsah, způsob zakončení). Při maximalizovaném zobrazení portlet nabízí prakticky všechny informace uložené v IS/STAG (anotace, podmínující a vylučující předměty, literaturu, zařazení do studijních plánů, ...).

Standardně portlet zobrazuje data aktuálního akademického roku, ale lze vybrat i starší období.

Není-li portlet nakonfigurován, zobrazuje formulář, pomocí kterého lze vybrat libovolný předmět zadáním jeho zkratky a garantující katedry.

IS/STAG - Informace o předmětu

Zkratka katedry: Zkratka předmětu:

Rok varianty předmětu:

Název: Logika
 Počet kreditů: 6
 Rozsah (Př. + Cv. + Sem.): 3 + 2 + 0 [HOD/TYD · HOD/TYD · HOD/TYD]
 Semestr: zimní
 Zápočet před zkouškou: ANO
 Typ zakončení: Zkouška
 Započítává se do průměru: ANO
 Obsah předmětu: Rozhodnutelnost výrokové logiky, logické úsudky, rozhodnutelnost predikátové logiky, rezoluční a sekvenciální metoda ve výrokové a predikátové logice, reduktivní úsudky, nezářetelná logika a přibližné usuzování, přirozená dedukce. Přednášky budou předneseny ve spisovně staroslovenštině, zaznamenaný gotickým písmem a doprovázeny bohatou gestikulací vyučujícího.

Obrázek 10.14: Informace o předmětu (normální zobrazení)

10.3. MAGION portlety

Díky spolupráci s firmou MAGION system, a.s. má OSU k dispozici ERA model ekonomického systému MAGION a svolení k read-only používání všech dat.

Následující portlety jsou používány na OSU a ZČU (UPOL nepoužívá MAGION ale SAP).

10.3.1. Personální informace

Portlet zobrazuje osobní údaje přihlášeného zaměstnance: jméno, tituly, pracoviště, dvě různá osobní čísla. Navíc portlet zobrazuje i tarifní třídu, nárok na dovolenou a její zůstatek.

Drobným nedostatkem portletu je, že u mezd nepozná, k uzávěrce jakého měsíce jsou data zobrazována.

Proměnná:	Hodnota
Pracovník:	Jan Valdman
Osobní číslo (mzdy):	██████
Osobní číslo (účetnictví):	██████
Druh pracovního poměru:	0
Středisko:	Laboratoř počítačových systémů
Mzdová třída:	██
Mzdový stupeň:	███
Nárok na dovolenou:	██
Převod dovolené z min. roku:	1
Čerpání dovolené:	0
Zůstatek dovolené:	██

Obrázek 10.15: Personální informace

10.3.2. Cestovní náhrady

Tento portlet z několika míst v MAGIONu posbírání informace o cestovních náhradách (tuzemských a zahraničních) a chronologicky je zobrazuje. V současné verzi zobrazuje všechny zaznamenané cesty, tj. zatím neumí omezit výběr na konkrétní rok.

Číslo příkazu	Vystaveno	Popis	Částka	Vyúčtováno	Začátek cesty	Konec cesty
CES/9802/01	-	Valdman - Praha	██ Kč	20.12.2004	13.12.2004	13.12.2004
CES/9801/00	29.11.2004	Ing. Valdman, Slovensko	██ Kč	13.12.2004	29.11.2004	30.11.2004
CES/9802/00	-	Valdman - Praha	██ Kč	30.11.2004	24.11.2004	24.11.2004
CES/9802/00	-	Valdman - Praha	██ Kč	18.10.2004	22.09.2004	22.09.2004

Obrázek 10.16: Cestovní náhrady

10.3.3. Kursovní lístek

Portlet zobrazuje kursovní lístek podle hodnot uložených v kursovní tabulce MAGIONu, která uchovává denní hodnoty kursů. Lze jej využít k dodatečnému vyúčtování nebo kontrole služebních cest.

Zkratka měny	Počet jednotek na korunu	Cena
CAD	1	18.642
CHF	1	19.476
EUR	1	30.14
GBP	1	43.611
HUF	100	12.272
PLN	1	7.41
SKK	100	78.963
USD	1	23.118

Obrázek 10.17: Kurzovní lístek

10.3.4. Stravenky

Portlet slouží ke kontrole výdeje stravenek zaměstnancům. Chronologicky jsou zobrazeny údaje ovlivňující nárok na stravenky: odpracované dny, služební cesty a závodní stravování.

Datum	Akce	Cena stravenky	Počet kusů	Typ pohybu	Popis	Číslo dokladu
18.01.2005	-	50	2	Srážka za cestovné.	Pízeň	27579
23.11.2004	-	50	-2	Srážka za cestovné.	Nové město	25072
31.12.2004	-	50	18	Stravenky prodány pokladnou.	Prodej stravenek	98519
13.02.2004	-	35	-1	Srážka za cestovné.	Praha	14286
13.02.2004	-	35	1	Srážka za cestovné.	Praha	14286
31.01.2004	+	35	21	Nárok na stravenky z mezd (podle odpracovaných dní).	Nárok z mezd 2004/01	3
31.01.2004	-	35	12	Stravenky vyčerpány v menze.	Menza 2004/01	2

Obrázek 10.18: Stravenky

10.3.5. Vývěska Magion

Portlet představuje jednoduchý systém vyvěšování zpráv, které se zobrazují seskupené dle data zadání. S IS MAGION portlet souvisí jen tematicky, po malé úpravě (zobecnění) jej půjde používat univerzálně.

Datum	Úpravy
08.01.2005	
04.01.2005	
03.12.2004	<ul style="list-style-type: none"> • Úpravy v modulu banka • Úpravy v modulu účetnictví • Úpravy v modulu majetek • Úpravy v cestovní náklady • Úpravy v objednávkách materiálů a kooperací • Úpravy v modulu sklady

Obrázek 10.19: Vývěska MAGION

10.4. Aplikace ZČU

Následující portlety slouží pouze pro ZČU, i když je lze snadno upravit pro další subjekty.

10.4.1. Telefonní seznam

Portletová obdoba klasického telefonního seznamu. Portlet získává data přímým dotazem do adresáře LDAP a kromě telefonních čísel zobrazuje i další údaje.

Příjmení	Jméno	Informace	Telefon	E-mail	Kancelář
VALDMAN	Jan	STAFF CIV	37 763 2834	valdman@civ.zou.cz	UI403
VALDMAN	Michal	student FAV		valdi@students.zou.cz	
VALDMAN	Tomas	student FST		tvaldman@students.zou.cz	

Nalezeno záznamů: 3
[Zpět](#)

Obrázek 10.20: Telefonní seznam

10.4.2. Telefonní seznam pracoviště

Tento portlet získává údaje přímo z telefonní databáze ZČU, která je využívána k online nastavování telefonních ústředěn. Na rozdíl od portletu Telefonní seznam, jenž je určen k vyhledávání tel. čísel osob, slouží tento portlet k sumárnímu přehledu pracoviště.

Předpokládá se, že portlet bude nakonfigurován na konkrétní pracoviště a umístěn na jeho stránku. Portlet zobrazuje buď seznam osob nebo seznam funkcí a je uzpůsoben pro tisk.

Není-li portlet nakonfigurován, zobrazuje formulář, který dovolí vybrat jakékoli pracoviště.

Osoba	Linky	Přímé linky	E-mail	Místnost
BALÁK Oldřich, Ing.	2856	-	obal@civ.zou.cz	UI404
BARTONIČEK Jan	2806	-	bartonio@civ.zou.cz	UI207
BISCHOFDOVÁ Marie	2857	-	bischof@civ.zou.cz	UI207
BORŘÍK Jiří, Ing.	2832	-	borik@civ.zou.cz	UI402
BURIÁNEK Kamil, Ing.	2826	-	kburi@civ.zou.cz	UI322
ZÝKA Miroslav	2847	-	zyka@civ.zou.cz	UI418

Obrázek 10.21: Telefonní seznam pracoviště

10.5. Aplikace OSU

Následující portlety byly vyvinuty a jsou používány na Ostravské univerzitě. Do konce proběhla první vlna zaškolení uživatelů formou distančního vzdělávání, viz [OSU-1].

10.5.1. Statistiky OSU

Tento portlet s charakterem MIS systému obsahuje množství pohledů, které zobrazují statistické přehledy OSU. Podkladová data pocházejí z IS/STAG.

Přehled Ostravské univerzity

Počty studentů bakalářského, magisterského a doktorského studijního programu na jednotlivých fakultách a ústavech OU (stav k 01.02.2005)

Fakulta	Studijní programy				Forma studia		Studenti celkem
	Bc.	Mgr.	Navazující	Ph.D.	prezenční	kombinované	
Fakulta filozofická	759	858	192	44	1619	234	1853
Fakulta pedagogická	1145	1629	134	11	1525	1394	2919
Fakulta přírodovědecká	882	641	80	74	1497	180	1677
Fakulta zdravotně sociální	712	164	46	21	825	118	943
Institut pro Umělecká studia	121	102	0	0	223	0	223
Celkem	3619	3394	452	150	5689	1926	7615

Z přehledu v tabulce jsou vyznačeni studenti, kteří studium přerušili, nebo ukončili.
Do přehledu nejsou zahrnuti studenti studijních programů oceloživotního vzdělávání a studenti distančního studia.

[Zpět](#)

Obrázek 10.22: Statistiky OSU

10.5.2. Akreditace

Velmi rozsáhlá aplikace, která připravuje kompletní agendu akreditačního řízení pro MŠMT. Aplikace je napojena na IS/STAG, odkud se berou studijní programy a obory. V aplikaci je poté potřeba doplnit množství dalších vyžadovaných údajů.

Výstupem aplikace jsou PDF soubory, jež stačí jen vytisknout, podepsat a odeslat na MŠMT.

Akreditace

Institut pro Umělecká studia - seznam studijních oborů pro program: B8201 - Hudební umění

Číslo	Název	Délka studia	Forma studia	Typ	Stav akreditace
NEAKREDITOVÁNO					
8201R046	harfa	0	Prezenční	Bakalářský	neakreditováno
8201R055	lesní roh	0	Prezenční	Bakalářský	neakreditováno
8201R057	pozoun	0	Prezenční	Bakalářský	neakreditováno
POŽADAVEK NA AKREDITACI					
8201R044	fagot	3	Prezenční	Bakalářský	požadavek na akreditaci
8201R045	flétna	3	Prezenční	Bakalářský	požadavek na akreditaci
8201R047	hoboj	0	Prezenční	Bakalářský	požadavek na akreditaci
8201R048	housele	0	Prezenční	Bakalářský	požadavek na akreditaci
8201R051	klarinet	0	Prezenční	Bakalářský	požadavek na akreditaci
8201R052	klavír	0	Prezenční	Bakalářský	požadavek na akreditaci
8201R053	kontrabas	0	Prezenční	Bakalářský	požadavek na akreditaci
8201R060	violoncello	0	Prezenční	Bakalářský	požadavek na akreditaci
8201R062	viola	0	Prezenční	Bakalářský	požadavek na akreditaci
8201R064	zpěv	0	Prezenční	Bakalářský	požadavek na akreditaci

Dotazy a připomínky posílejte na: portal@service.osu.cz

Obrázek 10.23: Akreditace

10.5.3. Databáze závěrečných prací

Portlet zpřístupňuje databázi závěrečných prací, které byly obhájeny na OSU. Diplomové práce lze vyhledat a zobrazit ve formátu PDF.



Obrázek 10.24: Databáze závěrečných prací

10.5.4. Státnicové otázky

Portlet umožňuje dosáhnout stav, kdy student u státnic dostane jen otázky z těch předmětů, jež skutečně absolvoval. Otázky se vkládají přímo v portletu podle jednotlivých předmětů, skutečně absolvované předměty se zjišťují z databáze IS/STAG.

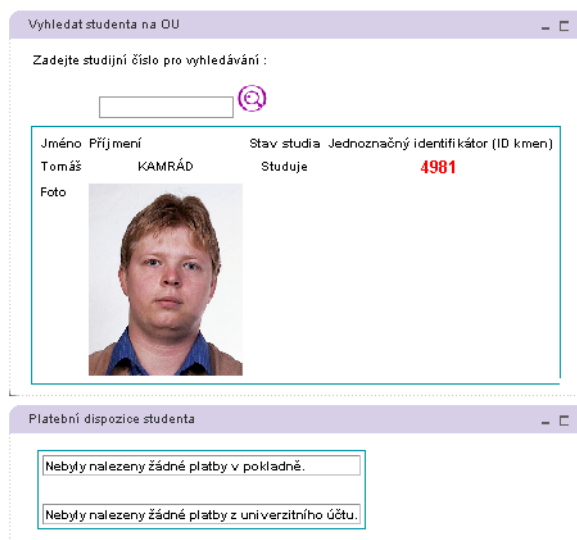


Obrázek 10.25: Státnicové otázky

10.5.5. Informace o studentovi

Portlet agreguje informace z několika databází. Studijní informace jsou v IS/STAG, fotografie z databáze ISIC karet. Portlet vyhledává studenta podle osobního čísla a zobrazí o něm základní informace.

Portlet zasílá zprávu do portletu "Platební dispozice studenta", který z databáze IS MAGION zobrazí všechny platby spárované se studentem.



Obrázek 10.26: Informace o studentovi

10.5.6. Ověření platnosti ISIC karty

Portlet umožňuje ověřit platnost ISIC karty na základě jejího čísla, které je do HTML formuláře v reálném provozu vložena čtečkou karet integrovanou do klávesnice.

Portlet zobrazí základní informace o držiteli karty.



Obrázek 10.27:

10.5.7. Jídelníček

Portlet zobrazuje jídelníček v univerzitních menzách včetně cen i počtu dostupných porcí a umožňuje rovněž jídla objednávat.

10.5.8. KNOS

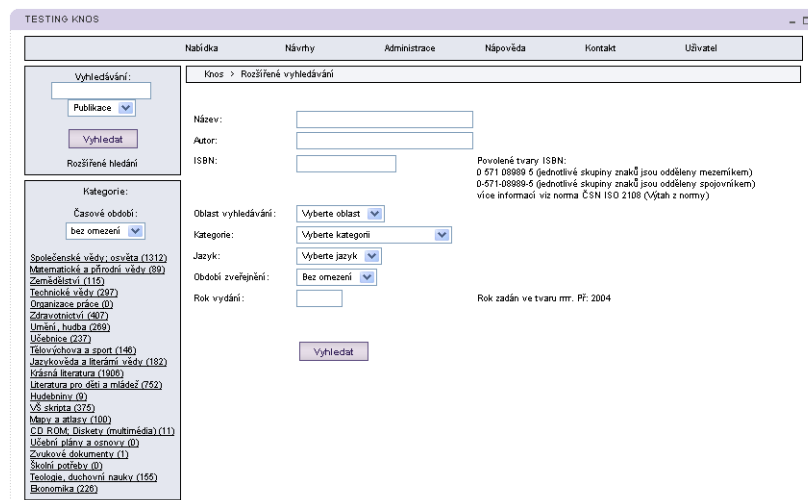
Kompletní aplikace “Knihovní objednávkový systém” byla v průběhu jejího vytváření upravena pro portálové prostředí.

Kapitola 10. Katalog vytvořených portletů



Obrázek 10.28: Jídelníček

Aplikace je napojena na zdroje knižních novinek, ze kterých si mohou zaměstnanci vybírat. Vybrané knihy jsou po schválení vedoucím pracoviště předány univerzitní knihovně k objednání. Aplikace obsahuje vlastní autorizační systém.



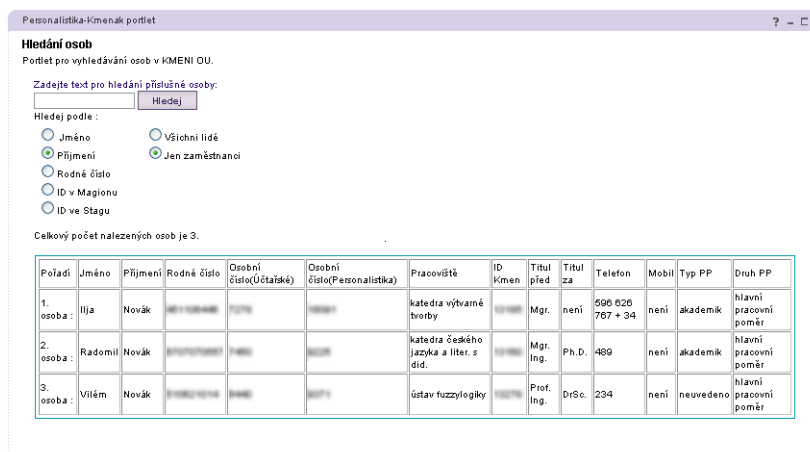
Obrázek 10.29: Knihovni objednávkový systém

10.5.9. Peoplefinder

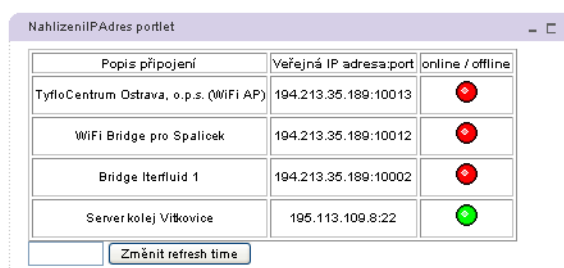
Portlet slouží k vyhledávání v centrálním registru osob na OSU. Hledat lze podle různých kritérií: jména, příjmení, rodného čísla a identifikátorů (osobních čísel) různých systémech.

10.5.10. Testování síťových služeb

Portlet slouží k elementárnímu testování dostupnosti zvolených síťových služeb formou pokusu o navázání spojení. Cílem není konkurovat nástrojům typu Nagios, ale poskytnout jednoduchou utilitu pro netechnické uživatele.



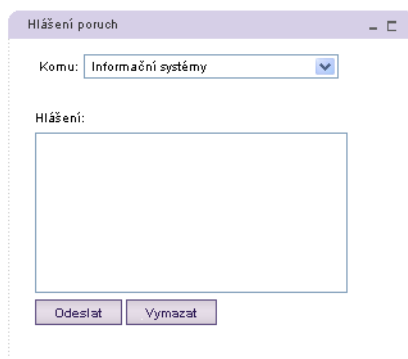
Obrázek 10.30: Hledání osob



Obrázek 10.31: Testování síťových služeb

10.5.11. Hlášení poruch

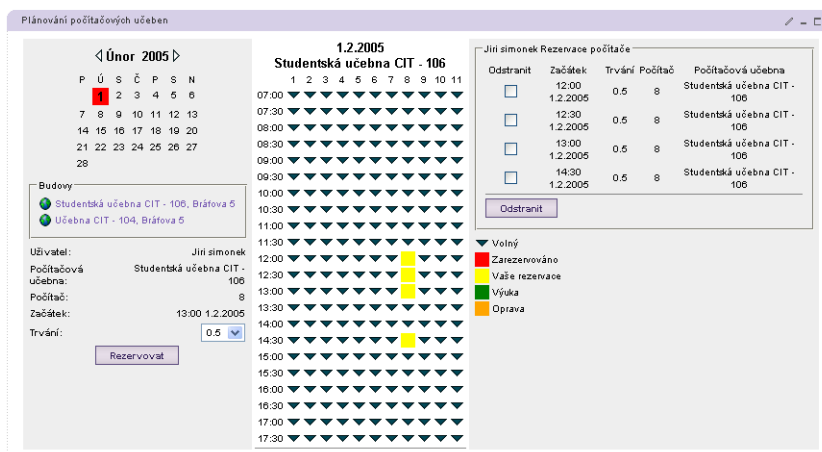
Portlet slouží jako webové rozhraní pro zadávání požadavků na HelpDesk. Každému adresátovi z rozpalovacího seznamu odpovídá e-mailová adresa, na kterou je odeslán text zadaný do formuláře.



Obrázek 10.32: Hlášení poruch

10.5.12. Rezervace počítačových učeben

Aplikace pro rezervaci veřejných počítačových učeben, jejíž zprovoznění se plánuje na jaro 2005.



Obrázek 10.33: Rezervace počítačových učeben

10.5.13. Webové statistiky

Tento portlet je zástupcem asi tuctu dalších portletů, které technologií web clipping „rozřezávají“ jistou www stránku s globálními statistikami oficiálního webu OSU, aby tyto fragmenty zprostředkovaly vybraným uživatelům.

Dokumenty počty přístupů	
DOKUMENTY	
Vnitřní předpisy, Statut DU	1137
Řídící normy	1186
Výroční zprávy	378
Organizační řád	908
Dlouhodobý záměr	402

Obrázek 10.34: Webové statistiky

10.6. Aplikace UPOL

Následující sada portletů byla vytvořena na Univerzitě Palackého v Olomouci pomocí integračních nástrojů pro SAP:

- **Využití portletu WPAI JDBC portlet builder**
Tento portlet byl využit pro zobrazení veřejně přístupných informací o zaměstnancích ze systému INIS.
- **Využití portletu WPAI pro SAP**

1. *SAPCostCenterList* – Seznam nákladových středisek – přehled nákladových středisek a jejich dlouhý a krátký název

2. *SapCustomerDetail* – Seznam odběratelů – obsahuje adresy odběratelů a jejich bankovních spojení
3. *SapListEmployeesByCostCenter* – Seznam zaměstnanců na středisku
4. *SAPAbsences* – Nepřítomnosti – přehled nepřítomností pracovníka v zadaném období
5. *SAPPersdata* – Vyhledání zaměstnance podle osobního čísla – zobrazení osobních údajů zaměstnance
6. *SAPBankDetails* – Bankovní spojení – po zadání osobního čísla zaměstnance zobrazí jeho bankovní spojení
7. *Kvalifikace* – po zadání osobního čísla zobrazí obory, ve kterých daný pracovník působí a jejich procentuální ohodnocení. Byla zde použita BAPI funkce `BAPI_QUALIFIC_GETLIST`.

- **Využití portletu IBM SAP Enterprise Application**

SAP ESS Remuneration – Výplatní páska – zobrazení výplatní pásky po zadání osobního čísla zaměstnance.

10.6.1. SAP Zaměstnanci

Portlet vyhledává a zobrazuje seznam zaměstnanců na pracovišti. Portlet je ze sady WPAI pro SAP a byl upraven nástrojem SAP Portlet Builder.

Středisko	Osobní číslo	Jméno
9630	0000000000	KRŠKOVÁ Milena RNDr.
9630	0000000000	ROGL Jan RNDr.
9630	0000000000	WEBEROVÁ Ivana Ing.
9630	0000000000	HANÁK Jiří Ing.

Obrázek 10.35: SAP zaměstnanci na středisku

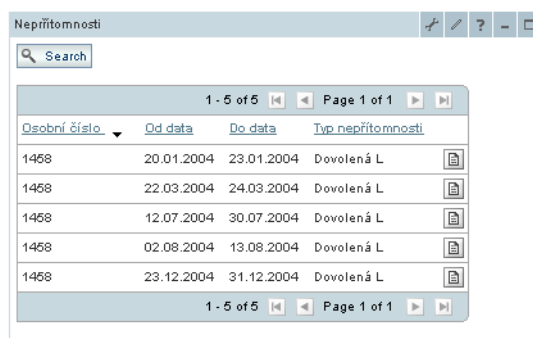
10.6.2. SAP Nepřítomnosti

Portlet vyhledává nepřítomnosti osob na daném pracovišti a zobrazuje jejich důvody. Portlet je ze sady WPAI pro SAP a byl upraven nástrojem SAP Portlet Builder.

10.6.3. SAP Kvalifikace

Portlet zobrazuje oborovou alokaci kvalifikace akademických pracovníků a její procentní ohodnocení. Portlet byl vytvořen nástroji SAP Business Object Editor a SAP Portlet Builder.

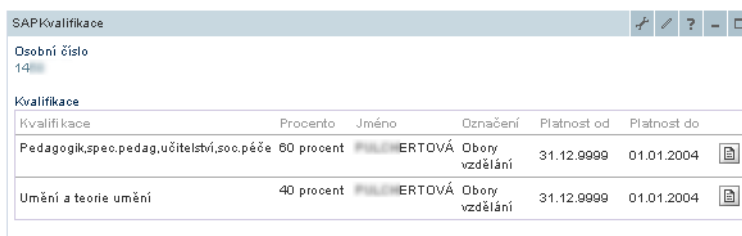
Kapitola 10. Katalog vytvořených portletů



The screenshot shows the SAP Absence Management (Nepřítomnosti) interface. It features a search bar at the top left and a table of absence records. The table has columns for 'Osobní číslo' (Employee ID), 'Od data' (Start Date), 'Do data' (End Date), and 'Typ nepřítomnosti' (Type of Absence). There are five records, all for employee 1458, with various dates and the type 'Dovolená L' (Paid Leave). The interface includes navigation controls for pages and records.

Osobní číslo	Od data	Do data	Typ nepřítomnosti
1458	20.01.2004	23.01.2004	Dovolená L
1458	22.03.2004	24.03.2004	Dovolená L
1458	12.07.2004	30.07.2004	Dovolená L
1458	02.08.2004	13.08.2004	Dovolená L
1458	23.12.2004	31.12.2004	Dovolená L

Obrázek 10.36: SAP nepřítomnosti



The screenshot shows the SAP Qualification (SAPKvalifikace) interface. It displays the employee ID '1458' and a table of qualification records. The table has columns for 'Kvalifikace' (Qualification), 'Procento' (Percentage), 'Jméno' (Name), 'Označení' (Designation), 'Platnost od' (Valid From), and 'Platnost do' (Valid To). There are two records, both for employee 1458, with percentages of 60% and 40%, and the name 'PAULINA ERTOVÁ'. The designations are 'Obory vzdělání' (Fields of Study) and 'Umění a teorie umění' (Art and Theory of Art). The validity dates are 31.12.9999 to 01.01.2004.

Kvalifikace	Procento	Jméno	Označení	Platnost od	Platnost do
Pedagogik,spec.pedag,učitelství,soc.péče	60 procent	PAULINA ERTOVÁ	Obory vzdělání	31.12.9999	01.01.2004
Umění a teorie umění	40 procent	PAULINA ERTOVÁ	Obory vzdělání	31.12.9999	01.01.2004

Obrázek 10.37: SAP kvalifikace

Závěr

Obsah

11.1 Shrnutí	151
11.2 Portál – ano či ne?	152
11.3 Vyplatí se portál?	152
11.4 WebSphere Portál – ano nebo ne?	152
11.5 WebSphere Portál – stojí za ty peníze?	153
11.6 Naše doporučení	153

11.1. Shrnutí

Během dvanácti měsíců se nám podařilo na třech (dvou?) školách vybudovat prakticky od nuly funkční portálové prostředí, napojené na místní infrastrukturu osazené reálnými aplikacemi. To vše v počtu čtyř řešitelů, čtyř spolupracujících osob a s pomocí několika dalších studentů–diplomantů.

Museli jsme vyřešit mnoho technických problémů a vypořádat se s nedostatečnými zkušenostmi, chybějící dokumentací, chybami v produktu i malou podporou ze strany IBM a jejich partnerů.

Vytvořené portály jsou „funkční“ (dá-li se to tak říci), nicméně budou potřebovat další rok na své etablování a „přijetí“ uživateli.

Chybou se ukázalo koncipovat grant jen na 12 měsíců. V této době je podle našeho názoru nemožné vybudovat od nuly plnohodnotný portál. S našimi dnešními zkušenostmi by to už asi možné bylo, ale v současné době nám chybí cca 6 měsíců na dotažení některých oblastí, větší etablování portálu v organizacích a také čas na prezentaci výsledků.

Žádná ze škol nedokázala vytvořit potřebný tým alespoň pěti osob, které by se na plný úvazek věnovaly portálu. Místo toho tvořili na každé škole jádro týmu 1–2 pracovníci, okolo kterých se *pohybovalo* několik dalších specialistů. Logicky pak chybí čas na hlubší porozumění produktu i jednotlivým technologiím a vzájemná opora. Tento problém se nám částečně povedlo obejít specializováním jednotlivých škol, nicméně našim případným následovníkům doporučujeme vybudovat tým o velikosti alespoň 10–15 osob. Jinak hrozí situace mnohokrát zopakovaná na anglických univerzitách: portál dělají 2–3 lidi, kteří jej za 2–3 roky dostanou do pilotního provozu.

11.2. Portál – ano či ne?

Řešitelé projektu se shodují, že portál typu WebSphere Portal, tj. portál 3. generace znamená pro jejich organizaci nezpochybnitelný přínos. Portál umožňuje do jednoho centrálně spravovaného místa integrovat množství různých aplikací/informací a řídit k nim přehledně přístup.

Produkt WebSphere Portal ve verzi 5.0 až 5.0.2 splnil naše požadavky a umožnil nám vybudovat portály dle našich původních záměrů. Jediným větším negativem jsou nedostatky ve správě dokumentů a správě obsahu, které jsou však již v novější verzi 5.1 odstraněny.

11.3. Vyplatí se portál?

I když nebudeme uvažovat v termínech TCO a ROI, lze snadno nahlédnout, že pořizovací náklady na portál jsou cca 1–2 mil. Kč na jednu školu, které by se měly nějakým způsobem vrátit.

V současném okamžiku, tj. ve fázi pilotního provozu portálu, je příliš brzo na nějaké ekonomické vyhodnocení jeho přínosu; jsme ale přesvědčeni, že má-li škola na začátku projektu jasnou vizi, je pro ni portál přínosem, minimálně v integraci webových aplikací *na jedno místo* a v decentralizované správě takového prostředí. Díky portálu lze prostě a snadno dělat věci, které by se jinak dělaly jen obtížně.

V této souvislosti bychom rádi upozornili, že nabídky komerčních firem na různé *quickstarty* nebo *uděláme vám portál za 30 dnů* za několik desítek tisíc korun, nelze brát vážně. Vesměs se jedná o pouhou instalaci portálového produktu.

Na druhou stranu by ale šlo s týmem 20–30 zkušených lidí jistě implementovat portály podobné těm našim za dobu podstatně kratší, řekněme 1–2 měsíce.

Nemalou část lidských zdrojů spotřebuje vývoj vlastních portletů. Vývoj portletu zabere zhruba jeden člověkotýden a za jeho zakázkový vývoj si komerční firmy účtují řádově stovky tisíc Kč. Proto je velmi důležité, že lze jednotlivé portlety mezi školami měnit, což se nám v praxi velmi osvědčilo.

Na začátku jsme neměli prakticky žádné univerzálně použitelné portlety pro univerzitní prostředí, nyní jich máme několik a k tomu ještě desítky dalších specializovaných portletů.

11.4. WebSphere Portál – ano nebo ne?

S produktem WebSphere Portal jsme víceméně spokojeni, systém je stabilní a splňuje funkce, které od něho požadujeme. Jako naprosto pravdivá se ukázala teze společnosti Gartner, že implementace IBM WebSphere Portal vyžaduje *nadměrné počáteční úsilí* ze strany zákazníka.¹

Zklamání jsme byli z absence některých základních portletů/funkcí jako jsou například novinky nebo mapa stránek, které je nutné naprogramovat.

Dalším nedostatkem je content management (složitá technologie WPCP, authoring jen pro Windows, atd.), který je však ve verzi 5.1 již nahrazen mnohem lepší technologií.

Ve verzi 5.0 jsou dodávány komponenty Lotus Collaborative Components, z pohledu nových implementací je ale výrazně výhodnější jít od verze 5.1 cestou Lotus Workplace, který nabízí stejnou funkčnost, ale na Java technologii a výrazně lepší integraci do portálu.

¹vztahováno sice na verze 4.x, ale platí i pro 5.x

WebSphere Portal Express 5.0 je dobrý produkt, architekturou a funkcemi zapadá bez problémů do univerzitního prostředí, přestože se jedná o podnikový produkt. Je potřeba se smířit s potřebou doprogramování mnoha konkrétních portletů pro univerzitní prostředí, ale to asi odpovídá filozofii WebSphere Portal (framework, nikoli balíček obchodních aplikací).

Jako největší problém vidíme špatně fungující support (Passport Advantage) a malé know-how v České republice, jak u IBM tak i u jejích partnerů. Situace se však postupně zlepšuje. K některým zásadním informacím se zákazník dostane pouze na základě osobních kontaktů nebo účasti na konferencích IBM.

11.5. WebSphere Portál – stojí za ty peníze?

Finanční efektivnost implementace WebSphere Portal je asi diskutabilní. Samotná licence za vlastní portál ve verzi Express není zanedbatelná (cca 21 000 Euro – cena pro školy), přičemž tato verze neumožňuje škálování portálu pomocí clusterů (takový požadavek se jistě časem objeví s postupujícím etablováním a zatěžováním portálu). K tomu je potřeba připočítat ještě mzdové náklady na zhruba dva člověko-roky práce a cenu za nákup know-how a dalších služeb.

WebSphere Portal určitě není levný, zvláště přihlédneme-li k množství použitého opensource kódu. Na druhou stranu ale zákazník získává licenci nejen na portál, ale i na aplikační server, LDAP server, databázi, webserver, vývojové nástroje, *instant messaging* a týmové nástroje.

Filozofie IBM je: „Dáváme vám portál a spolu s ním vše, co budete potřebovat.“ Ironií je, že část těchto komponent jsme vůbec nepotřebovali.

Zásadní otázku však vidíme jinde: WebSphere Portal ve verzi Express je licencován na „single box“ do 4 CPU, čili nelze provádět clusterování, a to ani spojením několika licencí Express. Odhlédneme-li od výkonových parametrů (netestovali jsme a zatím nepocitujeme problém), zůstává zde problém dosažení „high availability“ konfigurace. Otázkou tedy je, zda by úvahy neměly být vztahovány k verzi Enable, která pro školy stojí cca 36 000 Euro – čili skoro dvojnásobek ceny Express+.

11.6. Naše doporučení

Kdo potřebuje jen lehký portál (tedy portál 2. generace), necht' sáhne po opensource řešení (např. uPortal) a raději investuje úsilí do jeho počestění a dalšího vývoje. Důležité je striktně se držet standardu JSR-168, který zaručí, že vytvořené portlety bude v budoucnu možné využít i jiných portálech.

Kdo potřebuje solidní portálový aplikační framework, necht' sáhne po WebSphere Portal, pokud se mu to vyplatí (viz ceny verze Express nebo Enable). Využití komponent tvořících přidanou hodnotu (hledání, transcoding, atd.) je problematické, ale ve verzi 5.1 zákazník dostane solidní web content management a další užitečné nástroje Lotus Workplace pro online spolupráci.

IBM WebSphere Portal vám dá do rukou prázdný framework, který je uživateli k užítku pouze tehdy, je-li naplněn nějakým obsahem. Obsah může být buď statický (HTML, dokumenty), který již pravděpodobně máte k dispozici a zbývá nalézt cestu, jak jej zveřejnit v portálu.

Dynamickým obsahem jsou aplikace, které zřejmě budete muset sami vyrobit. Schopnost vlastního vývoje portletů v jazyce Java tedy považujeme za klíčovou. V tomto ohledu můžeme z vlastní zkušenosti doporučit strukturovat vývojový tým tak, aby v něm byli alespoň dva specialisté na portálové a další nezbytné technologie, kteří by kromě celkové architektury aplikace připravili i základní rámce komponent pro aplikační programátory, poskytovali jim podporu při řešení problémů a

dohlíželi na dodržování „dobrých zásad“ při implementaci. To proto, že *J2EE* prostředí je velice komplexní a zvládnutí programovacího jazyka *Java* (které již dnes konečně pomalu začíná být součástí základního vzdělání absolventa informačního oboru na vysoké škole) je nutnou, ale zdaleka ne postačující podmínkou.

Případným našim následovníkům doporučujeme plánovat podobné projekty minimálně na 18 měsíců a vyčlenit v nich finanční zdroje také na zaplacení alespoň tří odborníků, kteří se budou portálu plně věnovat.

Máte-li možnost podepsat IBM Scholars Agreement, určitě tak udělejte, protože je to nejnázší cesta, jak se dostat ke studijním materiálům a dalšímu software.

Literatura

- [1] Kolektiv: *IBM WebSphere Portal for Multiplatforms V5 Handbook*. IBM Redbook sg246098. IBM Press 2004.
- [2] Kolektiv: *IBM WebSphere Application Server V5.0 Management and Configuration*. IBM Redbook sg246195. ISBN 0-7384-2661. IBM Press 2004.
- [3] Kolektiv: *WebSphere Application Server-Express*. IBM Redbook sg246301. ISBN 0-7384-5315-3. IBM Press 2004.
- [4] Kolektiv: *A Secure Portal*. IBM Redbook sg246077. IBM Press 2003.
- [5] Kolektiv: *IBM WebSphere V5.0 Security*. IBM Redbook sg246573. ISBN 0-7384-2549-4. IBM Press 2004.
- [6] Kolektiv: *IBM WebSphere Portal V5 A guide for Portlet Application Development*. IBM Redbook sg246076. ISBN 0-7384-9851-3. IBM Press 2004.
- [7] Kolektiv: *IBM WebSphere V5.0 for Linux, Implementation and Deployment Guide*. IBM RedPaper 3601. IBM Corporation 2003.
- [8] Blinstubas Peter: *Portlet Development with portal Toolkit V5*. IBM Corporation, 2003.
- [9] Kolektiv: *Guide to WebSphere Portal 5.0* IBM Corporation, 2003.
- [10] Iyengar Ashok, Gadepalli Venkarta: *IBM WebSphere Portal Primer*. ISBN 1-931182-13-2. IBM Press 2003.
- [11] Kolektiv: *Mastering IBM WebSphere Portal*. ISBN 0-7645-3991-4. Willey Publishing, Indianapolis 2004.
- [12] Anonymous: *IBM WebSphere Portal Version 5 Administration. Student Notebook*. Studijní curriculum IBM, course code SW532. IBM Corporation, 2004.
- [13] Anonymous: *IBM WebSphere Portal Version 5 Administration. Student Exercises*. Studijní curriculum IBM, course code WF381. IBM Corporation, 2004.
- [14] Anonymous: *IBM WebSphere Portal Version 5 Application Development. Student Notebook*. Studijní curriculum IBM, course code SW531. IBM Corporation, 2004.
- [15] Anonymous: *IBM WebSphere Portal Version 5 Application Development. Student Exercises*. Studijní curriculum IBM, course code WF531. IBM Corporation, 2004.
- [16] Alcott Tom: *WebSphere V5 Architecture and Technology Overview*. IBM WebSphere technical Conference, Innsbruck 2003.
- [17] Botzum Keys: *WebSphere 5.0 Security – Hardening*. IBM WebSphere technical Conference, Innsbruck 2003.
- [18] Botzum Keys: *WebSphere 5.0 Security – Advanced Topics*. IBM WebSphere technical Conference, Innsbruck 2003.

- [19] Schuster Ingo: *WebSphere Portal – Setting Up a Secure Portal*. IBM WebSphere technical Conference, Innsbruck 2003.
- [20] McGuinness Keith: *WebSphere Application Server V5 Administration, Architecture and Topology*. IBM WebSphere technical Conference, Innsbruck 2003.
- [21] Polleck Ken: *Project Managing Your WebSphere Portal Implementation*. IBM WebSphere Portal and Lotus Workplace Technical Conference, Stuttgart 2004.
- [22] Buehler Dieter: *WebSphere Portal Security Administration and Role Based Access Control*. IBM WebSphere Portal and Lotus Workplace Technical Conference, Stuttgart 2004.
- [23] Botzum Kyes, Neuchterlein Bruce: *Advanced Configuration and Performance Tuning of The Portal Server 5.0 User Registry*. IBM WebSphere Portal and Lotus Workplace Technical Conference, Stuttgart 2004.
- [24] Botzum Keys, Singh Irina: *WebSphere Portal V5.X Security with Tivoli Access Manager v5.1*. IBM WebSphere Portal and Lotus Workplace Technical Conference, Stuttgart 2004.
- [25] Smolej Dušan: *Metodologie portálu veřejné správy*. Příspěvek na setkání portálové skupiny CESNET, Pardubice 2004.
- [26] Schellman Peter: *Implementing a University-wide Portal. The European Perspective*. Conference PEPC2004, Nottingham 2004.
- [27] Agarwal Sarah, Browning Paul: *Focusing on Users: Case Study of Portal User Requirements Analysis and User-testing Sessions at The University of Bristol*. Conference PEPC2004, Nottingham 2004.
- [28] Viceic Predrag, Gartel Philippe: *MyUNIL Portal*. Conference PEPC2004, Nottingham 2004.
- [29] Percoco Joseph, Thompson William: *Roadmap of myRudgers – a University Portal*. Conference PEPC2004, Nottingham 2004.
- [30] Moore Graham, Clark David, Zackrisson Mike: *Implementing uportal via the SCT Luminis Portal Family*. Conference PEPC2004, Nottingham 2004.
- [31] Ron Tell: *Make the portal your own!* IBM WebSphere Portal and Lotus Workplace Technical Conference, Stuttgart 2004.
- [32] Thomas Shaeck: *WebSphere Portal Foundation Architecture*. IBM WebSphere Portal and Lotus Workplace Technical Conference, Stuttgart 2004.
- [33] IBM DeveloperWorks: *Understanding How WebSpherePortal Displays Portal Pages*.
- [34] IBM DeveloperWorks: *Creating a Distinctive Look and Feel for Your Portal*.
- [35] Kolektiv: *IBM WebSphere Portal Version 5.0 Tuning Guide* Whitepaper. IBM Corporation, 2003.
- [36] Kolektiv: *Member Services in WebSphere Portal: Setting up a custom user data store*. Whitepaper. IBM Corporation, 2003.
- [37] G. Hohpe, B.Wolf: *Enterprise Integration Patterns*. ISBN: 03-212-0068-3, Addison-Wesley 2003.
- [38] Meilir Page-Jones: *Základy objektově orientovaného návrhu v UML*. ISBN 80-247-0210-X, Grada 2001.
- [39] Jim Arlow, Ila Neustadt: *UML a unifikovaný proces vývoje aplikací*. ISBN 80-7226-947-X, Computer Press.
- [40] Petr Václavek: *JavaScript Hotová řešení*. ISBN 80-7226-854-6, Computer Press.
- [41] Jim McCarthy: *Softwarové projekty*. ISBN 8072261940, Computer Press.
- [42] Snížek Martin: *CSS pro zelenáče*. ISBN 80-86330-14-1, Neocortex.
- [43] Jack Shirazi: *Java - vyladování výkonu*. ISBN 80-247-0752-7, Grada 2003.

Literatura

- [44] E. Gamma, R. Helm, R. Johnson, J. Vlissides: *Návrh programů pomocí vzorů*. ISBN 80-247-0302-5, Grada 2003.
- [45] Martin Fowler a kolektiv: *Refaktoring*. ISBN 80-247-0299-1, Grada 2003.
- [46] Piotr Wróblewski: *Algoritmy*. ISBN 80-251-0343-9, Computer Press.

Publikace a vystoupení na konferencích

- [ZCU-1] Valdman Jan: *Studie portálových řešení*. ZČU, Plzeň 2003.
- [UNIN04] Valdman Jan, Rychlík Jan: *Úloha portálu v projektu eUNIVERZITA Západočeské univerzity v Plzni*. Příspěvek na konferenci UNINFOS'04. STUBA, Bratislava 2004.
- [CPG-1] Valdman Jan: *Co nového v portálech*. Příspěvek na setkání portálové skupiny Cesnet. Pardubice 2004.
- [CPG-2] Otta Maxmilián: *Portály v opensource*. Příspěvek na setkání portálové skupiny Cesnet. Pardubice 2004.
- [CPG-3] Šimonek Jiří: *Vývoj portletů pomocí WSAD+PT*. Příspěvek na setkání portálové skupiny Cesnet. Pardubice 2004.
- [CPG-4] Šimonek Jiří: *Srovnání IBM API a JSR168*. Příspěvek na setkání portálové skupiny Cesnet. Pardubice 2004.
- [CPG-5] Valdman Jan: *PORTA@L – Grant FR CESNET*. Příspěvek na setkání portálové skupiny Cesnet. Pardubice 2004.
- [EO-1] Valdman Jan: *Co je to portál? Výběr portálového projektu a implementace portálu ZČU*. Příspěvek na XXIV. konferenci Europen.CZ. Kralický Sněžník 2004.
- [EO-2] Otta Maxmilián: *Open Source portály*. Příspěvek na XXIV.konferenci Europen.CZ. Kralický Sněžník 2004.
- [LPPL04] Valdman Jan: *Návrh portálu ZČU metodikou DDT*. Příspěvek na konferenci LPPL'2004. ZČU, Plzeň 2004.
- [ZCU-2] Gallistl Zdeněk: *Vývoj aplikací pro WebSphere Portal*. Diplomová práce obhájená na Katedře informatiky a výpočetní techniky, FAV ZČU. Vedoucí Jan Valdman, oponent Jiří Šimonek. Plzeň 2004.
- [ZCU-3] Halas Pavel: *Katedrální portál v technologii WebSphere*. Diplomová práce obhájená na Katedře informatiky a výpočetní techniky, FAV ZČU. Vedoucí Přemysl Brada, oponent Jan Valdman. Plzeň 2004.
- [OSU-1] Šimonek Jiří: *Portál Ostravské univerzity*. Skripta pro distanční vzdělávání. OSU, Ostrava 2004.
- [UN0105] Valdman Jan: *Seznámení s portálem ZČU*. Univerzitní Noviny, č. 1-XI, str. 10-11. Vydává Západočeská univerzita v Plzni. Plzeň 2005.
- [UN0904] Šimonek Jiří: *Portál Ostravské univerzity*. Listy Ostravské univerzity, č. 9, ročník 12, str. 5-6. Vydává Ostravská univerzita. Ostrava 2004.
- [OSU-2] Šimonek Jiří: *Standardy ve vývoji portletů*. VI. konferencia doktorandov a mladých vedeckých pracovníkov. Univerzita Konštantýna filozofa v Nitre. Nitra 2005.

Část III

Přílohy

Příloha **A**

Řešitelské týmy

A.1. Západočeská univerzita v Plzni

ZÁPADOČESKÁ UNIVERZITA Úsek prorektora pro informatiku
Univerzitní ul.8, 306 14, tel. 377 63 1055

Portál ZČU - pilotní projekt

Téma : Adresář řešitelů

RADA PROJEKTU

Doc. RNDr. František JEŽEK CSc. rozvoj@rek.zcu.cz 1040 UR411
Prorektor pro strategii a rozvoj
Dr. Ing. Jan RYCHLÍK rychlik@civ.zcu.cz 1050 UR211
Prorektor pro informatiku
Ing. Vladimír RUDOLF dolf@civ.zcu.cz 2700 UI126
Ředitel CIV

VEDOUcí PROJEKTU

Rndr. Daniela Šafaříková safarid@rek.zcu.cz 1055 UR211
Úsek Prorektora pro informatiku

ŘEŠITELÉ

Za dodavatele technologie a SW - Centrum informatizace a výpočetní techniky

Ing. Jan VALDMAN Ph.D. valdman@civ.zcu.cz 2834 UI403
Ing. Maxmilián OTTA Ph.D. otta@civ.zcu.cz 2812 UI301
Ing. Petr GROLMUS indy@civ.zcu.cz 2851 UI424

Za poskytovatele (i konzumenty) informací

Ing. Přemysl BRADA brada@kiv.zcu.cz 2435 UK409
KIV FAV
Doc. Ing. František VÁVRA CSc. vavra@kiv.zcu.cz 2409 UK408
KIV FAV
Prof. Ing. Jiří ŠAFAŘÍK CSc. safarikj@kiv.zcu.cz 2439 UK413
KIV FAV
Ing. Jan ČEPIČKA Ph.D. cepicka@kma.zcu.cz 2624 UK609
KMA FAV
Ing. Radka TICHÁ tich@uk.zcu.cz 7726 UB206
Univerzitní knihovna
Ing. Barbora KATOLICKÁ bar@uk.zcu.cz 7727 UB206
Univerzitní knihovna
Jarmila SMAZALOVÁ smazalov@rek.zcu.cz 1070 UR413
Rektorát
Irena RABOCHOVÁ rabochov@rek.zcu.cz 1071 UR412
Rektorát

Za převážně konzumenty informací

Marek Horák mh@slunicko.net
Student FAV
Antonín Předota apredota@students.zcu.cz
Student FEL
Tomáš Wünsch towunsch@students.zcu.cz
Student FAV

A.2. Univerzita Palackého v Olomouci

UNIVERZITA PALACKÉHO V OLOMOUCI
Křížkovského 8, 771 47 Olomouc

Portál UP - pilotní projekt

Téma : Adresář řešitelů

RADA PROJEKTU

Doc. RNDr. Richard Horák CSc.	richard.horak@upol.cz	585634254
Prorektor pro záležitosti organizace a informačních systémů		
RNDr. František Zedník	frantisek.zednik@upol.cz	585631701
Ředitel CVT		

ŘEŠITELÉ

RNDr. Hana Matochová	hana.matochova@upol.cz	585631821
Mgr. Kateřina Rašková	katerina.raskova@upol.cz	585631890
Mgr. Petr Kolašín	petr.kolasin@upol.cz	585631819
Bc. Petr Čírtek	petr.cirtek@upol.cz	585631816

A.3. Ostravská univerzita v Ostravě

OSTRAVSKÁ UNIVERZITA V OSTRAVĚ
Bráfova 5, 701 03

Úsek prorektora pro informatiku

Portál OSU - pilotní projekt

Téma : Adresář řešitelů

RADA PROJEKTU

Doc. RNDr. Dalibor Dvořák, CSc.	dalibor.dvorak@osu.cz	197
Prorektor pro informatiku		
Mgr. Martin Malčík	martin.malcik@osu.cz	132
Ředitel CIT		

VEDOUCÍ PROJEKTU

Mgr. Martin Malčík	martin.malcik@osu.cz	132
Ředitel CIT		

ŘEŠITELÉ

Za dodavatele technologie a SW - Centrum informačních technologií

Mgr. Jiří Šimonek	jiri.simonek@osu.cz	133
Ing. Robert Bohoněk	robert.bohonek@osu.cz	130
Ing. Daniel Holešínský	daniel.holesinsky@osu.cz	130
Bc. Marek Trembacz	marek.trembacz@osu.cz	118
Mgr. Martin Haft	martin.haft@osu.cz	123

Za poskytovatele (i konzumenty) informací

Příloha A. Řešitelské týmy

Tomáš Kamrád CIT	tomas.kamrad@osu.cz	134
Ing. Jaroslav Švehlák CIT	jaroslav.svehlak@osu.cz	137
Doc. Ing. Cyril Klimeš, CSc. KIP	cyril.klimes@osu.cz	238
Mgr. Jaroslav Knybel KIP	jaroslav.knybel@osu.cz	204
Martin Hasník CIT	martin.hasnik@osu.cz	115
Ing. Jana Poloková FPD	jana.polokova@osu.cz	163
Dagmar Měšťánková	dagmar.mestankova@osu.cz	187

Za převážně konzumenty informací

Dagmar Milátová Studijní oddělení FPD	dagmar.milatova@osu.cz
Natalie Bosová Studijní oddělení FPR	natalie.bosova@osu.cz

Příloha **B**

JSP a IBM Portlet API

V této kapitole si přiblížíme základní technologie použité v IBM WebSphere Portal V5.0. Vysvětlíme si princip Java Servletů a jejich rozdíly oproti portletům. Přiblížíme si technologii JSP. Vysvětlíme si princip obecné MVC architektury a Portlet MVC. Popíšeme si životní cyklus portletu a Portlet API.

B.1. Java Servlety

Java servlety představují velmi rozsáhlou, na platformě nezávislou, technologii pro rozšíření funkcí webových serverů. Ryzí webové nebo aplikační Java servery mohou servlet kontejner implementovat jako vlákno uvnitř hlavního procesu. Oblíbenější webové servery jako *Apache*, *iPlanet* nebo *IIS* vyžadují odlišné technologie v závislosti na jejich vlastních rozhraních. Architektura servletů zahrnuje také zásuvný modul webového serveru, který přeměňuje požadavky servletu do odděleného procesu Javy, jež je implementací servlet kontejneru. Servlet kontejner komunikuje se zásuvným modulem tím způsobem, že odposlouchává nastavitelný síťový port, který je vyhrazen pro přeměrování požadavků.

B.2. JavaServer pages

Používá se v technologii *server-side scripting* (kód interpretovaný na straně serveru). JSP doplňuje architekturu Java servletů. Poskytuje JSP kontejner, což je proces na straně serveru. Zajišťuje správu JSP stránek a jejich překlad na servlety. JSP stránka se podobá stránce v jazyce HTML, obsahuje však zvláštní syntaktické elementy, díky kterým může stránka dynamicky měnit svůj obsah viz. kapitola B.2.3..

B.2.1. JSP kontejner

J2EE (Java 2 Enterprise Edition) definuje několik kontejnerů včetně kontejneru JSP, servlet kontejneru a *Enterprise JavaBeans* kontejneru. Kontejner je v žargonu objektivě orientovaného programování třída, nebo komponenta, která uspořádává ostatní třídy nebo komponenty. Specifikace JSP tento původní význam rozšiřuje. Kontejnery J2EE poskytují úplné aplikační prostředí, kde řídí životní cyklus komponent a poskytují jim různé služby.

Kontejnery J2EE nemohou pracovat korektně, pokud programátor nepoužívá software komponenty v souladu s pravidly definovanými kontejnerem. Vývojář musí tato pravidla respektovat, neboť je kontejner v různých fázích komponent předpokládá. Specifikace JSP zdůrazňuje význam pravidel tím, že je nazývá *dohodami*. Kontejnery dodržování *dohod* řeší tak, že vyžadují po aplikaci implementaci přesného Java rozhraní.

Každý kontejner J2EE poskytuje služby jen těm komponentám, za které má zodpovědnost. JSP kontejner překládá JSP stránky do kódu Java servletů a výsledek poté překládá a načítá do servlet kontejnerů. Dále také koordinuje vzájemný vztah mezi servlet kontejnerem a přeloženými JSP stránkami. Servlet kontejner poskytuje aplikační prostředí pro Java servlety.

B.2.2. Postup zpracování JSP

Pokud webový prohlížeč zažádá o stránku JSP, vyvolá sled následujících operací.

1. Interpretuje se JSP stránka
2. Vygeneruje se Java servlet

3. Servlet se pomocí standardního překladače, který je dodán s JSP kontejnerem, převede do bajtového kódu Java.
4. Servlet je načten do servlet kontejneru (virtuálního stroje Java - JVM).
5. U servletu je vyvolána služební metoda

Požaduje-li prohlížeč následně tutéž stránku a nebyla-li stránka od posledního volání změněna, musí JSP kontejner provést pouze krok (5). Pokud se stránka změnila, je nutné před odesláním odpovědi znovu zopakovat všechny kroky. Zpracováním kroků (1) až (4) lze vysvětlit, proč mají JSP stránky pomalejší odezvu při prvním zobrazení v prohlížeči.

B.2.3. JSP direktivy

JSP stránka může volat metody příslušného kontejneru s popisem akcí, jež má kontejner v její prospěch vykonat. Tyto zprávy se nazývají *direktivy*. Direktivy podle specifikace JSP 1.1 mají tento formát :

```
<%@ direktiva {atribut = "hodnota"} %>
```

Direktiva začíná značkou <%@, poté následuje název direktivy a případně několik atributů, vždy jako dvojice název-hodnota, a končí značkou %>. Direktiva na JSP stránce způsobí, že JSP kontejneru vyšle požadavek na provedení určité služby. Ve vygenerovaném dokumentu se však přímo neobjeví.

Specifikace JSP 1.1 definuje tři typy direktiv: `include`, `page` a `taglib`. Direktiva `page` umožňuje řídit strukturu servletu importováním tříd, upravením třídy servletu (překlad), nastavením typu obsahu apod. Direktiva `page` se může v dokumentu nacházet kdekoliv. Druhá direktiva `include` umožňuje vložit do třídy servletu soubor v době, kdy se soubor JSP překládá do servletu. Direktiva `include` by měla být v dokumentu na místě, kam chceme soubor vložit. JSP 1.1 zavádí třetí direktivu `taglib`, kterou lze použít pro definování uživatelských značek (tagů).

B.2.4. JSP skriptovací značky

Skriptovací značky JSP umožňují vložit kód do servletu, který se vytvoří z jsp stránky.

Existují tyto tři:

<%= výraz %> Hodnota výrazu se vloží do výstupu servletu.

<%= kód %> Vkládání skriptu do metody servletu `_jspService`.

<! kód %> Deklarace, která se vkládá do těla servletu mimo všechny existující metody.

B.3. Standard MVC

MVC (*Model - View - Controller*) je léty prověřený přístup k vývoji GUI aplikací nezávislý na technologii. Aplikace je rozložena na tři části.

Model: reprezentuje funkční logiku aplikace. V programovacím jazyce Java je z pravidla reprezentován Java beans, EJB nebo jinými třídami. Musí být nezávislý na ostatních komponentách MVC.

View: slouží pro interakci s uživatelem. Je zodpovědný za vizuální zpracování výsledku aplikace.

Controller: je zodpovědný za zpracování uživatelských požadavků. Rozhoduje, které modely ke zpracování použít.

B.4. Portlet MVC

Architektura portletů se od standardní MVC architektury v podstatě neliší. Samotný Portal Server je v souladu s MVC navržen.

Model: Model v portletových aplikacích se neliší od modelu v jiných Java Server side aplikacích¹. Řízení může mít na starosti servlet, portlet, nebo jednoduchá Java třída.

View: Podobně jako servlet MVC implementace, je i View tradičně implementován s využitím JSP nebo prostého HTML. Protože HTML, které View mód vrací, bude pouze do konečné stránky začleněno, nemusí obsahovat tagy na úrovni stránek. Portlet API poskytuje velmi užitečné knihovny JSP tagů.

Controller: Controller je odpovědný za detekci vyžádaného režimu portletu, vykonání patřičných obslužných metod a zprostředkování správného obsahu. Třída portletu sama působí jako controller.

B.5. Portlet API

Portlet je aplikace na straně serveru zasazená do kontextu portálu. Je základním stavebním prvkem finální portálové stránky. Podobnost se servlety není náhodná. Portlet je potomkem `javax.servlet.http.HttpServlet` třídy. Je realizován v Portlet kontejneru spravovaném aplikačním serverem.

Ačkoli je portlet schopen zajistit dvojí funkčnost (funkce servletu i portletu), je rozhodně lepší nepoužívat je současně. Portlet je viditelný na portálové stránce jako orámovaný prostor s určitým obsahem, kterých portál může obsahovat mnoho. Rám portletu je záležitostí portálu.

B.5.1. Pojmy používané v souvislosti s portlety

Objasníme si základní pojmy, které se v souvislosti s portlety používají.

- **Rám portletu**

Tim je myšlen rám, která obklopuje portlet. Součástí je název a několik tlačítek podobně, jako u rámu klasické aplikace provozované v grafickém prostředí operačního systému.

- **Stav**

Stavem se rozumí způsob zobrazení daného portletu. Může být standardní, minimalizovaný, nebo maximalizovaný.

- **Mód**

Portlet může přecházet mezi módy *View*, *Edit*, *Configure* a *Help*. Přístup ke každému z módů je závislý na konkrétních právech uživatele, které definuje privilegovaný uživatel.

Význam jednotlivých módů:

¹Java aplikace běžící na straně serveru

- **View mód**
Jde o základní mód, kde portlet zobrazuje obsah, pro který byl určen.
- **Edit mód**
Do *Edit* módu se lze přepnout stejnojmenným tlačítkem. Umožňuje měnit některé konfigurační parametry. Využívá se především pro potřeby samotného uživatele k dodatečnému nastavení obsahu *View* módu (např. město pro zobrazování počasí).
- **Configure mód**
Configure mód je koncepčně podobný s *Edit* módem. Oprávněný uživatel má možnost do jisté míry konfigurovat portlet (např. parametry přístupu k databázi, či jiné). Sem se umísťují informace, které pro běžného uživatele nesmějí být přístupné.
- **Help mód**
Mód zpravidla obsahuje informace o konkrétním portletu a uživatelský manuál.

Portlety mohou přistupovat k různým informacím prostřednictvím objektů *Portlet API*, které umožňují detekovat stav portletu. *PortletState* informuje portlet kdykoliv jej uživatel minimalizuje, maximalizuje, nebo jen požaduje obnovit (nejběžnější stav). Portlet by měl upravit obsah, aby odpovídal požadovanému stavu. V maximalizovaném tvaru může portlet například využít většího prostoru a přeskupit, nebo rozšířit původní obsah.

B.5.2. Sestavení stránky

Portlet má možnost, kromě zápisu do vlastní orámované oblasti, psát i do horní části portálové stránky, která je pro tyto účely vyhrazena. Možnosti se využívá především pro informační hlášení, nebo vkládání dodatečných JavaScriptů. Při programování je dobré uvědomit si, že tuto schopnost mají všechny portlety apodle toho svou aplikaci přizpůsobovat. Tím se především myslí již zmiňovaná jedinečnost názvů v JavaScriptech.

B.5.3. Meziportletová komunikace

Portlety mohou spolupracovat jeden s druhým v rámci jediného požadavku, pokud jsou součástí stejné aplikace. Pomocí zpráv mohou ovlivňovat jeden druhého. Portlet může například informovat jiné portlety v rámci jedné aplikace o konkrétních změnách, jež uživatel provedl. Naslouchající portlety mohou odpovídajícím způsobem reagovat a přizpůsobovat svůj obsah.

B.5.4. Obsluha událostí

Události jsou v podstatě výjimkami v terminologii programovacího jazyka Java. Například, při kliknutí uživatele na tlačítko, se generuje událost, která v portletu vyvolá obslužnou metodu dané události. Portlet API umožňuje pracovat např. s událostmi **WindowEvents**² a **MessageEvents**³.

B.5.5. Bezpečnost

Portlet pracuje pouze v kontextu portálového serveru a nemůže být volán uživatelem odděleně. Portal Server je odpovědný za autorizaci a autentizaci uživatele.

²Událost vyvolaná změnou okna

³Příchod zprávy

Přístupová práva se nastavují v administrační části portálu. Pro programátora tak odpadá starost o přístupová práva a do jisté míry i řešení bezpečnosti.

B.5.6. Portletová aplikace

Portletová aplikace je skupina portletů, které spolu logicky souvisí. Nejjednodušší aplikací může být samotný portlet, jako například portlet pro sledování počasí. Aplikace Exchange2000 obsahuje například portletů pět. Portlety obsažené ve stejné aplikaci sdílejí nastavení konfiguračních parametrů v *deployment descriptoru*. Parametry může administrátor během života aplikace měnit.

B.5.7. Srovnání servletu a portletu

Servlet je základem portletu. Zde si však ukážeme několik odlišností. Při návrhu portletové aplikace je třeba brát tyto odlišnosti v úvahu.

Práce se servlety je oproti portletům snadnější, neboť servlety generují celou webovou stránku, která se zasílá klientovi. Programátor servletu má pod kontrolou generování HTML kódu celé webové stránky.

Portlet naopak představuje pouze část webové stránky. Na stejné stránce může existovat několik portletů současně, které spolu nemusí přímo souviset. V rozsáhlejších aplikacích vyžaduje řešení komunikaci mezi portlety. Programátor se musí držet zásady, že jeden portlet nesmí ovlivňovat funkčnost ostatních portletů (s výjimkou portletů stejné aplikace) a proto musí zajistit jedinečnost názvů proměnných a funkcí nejen JavaScriptů jednotlivých portletů. Každý portlet je evidován pod jedinečným ID, aby byl portál schopen portlety rozlišovat.

B.6. Základní objekty portletu

V této části si popíšeme většinu objektů Portlet API. Vysvětlíme si jejich význam, vyjmenujeme si jejich metody a ukážeme si názorně použití.

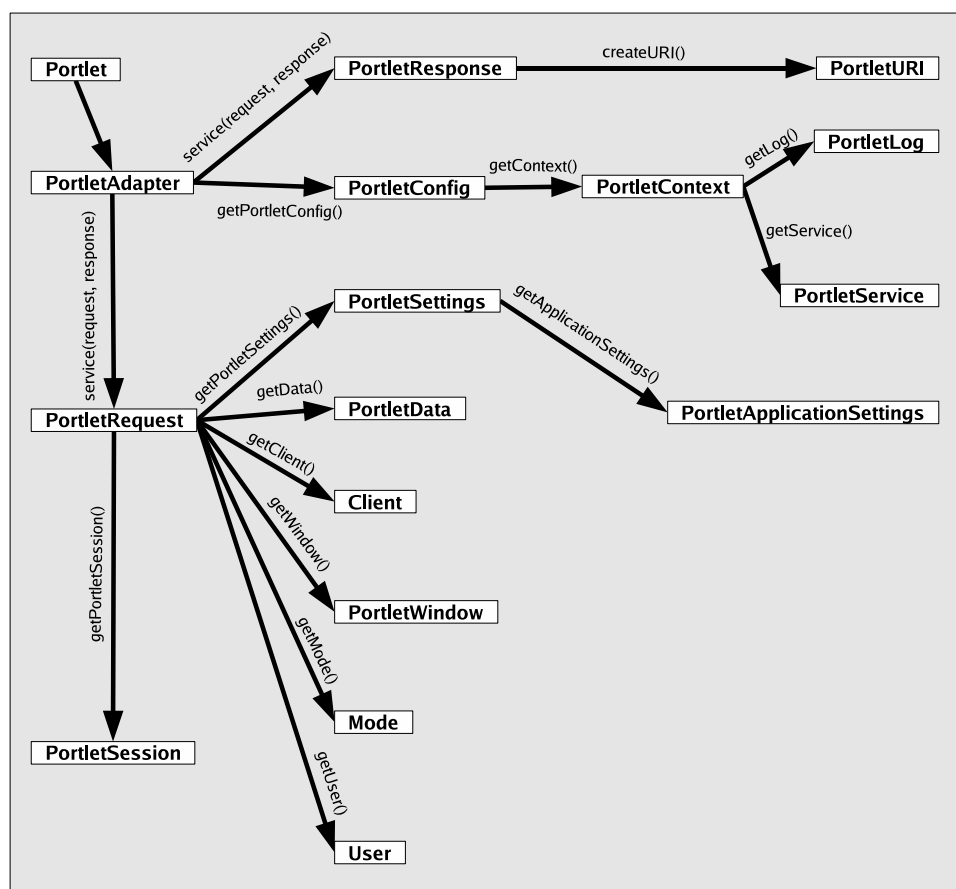
B.6.1. Portlet

Tato abstraktní třída definuje základní metody související s životním cyklem portletu jako např. `init()`, `service()` a `destroy()`. Pro přesnost je třeba uvést, že jsou tyto abstraktní metody implementovány ve třídě `PortletAdapter`. `PortletAdapter` poskytuje obslužné metody se základními funkcemi pro rozpoznání typu požadavku a předání požadavku odpovídající metodě `doView()`, `doConfigure()`, `doHelp()`, nebo `doEdit()`.

B.6.2. PortletAdapter

Třída je považována za základní implementaci třídy `Portlet`. Doporučuje se používat tuto třídu spíše, než třídu `Portlet`.

Metody `getVariable()`, `setVariable()` a `removeVariable()` umožňují práci s proměnnými, které můžeme nastavit v rámci konkrétního portletu. Je důležité mít na paměti, že proměnné jsou na určité úrovni a nebudou sdíleny s jinými portlety. Proměnné jsou přístupné jen z programu, není možné s nimi pracovat v administrační části portálu.



Obrázek B.1: Objekty Portlet API

Příklad použití :

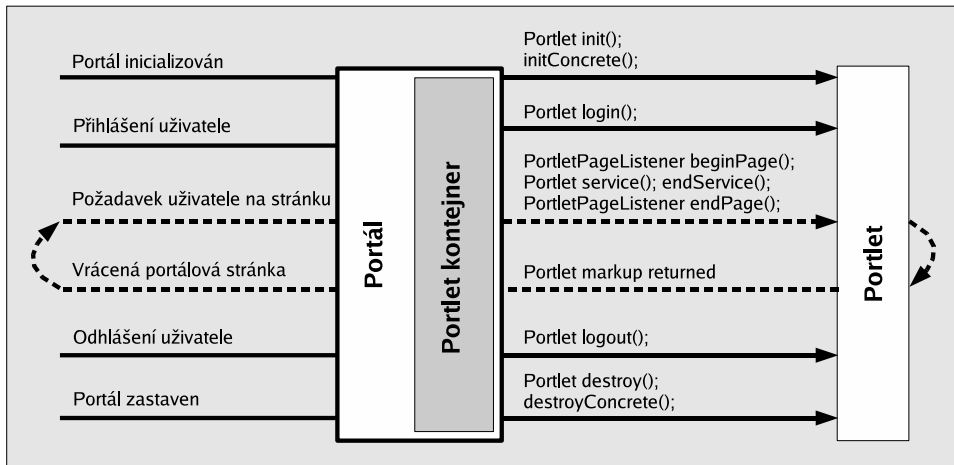
```

setVariable("promenna", "hodnota");
String vlastniPromenna = (String) getVariable("promenna");
    
```

B.6.3. PortletRequest

PortletRequest je potomkem třídy HttpServletRequest. Reprezentuje požadavek uživatele podobně, jako ServletRequest. Obsahuje informace o uživateli a klientu. Metody PortletRequest:

- `getAttribute() / setAttribute() / removeAttribute()`
Metoda ukládá v podobě krátkých výrazů. PortletRequest je specifický pro každý portlet, proto data uložená tímto způsobem nejsou dalšími portlety sdílená.
- `getParameter()`
Metoda zpřístupňuje parametry získané přes HttpServletRequest.



Obrázek B.2: Životní cyklus portletu

- `getCookie()`
Metoda umožňuje přístup ke *cookies* uloženým výchozí doménou na straně klienta. Je vrácen objekt pole *cookies*.
- `getHeader()`
Metoda zpřístupňuje HTTP hlavičky zasílané klientem.
- `getLocale()`
Metoda vrací preferovanou jazykovou lokalizaci uživatele. Ta se definuje při registraci nového uživatele v portálu.
- `getPreviousMode()`
Metoda je schopna zjistit předchozí mód portletu. Ve WebSphere Portalu V 4.1.2 vrací vždy `null`.

Ukázka práce s objektem `PortalRequest`:

```
// Nastavení hodnoty proměnné uri
request.setAttribute("uri", uri);

// Získání celého jména uživatele
String celeJmeno = request.getParameter("f_name");

// Získání preferované lokalizace uživatele
java.util.Locale lokalizace = request.getLocale();
```

B.6.4. `PortletResponse`

Objekt `PortletResponse` je potomkem tříd `HttpServletResponse` a `ServletResponse`. Zapouzdřuje odpovědi, které předává Portal Serveru. Ten pak informaci zasílá klientovi jako součást webové stránky. U `ServletResponse` je situace odlišná. Odpověď se zasílá přímo klientovi. Některé z užitečných metod si nyní popíšeme.

- `getWriter()`
Metoda vrací `java.io.PrintWriter` objekt Portal Serveru. Data předávaná prostřednictvím objektu `PrintWriter` jsou zakomponována do obsahu portálové stránky na vyhrazené místo.
- `encodeNamespace()`
Vstupem metody je řetězec, k němuž se přiřadí identifikátor portletové aplikace jako prefix. (Např. název proměnné `variable_one` je zakódována jako `PC_175_variable_one`).
- `addCookie()`
Metoda umožňuje přidávat *cookie* do odpovědi klientu. Název *cookie* musí být v rámci portálu jedinečný. Doporučuje se použít k překódování metodu `encodeNamespace()`.
- `addHeader()/setHeader()/containsHeader()`
Metody umožňují pracovat s HTTP hlavičkou portálové stránky zasílané klientovi prostřednictvím Portal Serveru.
- `encodeURL()`
Metoda doplní vstupní řetězec do kompletního tvaru URL Portal Serveru. Např. řetězec `ukazka.gif`⁴ převede na `http://www.nazevdomeny.cz/wps/WPS_PA_351/ukazka.gif`.
- `createURI()/createReturnURI()`
Metody vytvoří URI objekt obsahující URL k jednotlivým módům portletu.

Ukázka práce s objektem `PortalResponse` :

```
// Vypis retezce na stranku portalu
java.io.PrintWriter out = response.getWriter();
out.println("Hello World");

// Vytvoreni URI
PortletURI uri = response.createURI();

// Zakodovny nazvu funkce Javascriptu pro jeji jedinecnost na strance
String nazevFunkce = response.encodeNamespace("funkce");
```

B.6.5. PortletSession

Objekt `PortletSession` je potomkem `HttpSession`. Je vytvořen při přihlášení uživatele. Používá se pro udržení obsahu proměnných mezi jednotlivými požadavky. Data nejsou přístupná pro jiné portlety, jsou jedinečná pro každého uživatele podobně, jako

`PortletData` viz. kapitola B.6.11.. Metodu `getPortletSession()` lze použít i k vytvoření *session* anonymního uživatele.

Ukázka použití :

```
PortletSession session = request.getPortletSession();
```

Mnohem důležitějšími metodami pro práci se *session* jsou :

⁴Relativní cesta k obrázku

- `getAttribute()/setAttribute()/removeAttribute()`
Metody umožňují čtení, zápis a rušení objektů v `PortletSession`.

B.6.6. Client

Nese informace o zařízení, ze kterého pochází požadavek. Nejedná se o osobní údaje daného uživatele. Objekt může být získán od objektu `PortletRequest`. Ukážeme si několik příkladů práce s objektem `Client`.

```
Client client = request.getClient();
out.print("Firma : " + client.getManufacture() + "<BR>");
out.print("MimeType : " + client.getMimeType() + "<BR>");
out.print("Model : " + client.getModel() + "<BR>");
out.print("Prohlížeč : " + client.getUserAgent() + "<BR>");
out.print("Verze : " + client.getVersion() + "<BR>");
```

B.6.7. PortletConfig

Atributy uložené v objektu `PortletConfig` jsou sdíleny všemi kopiemi téhož portletu. Objekt lze použít pro operace nad souborem `web.xml`. Umožňuje detekovat podporované módy portletu. Nejvyužívanějšími metodami jsou:

- `supports()`
Jako parametr metody se uvádí název módu portletu, metoda vrací logickou hodnotu, zda je mód podporován, či nikoliv.
- `getContext()`
Metoda vrátí objekt `PortletContext` popsáný v kapitole B.6.8..

B.6.8. PortletContext

Objekt zpřístupňuje portletu služby *Portlet kontejneru*, ve kterém portlet běží (např. `PortletLog`, `Credentials Vault`, `PersistentConnection` a další). Parametry, které jsou prostřednictvím `PortletContext` zpřístupněny, pocházejí ze souboru `web.xml`. `PortletContext` je potomkem objektu `PortletConfig`. Hlavními metodami jsou:

- `include()`
V dobře navrženém MVC vykoná portlet jednu, nebo více úloh pro obsluhu požadavku. Po dokončení je volán JSP pro zpracování výstupní formy.
- `getContainerInfo()`
Metoda vrací verzi Portal Serveru.
- `getText()`
Metoda na základě překladového souboru NLS (National Language Support) překládá konkrétní řetězec do jazyka, jenž je specifikován parametrem metody.

B.6.9. PortletSettings

Data uložená v objektu `PortletSettings` jsou trvalého charakteru sdílená všemi uživateli portletu. Ukládají se do souboru `portlet.xml` uzavřené v tagu `<concrete-portlet>`. Slouží pro uchování konfiguračních dat portletu. Hodnoty mohou být upravovány v průběhu života portletu. Privilegovaný uživatel může hodnoty měnit v administrační části portálu. Objekt `PortletSettings` se zpřístupní přes `PortletRequest`, nebo je předáván jako parametr metody `initConcrete()` a `destroyConcrete()`. Práce s objektem je možná pouze v rámci metody `doConfigure()`! Základními metodami jsou:

- `getAttribute()/setAttribute()/removeAttribute()`
Metody slouží pro práci s konfiguračními daty.
- `getTitle()`
Metoda vrací řetězec, který představuje název portletu pro daného klienta a jeho specifickou lokalizaci. Název může administrátor měnit v administrační části portálu.
- `getDefaultLocale()`
Metoda vrací výchozí lokalizaci definovanou v souboru `portlet.xml`.
- `getPortletApplicationSettings()`
Metoda vrací objekt `PortletApplicationSettings`.

Příklad použití :

```
// Deklarace promenne a zapis hodnoty
request.getSettings().setAttribute("promenna", "hodnota" );

// Zapis do portlet.xml
request.getSettings().store();
```

Poznámka: Každý definovaný atribut, kterému jsme přiřadili hodnotu, musíme zvlášť ukládat metodou `store()`.

B.6.10. PortletApplicationSettings

Data uložená v objektu `PortletApplicationSettings` mají stejné vlastnosti, jako data v objektu `PortletSettings` viz. kapitola B.6.9.. Zapisují se do souboru `portlet.xml` uzavřené do tagu `<concrete-portlet-app>`. Slouží jako konfigurační parametry společné pro všechny portlety jedné portletové aplikace. Zápis parametrů je možný pouze v metodě `doConfigure()`, nebo `actionPerformed()`.

Příklad použití :

```
// Definovani promenne a její hodnoty
request.getAppSettings().setAttribute("promenna", "hodnota" );

// Zapis do portlet.xml
request.getAppSettings().store();
```

B.6.11. PortletData

Objekt `PortletData` je potomkem třídy `ConcretePortlet`. Umožňuje rychle a bezpečně pracovat s daty trvalého charakteru bez potřeby JDBC. Data nejsou závislá na životním cyklu portletu. Jsou jedinečná pro každého uživatele. Dokud uživatel data během používání portletu nemodifikuje, jsou používány hodnoty definované privilegovaným uživatelem, který jako první portlet na stránky umístí a hodnoty nastaví. Data nesmí být ukládána s hodnotou `null`.

Zápis dat je možný pouze uvnitř metody `doEdit()`, nebo `actionPerformed()`!

Příklad použití :

```
// Ctení dat
PortletData data = request.getData();
String promenna = (String) data.getAttribute("promenna");

// Zapis dat - pouze v Edit modu.
PortletData data = request.getData();
data.setAttribute("promenna", "hodnota");
data.store();
```

B.6.12. PortletLog

Tento objekt umožňuje snadno zaznamenat informaci o chybě, nebo jinou informaci do `log`⁵ souboru. Název souboru je `wps_<časová značka>.log`. Nachází se v adresáři `<WPS portal adresář>/log`. Existují čtyři úrovně zápisu hlášení do `log` souboru: *info*, *debug*, *warn* a *error*. Význam jednotlivých úrovní plyne z následující ukázky:

```
PortletLog log = getPortletConfig().getContext.getLog();
if (log.isDebugEnabled()) log.debug("debug enabled:" + zprava);
if (log.isWarnEnabled()) log.debug("warn enabled:" + zprava);
if (log.isInfoEnabled()) log.debug("info enabled:" + zprava);
if (log.isErrorEnabled()) log.debug("error enabled:" + zprava);
```

B.6.13. PortletException

Objekt je potomkem třídy `ServletException`. Slouží jako základ většiny výjimek. Obsahuje `UnavailableException`.

B.6.14. UnavailableException

Vyjímek může nastat v průběhu inicializace portletu, kdy metoda `init()` volá metodu `super.init()` a dojde k chybě.

⁵Soubor, kam se zaznamenávají důležité informace týkající se činnosti portletu

- `getUnavailableSeconds()`
Metoda vrací celočíselnou hodnotu, která představuje čas, po který je portlet nedostupný.
- `isPermanent()`
Metoda vrací logickou hodnotu, která říká, zda je portlet permanentně nedostupný.

B.6.15. PortletWindow

Objekt reprezentuje rám obklopující portlet. Používá se především pro rozpoznání aktuálního stavu portletu (např. zda je maximalizován, nebo minimalizován).

B.6.16. User

Objekt reprezentuje autentizovaného uživatele. Je potomkem třídy `PortletRequest`. API obsahuje mnoho funkcí `get` a `set` pro práci s osobními údaji uživatele. Umožňuje pracovat se základními daty (LDAP) a zároveň s rozšiřujícími informacemi (Portal Server databáze).

Příklad použití :

```
String oblíbenábarva = user.getAttribute("favColor");  
String telefon = user.getAttribute("phoneNumber");
```

Tutorial: Vývoj portletů pro WebSphere Portal

Tato kapitola popisuje postup, jak naprogramovat jednoduchý portlet především pro portál WebSphere V5.0. Popíšeme si problémy, se kterými se programátor potýká při vývoji a ukážeme si příklady řešení. Krok po kroku si projdeme vývoj od začátku až ke zprovoznění v prostředí portálu.

Předpokladem je nainstalované WebSphere Studio Site Developer V 5.0.1. a nainstalovaný, nebo zpřístupněný WebSphere Portal V5.0.

Naše aplikace bude mít *View* mód, *Configure* mód a *Help* mód. Bude okénkem do jedné tabulky databáze. Configure režim bude určen k editaci položek tabulky (zápis nové, úprava existujících a rušení) a výběr jedné, která se zobrazí ve *View* módu. Nastavit parametry databáze bude moci pouze privilegovaný uživatel v administrátorském režimu portálu. *Help* mód bude obsahovat informace o portálu a návod k použití.

C.1. Hello world projekt

Přes menu `New→Project→Portlet development→Portlet Application Project` si vygenerujeme kostru jednoduchého projektu s pracovním názvem `HelloWorld`. Tlačítkem `Next` se dostaneme až k formuláři `Miscellaneous`, kde zaškrtneme volby `Add help mode` a `Add configure mode`. Tím si zajistíme, že náš portlet bude mít kromě `view` módu i `Edit` a `Help` mód.

C.2. První spuštění

C.2.1. Preview portlet

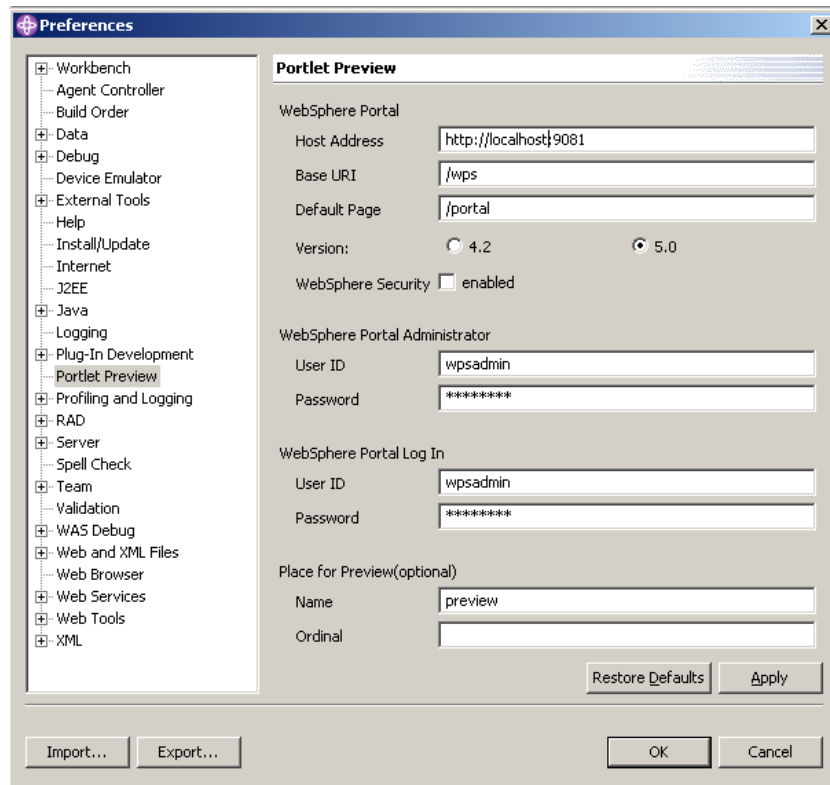
Tento způsob spuštění portletové aplikace je použitelný v případě, že potřebujeme projekt testovat na vzdáleném serveru nebo na lokálním portálu. Jako první krok si musíme nastavit parametry přístupu k portálu přes menu `Window→Preferences→Portlet Preview` viz. obrázek C.1

Pravým tlačítkem myši klikneme na projekt a vybereme možnost `Preview Portlet`. Automaticky se otevře okno prohlížeče s portletem.

C.2.2. Run on server

Podmínkou pro tento postup je lokálně nainstalovaný *WebSphere Portal Test Environment*. V perspektivě *Portlet*, kde na levé straně vidíme stromovou strukturu se soubory projektu, se můžeme přepnout přes záložku `Server Configuration` do druhého okna, kde se upravují nastavení testovacích serverů. Před samotným spuštěním portletu se musí vytvořit nový server kliknutím pravého tlačítka myši na položku `Servers→New→Server and Server Configuration`. Zvolíme `WebSphere Portal Version 5.0→Test Environment` viz. obrázek C.2.

Nyní máme nakonfigurovaný testovací server a zbývá nám definovat projekty, které budeme na portálu testovat. V zásadě můžeme najednou na stránce testovat několik různých projektů. Přepneme se přes záložku `J2EE navigator` do původního pohledu na projekt. Otevřeme si soubor `DefaultEAR→META-INF→application.xml`. Zobrazí se formulář pro editaci. Ve spodní části klikneme na záložku `Module` a přesvědčíme se, že máme v seznamu `war` souborů i náš `HelloWorld.war`. Pokud není, přes tlačítko `Add` snadno soubor přidáme. Vrátime se zpět ke konfiguraci serveru přes záložku `Server Configuration`. Pravým tlačítkem klikneme na náš vytvořený server a v menu vybereme položku `Add→DefaultEAR`. Nyní jsme s konfigurací serveru hotovi. Pravým tlačítkem klikneme na projekt a volbou `Run on Server`.



Obrázek C.1: Nastavení parametrů pro *portlet preview*

C.3. Uchování obsahu proměnných

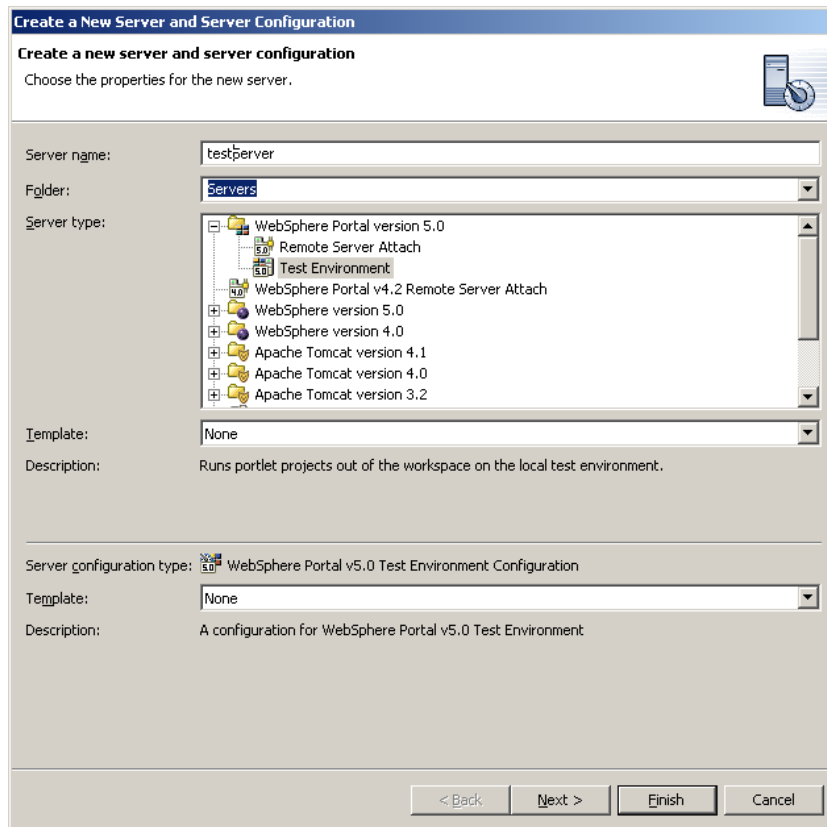
Podobně, jako standardní počítačový program potřebuje uchovávat konfigurační data a jiné informace nejen po dobu svého běhu, ale i pro příští spuštění, tak i naše portálová aplikace musí umět s daty podobně pracovat. Předvedeme si konkrétní ukázky některých způsobů práce s daty. Kompletní popis všech objektů pracujících s daty je uveden v kapitole B.6..

C.3.1. PortletSession

Data ukládaná v objektu PortletSession jsou pouze dočasná a jedinečná pro každého uživatele. K dispozici jsou jen v rámci jednoho přihlášení uživatele. Po odhlášení uživatele nenávratně zanikají.

Pro práci s daty tohoto typu slouží v našem projektu třída HelloWorldPortletSessionBean. Budeme-li chtít pracovat například s proměnnou typu String, jejímž obsahem bude název zobrazované položky databáze, doplníme třídu o globální proměnnou toShow a přidáme metody pro čtení a zápis viz. kapitola C.8.6..

Prakticky ve kterékoli části projektu můžeme proměnnou plnit a číst. To je vhodný způsob, jak předávat různé informace mezi kódem JSP a Javou.



Obrázek C.2: Nový server a jeho nastavení

```

HelloWorldPortletSessionBean sessionBean = getSessionBean(request);

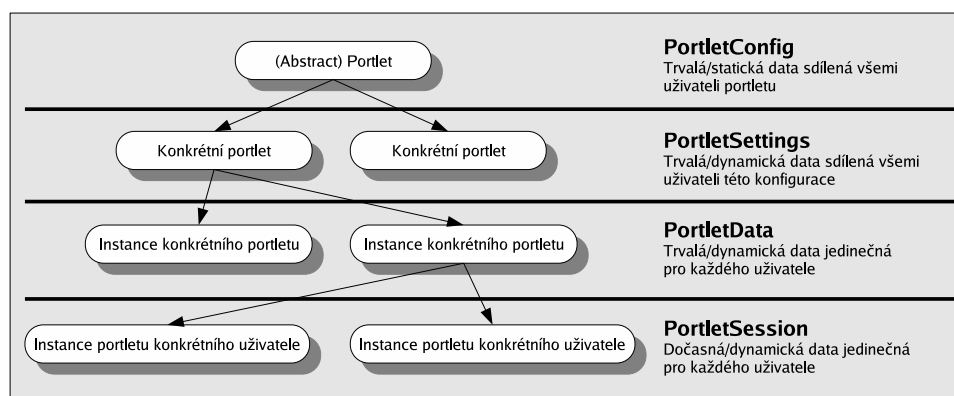
// Osetreni, kdyby doslo k chybe pri vytvareni objektu
if( sessionBean==null ) {
    response.getWriter().println("<b>NO PORTLET SESSION YET</b>");
    return;
}

// Zapis do session promenne
sessionBean.setToShow("obsah promenne");

// Cteni ze session promenne
String toShowString = sessionBean.getToShow();
    
```

C.3.2. PortletSettings

Data uložená jako `PortletSettings` jsou trvalého charakteru. Ukládají se přímo do souboru `portlet.xml` a jsou společná pro všechny uživatele. Ukládání je možné jen v rámci metody `doConfigure()` a `actionPerformed()`. Takto budeme ukládat ID položky v tabulce (`String toShow`), která se bude ve *View* režimu zobrazovat. Hodnotu sice máme uloženu jako `session` proměnnou, aby se však obsah po odhlášení neztratil, musíme jej uložit též v objektu `portletSettings`. Při opětovném přihlášení uživatele se v rámci inicializační metody portletu opět



Obrázek C.3: Parametrizace portletu

obsah do *session* proměnné načte (viz. úprava třídy `HelloWorldportlet`, kapitola C.8.1.).

```
// Zapis hodnoty do promenne
request.getPortletSettings().setAttribute("jmeno_promenne", "hodnota");
try {
    // Ulozeni hodnot
    request.getPortletSettings().store();
}
catch (IOException e) {
}

// Zpetne nacteni hodnoty do promenne typu String
String Promenna = settings.getAttribute("jmeno_promenne");
```

C.3.3. PortletData

Umíme ukládat trvalá konfigurační data (`PortletSettings`), ovšem některé aplikace vyžadují jedinečnost dat pro různé uživatele. K tomu slouží právě objekt `portletData`. Data se mohou ukládat pouze v rámci metody `doEdit()`, nebo `actionPerformed()`. V našem projektu tento způsob nevyužijeme, v jiných aplikacích však může být velmi užitečný.

```
// Ulozeni promenne
PortletData data = request.getData();
data.setAttribute("jmeno_promenne", "hodnota");
try {
    data.store();
}
catch (IOException e) {
}
```

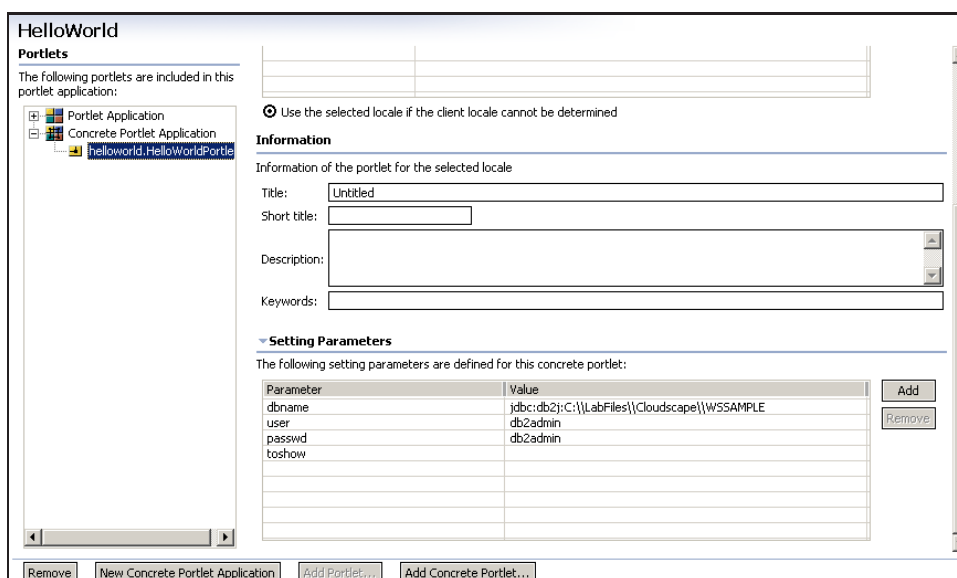
```
// Zpetne nacteni
String promenna = (String) request.getData().getAttribute("jmeno_promenne");
```

C.4. Konfigurace aplikace

C.4.1. portlet.xml

Soubor může sloužit k ukládání nejruznějších konfiguračních parametrů, která jsou společná pro všechny uživatele. Pro naše potřeby do souboru umístíme parametry připojení databáze a ID zobrazované položky tabulky databáze. Lze sem však ukládat informace i v rámci *Configure* módu portletu (viz. kapitola C.3.2.). Pro přístup k databázi potřebujeme uchovávat URL, jméno uživatele a heslo. Přidáme si proměnné s názvy *dbname*, *user* a *passwd*. Pro úplnost přidáme i proměnnou *toshow* s prázdným obsahem, kterou budeme v projektu také potřebovat.

Otevřeme si soubor *portlet.xml*. V konfiguračním formuláři se přepneme do *Concrete Portlet Application* → *helloworld.HelloWorldPortlet* a do tabulky *Settings Parameters* přidáme tlačítkem *Add* všechny proměnné dle obrázku C.4.



Obrázek C.4: Přidání proměnných a jejich hodnot do souboru *portlet.xml*

Přepneme-li se přes záložku *Source*, můžeme si prohlédnout přímo obsah souboru. Po předchozím zásahu zde přibyly tyto řádky:

```
<config-param>
  <param-name>dbname</param-name>
  <param-value>jdbc:db2j:C:\\LabFiles\\Cloudscape\\WPSADMIN</param-value>
</config-param>
<config-param>
```

```
<param-name>user</param-name>
<param-value>db2admin</param-value>
</config-param>
<config-param>
  <param-name>passwd</param-name>
  <param-value>db2admin</param-value>
</config-param>
<config-param>
  <param-name>toshow</param-name>
  <param-value> </param-value>
</config-param>
```

C.5. Příprava pro práci s databází

WebSphere Site Developer má účinné nástroje pro vytváření databází. Bylo by škoda je nevyužít. Nejprve si musíme vytvořit propojení studia s databází. Použijeme standardní databázi Cloudscape. Použití Oracle databáze je však téměř shodné. Pro testování je třeba stáhnout si ze stránek <http://ibm.com> soubor SG246076.zip, který je též na CD-ROM přiloženém k dokumentaci. Vytvořme si adresář C:\LabFile\, kam nakopírujeme po rozbalení adresář Cloudscape.

C.5.1. Vytvoření spojení

Přepneme se do perspektivy *Data*. V levé dolní části vidíme sekci DB Servers. Pravým tlačítkem vybereme v menu položku New Connection. Formulář vyplníme podle obrázku C.5.

The screenshot shows the 'New Database Connection' dialog box. The fields are filled as follows:

- Connection name: Connection
- Database: WP5SAMPLE
- User ID: db2admin
- Password: *****
- Database vendor type: Cloudscape, V5.0
- JDBC driver: Cloudscape Embedded JDBC Driver
- Host: (empty)
- (Optional) Port number: (empty)
- Server name: (empty)
- Database Location: C:\LabFiles\Cloudscape\WP5SAMPLE
- JDBC driver class: com.ibm.db2.jdbc.DB2JDriver
- Class location: C:\Program Files\IBM\WebSphere Studio\runtimes\portal_v50\shared\app\cloudscape\db2.jar
- Connection URL: jdbc:db2:C:\LabFiles\Cloudscape\WP5SAMPLE

Obrázek C.5: Konfigurace připojení k databázi.

C.5.2. Tvorba databáze

Otevřeme novou položku `Connection` a pravým tlačítkem klikneme na databázi `WSSAMPLE`. V menu zvolíme `Import to Folder` a importujeme obsah do adresáře `HelloWorld/Web Content/WEB-INF/`. Nyní máme v projektu nový adresář `database`. Pravým tlačítkem klikneme na podadresář `Tables` a zvolíme nabídku `New→New table definition`. Novou tabulku nazveme `EXAMPLE`. Vytvoříme položku `ID` typu `VARCHAR 20` s příznakem `key`, `CONTENT` typu `LONG VARCHAR` a potvrdíme tlačítkem `Finish`. Kliknutím pravým tlačítkem na tuto novou tabulku zvolíme možnost `Export to server`, čímž novou tabulku exportujeme ze studia do databáze.

C.5.3. Java Beans pro komunikaci s databází

Nyní máme vše připravené a můžeme začít programovat vlastní komunikaci. Studio nám usnadní práci tím, že předpřipraví třídy pro navázání spojení, zaslání SQL dotazu databázi a zpracování výsledku. Je však potřeba definovat, jaké dotazy to budou.

První dotaz bude typu `INSERT` pro přidání nové položky do tabulky. Pravým tlačítkem klikneme na adresář `database/WSSAMPLE/Statements` a vybereme `New→New Insert Statement`. Nazveme jej `SQLInsert`. Zobrazí se nám formulář pro vytváření dotazu pro přidání nové položky do tabulky. Pravým tlačítkem klikneme do okénka `Tables` a zvolíme `Add Table→ADD.EXAMPLE`. Zaškrtneme ty položky, které budeme přidávat (v našem případě tedy `ID` a `CONTENT`). Jelikož obě položky jsou typu `VARCHAR`, musí být jejich hodnoty uzavřeny do apostrofů. V okně `SQL Source` vidíme konečný SQL dotaz, který vykoná náš požadavek.

Podobný postup zopakujeme pro nový dotaz typu `SELECT` jenž naopak vybere data z tabulky. Do adresáře `Statement` přidáme nový přes menu `New→New Select Statement`. Nazveme jej `SQLSelect`. Opět zvolíme tabulku, se kterou budeme pracovat. Zaškrtneme ty položky, které z tabulky chceme získat (tedy `ID` i `CONTENT`).

Přistoupíme nyní ke generování obslužných tříd. Pravým kliknutím na soubory `SQLInsert` a `SQLSelect` v menu vybereme `Generate Java Bean`. `Source Folder` bude `HelloWorld/Java Source`, `package` zvolíme `heloworld`. Jména tříd zvolme stejná, jako `Statement` soubory.

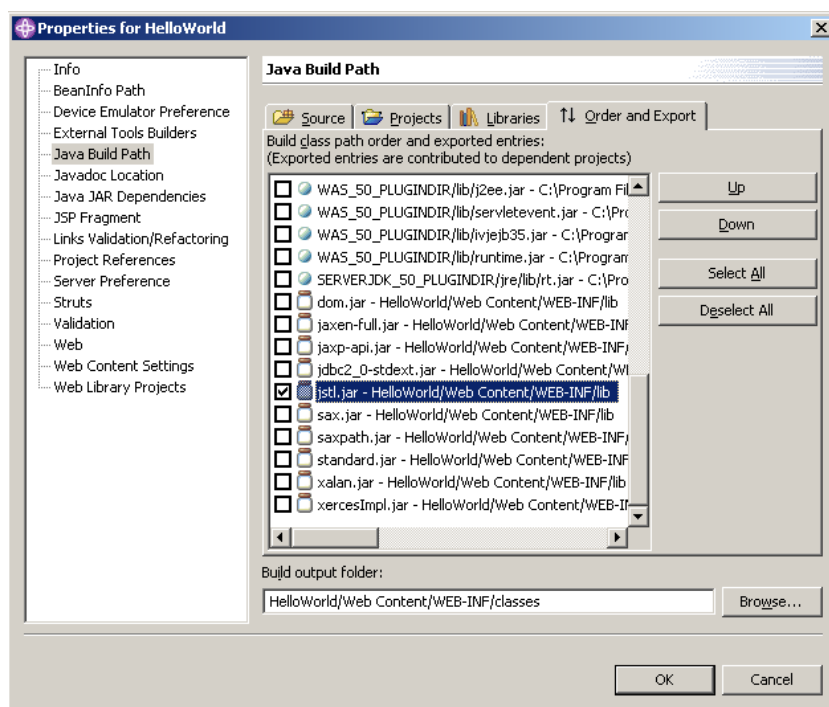
Funkčnosti obou dotazů si můžeme vyzkoušet přes nabídku `Execute` v menu souborů `SQLInsert` a `SQLSelect` v adresáři `Statements`.

C.6. Lokalizace

Ze všeho nejdříve musíme do projektu přidat speciální knihovnu `jstl.jar`, která má na starosti práci s NSL překladovými soubory. Pravým tlačítkem klikneme na projek, zvolíme položku `Properties$ \rightarrow $Web` a označíme možnost `Include the JSP Standard Tag Library`. Výběr potvrdíme tlačítkem `Apply`. Ze sekce `Web` přejdeme do `Java Build Path` (viz obrázek C.6). Ve výčtu knihoven bychom měli nalézt soubor `jstl.jar`, zaškrtneme jej a potvrdíme tlačítkem `OK`.

C.6.1. NLS (National Language Support)

Tento způsob lokalizace použijeme pro většinu textu. Vytvoříme si adresář `HelloWorld/Java Source/nls` a v něm prázdné soubory `NLS_cs.properties`



Obrázek C.6: Zakomponování knihovny jstl.jar do projektu.

a NLS_en.properties. Obsah souborů by se měl plnit dle následujícího vzoru.

NLS_cs

```
submitButton = Ulo\u017Eit
deleteButton = Odebrat
selectButton = Vybrat
```

NLS_en

```
submitButton = Save
deleteButton = Remove
selectButton = Select
```

Používáme-li speciální znaky nebo znaky s interpunkčními znaménky, použijeme tzv. *unicode*. Jedná se o univerzální zápis ve formátu \uXXXX, kde XXXX představuje číslo v šestnáctkové soustavě. Bližší informace o *unicodu* najdete na www.unicode.org.

Do souboru typu JSP importujeme knihovnu jstl a definujeme umístění NLS souborů. Název se udává bez posledních třech znaků představující lokalizaci jazyka.

```
<%@ taglib uri="http://java.sun.com/jstl/fmt" prefix="fmt" %>
<fmt:setBundle basename="nls.NLS"/>
```

Na místo, kam chceme vypsát přeložený text, vložíme příkaz:

```
<fmt:message key="submitButton"/>
```

C.6.2. Překlad celého souboru

Mohlo by se stát, že potřebujeme přeložit celý soubor a nevystačíme pouze s klíčovými slovy či krátkými větami. Pro specifické jazyky mnohdy potřebujeme provést i větší zásahy do programu (např. zarovnání textu na pravou stranu a pod.). I na tyto případy je portál dobře připraven.

Vytvoříme si následující adresáře :

```
HelloWorld/Web Content/helloworld/jsp/html/cs  
HelloWorld/Web Content/helloworld/jsp/html/en
```

Do těchto adresářů umístíme kopie téhož jsp souboru pro různé lokalizace. Soubory ponechané v adresáři o úroveň výše (tedy HelloWorld/Web Content/helloworld/jsp/html) se lokalizují předchozím způsobem.

C.7. Jedinečné názvy proměnných a funkcí

Mohlo by se stát, že na stránce portálu se sejde více portletů zároveň. V nich by mohly kolidovat shodně pojmenované funkce a proměnné JavaScriptu. Portál má účinný nástroj, jak zajistit jedinečnost názvu v rámci stránky.

C.7.1. JavaScript

Nejkritičtějším místem kolize je již zmiňovaný JavaScript. Proto v místě ve zdrojovém kódu JSP, kde potřebujeme použít takovýto kritický kód, deklaruujeme název globální proměnné ve scriptu dle následujícího vzoru (deklarace pole s původním názvem item):

```
<SCRIPT language="JavaScript">  
  <portletAPI:encodeNamespace value="item"/> = new Array();  
</SCRIPT>
```

Prohlédneme-li si vygenerovaný HTML zdrojový kód, který je zaslán uživateli, uvidíme, do jakého tvaru byl název pole zakódován:

```
<SCRIPT language="JavaScript">
  PC_7_0_1BQ_item = new Array();
</SCRIPT>
```

C.8. Konkrétní úpravy projektu

Nyní známe vše potřebné a můžeme přejít k samotné realizaci portletu. Projdeme si všechny soubory, ve kterých je třeba učinit změny. Pokud se budete držet přesně pokynů, měli byste na konci získat plně funkční portlet a měli byste být schopni podobnou aplikaci naprogramovat sami. Nelekejte se občasnou rozsáhlostí výpisů zdrojových kódů a snažte se pochopit hlavní myšlenky.

C.8.1. HelloWorldPortlet.java

3.8.1.1. Import knihoven

```
import java.sql.SQLException; // Pro práci s SQL
import java.util.Locale;     // Pro práci s lokalizací
import helloworld.SQLSelect; // Utility pro zpracování SELECT SQL dotazu
import helloworld.SQLInsert; // Utility pro zpracování INSERT SQL dotazu
```

3.8.1.2. Globální proměnné

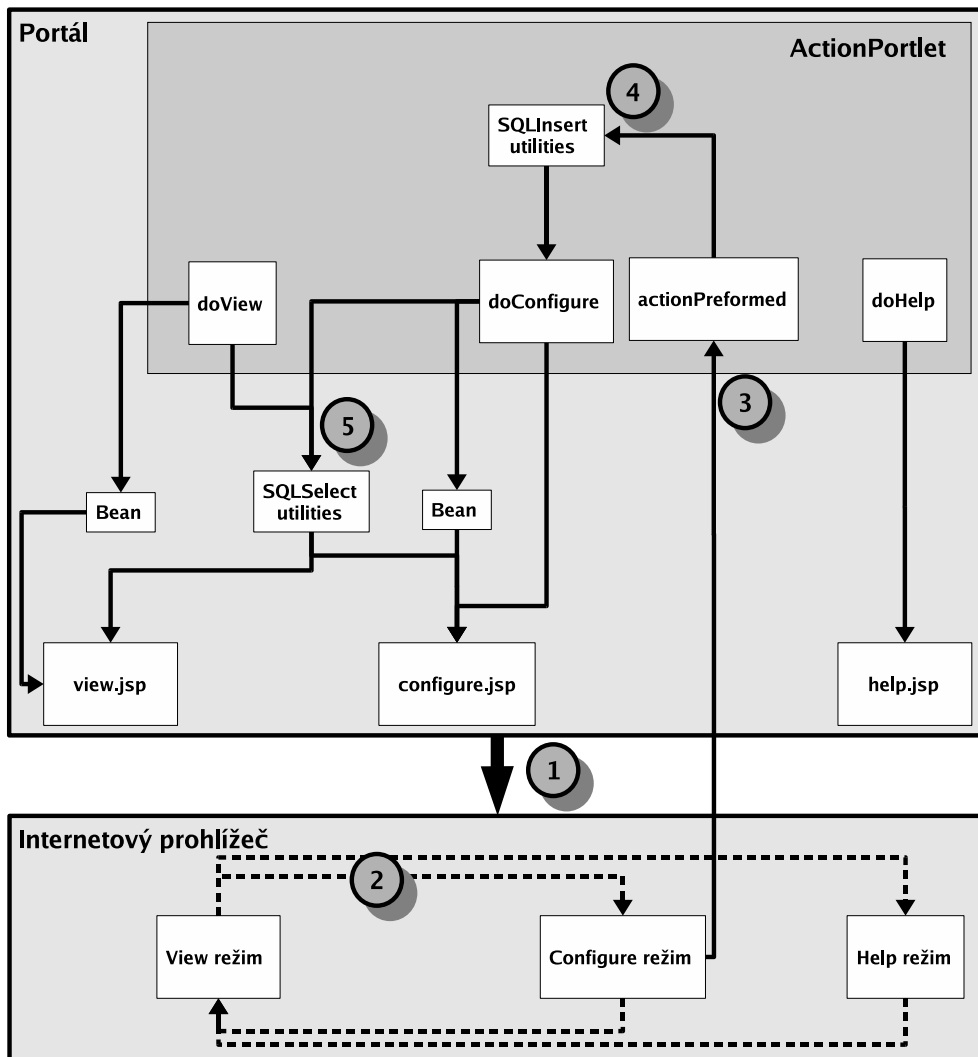
Přidáme nezbytné globální proměnné a konstanty.

```
public static final String CONFIG_ACTION =
    "helloworld.HelloWorldPortletConfigFormAction";
public static final String NLS = "nls.NLS";

public static final String SUBMIT_BUTTON = "submitButton";
public static final String REMOVE_BUTTON = "removeButton";
public static final String SELECT_BUTTON = "selectButton";

//Název, podle kterého se pozná nová položka
public static String newItemName = "new";

public static String dbname = "";
public static String user = "";
public static String passwd = "";
public static String toShow = "";
```



Obrázek C.7: Blokové schéma HelloWorld portletu.

- CONFIG_ACTION slouží jako identifikační jméno, pod kterým bude vystupovat požadavek zasláný formulářem v Config módu. Pro každý mód se musí použít jiný název.
- NLS nese název souboru lokalizace.
- SUBMIT_BUTTON - univerzální název pro tlačítko uložit.
- REMOVE_BUTTON - univerzální název pro tlačítko Odebrat.
- SELECT_BUTTON - univerzální název pro tlačítko Vybrat.
- newItemName - univerzální pojmenování nové položky ve formuláři.
- dbname - URL databáze, načítá se stejně, jako user, passwd a toShow při inicializace portletu.
- user - uživatelské jméno pro přístup k databázi.
- passwd - heslo pro přístup k databázi.
- toShow - ID položky, který se bude ve View módu zobrazovat.

3.8.1.3. Metoda doView()

V první řadě musíme inicializovat konfigurační parametry obsažené v souboru portlet.xml (dbname, user, passwd a toshow). Parametry se načítají při každém volání metody doView(). V praxi to však není standardní postup. Je lépe parametry načíst pouze jednou a udržovat v *session* proměnných, jako proměnnou toShow. Z databáze musíme získat informace, které bude portlet ve *View* módu zobrazovat. Vytvoříme objekt typu SQLSelect a zavoláme metodu execute() s parametry databáze a konkrétním SQL dotazem. Metodou populateData() výsledek zjednodušeně řečeno zpřístupníme pro *View* mód.

```
// Nacteni konfigurace a ulozeni do globalnich promennych
PortletSettings settings = request.getPortletSettings();

dbname = (String) settings.getAttribute("dbname");
user = (String) settings.getAttribute("user");
passwd = (String) settings.getAttribute("passwd");
toShow = (String) settings.getAttribute("toshow");

// Pokud neni vybrana zadna polozka, veme se polozka nazctena z portlet.xml
if (sessionBean.getToShow().length()==0)
    sessionBean.setToShow(toShow);

SQLSelect sqlUtility = new SQLSelect();
try {
    sqlUtility.execute(user, passwd, dbname, "SELECT CONTENT FROM EXAMPLE WHERE ID =
    '" + sessionBean.getToShow() + "'");
}
catch (SQLException e) {
    e.printStackTrace();
}
sqlUtility.populateDataView(viewBean);
```

3.8.1.4. Metoda doConfigure()

V metodě doConfigure() provedeme velmi podobné úpravy. Od metody doView() se významně neliší. Potřebujeme opět získat určité informace z databáze, abychom

je v *Configure* módu mohli zobrazovat a editovat. Jelikož budeme muset při zpracování požadavku z tohoto formuláře rozlišovat mód, ze kterého požadavek přišel, musíme *Configure* mód označit tzv. URI, do kterého uložíme konstantu `CONFIG_ACTION`.

```
SQLSelect sqlUtility = new SQLSelect();
// Vykonani SQL dotazu
try {
    sqlUtility.execute(user, passwd, dbname, "SELECT * FROM EXAMPLE");
}
catch (SQLException e) {
    e.printStackTrace();
}
sqlUtility.populateDataConfig(configBean);

// Nastaveni actionURI
PortletURI formActionURI = response.createURI();
formActionURI.addAction(CONFIG_ACTION);
configBean.setFormActionURI(formActionURI.toString());
```

3.8.1.5. Metoda `actionPerformed()`

Metoda zpracovává všechny HTTP požadavky zaslané od uživatele. Rozlišuje mód, ze kterého požadavek pochází podle řetězce získaného metodou `event.getActionString()`. Do metody se standardně dává podmínka pro rozpoznání módu, ze kterého požadavek pochází. Studio vygenerovalo pouze podmínku pro rozpoznání a obsluhu požadavku pocházejícího z módu *View*. Musíme tedy doplnit druhou podmínku pro rozpoznání *Configure* požadavku. Pro práci s konkrétními názvy tlačítek formulářů musíme zjistit preferovanou lokalizaci uživatele a přeložit potřebné názvy.

```
//Action in Configure mode
if (CONFIG_ACTION.equals(actionString)) {

    // Zjistí lokalizaci
    Locale locale = event.getRequest().getLocale();
    SQLInsert sqlUtility = new SQLInsert();
    HelloWorldPortletConfigBean configBean = new HelloWorldPortletConfigBean();

    // Preklad nazvu tlacitek
    String submitButton = getPortletConfig().getContext().getText(NLS,
        SUBMIT_BUTTON, locale);
    String removeButton = getPortletConfig().getContext().getText(NLS,
        REMOVE_BUTTON, locale);
    String selectButton = getPortletConfig().getContext().getText(NLS,
        SELECT_BUTTON, locale);

    String SQLstring = "";

    // Nacteni obsahu formulare
    String button = (String) request.getParameter("button");
    String oldItem = (String) request.getParameter("item").replace(' ', '_');
    String newItem = (String) request.getParameter("new_item").replace(' ', '_');
```

```

String content = (String) request.getParameter("content");
String toShow = (String) request.getParameter("toShow");

// Stisknuto tlačitko uložit
if (button.equals(submitButton) && !newItem.equals("")) {
    if (oldItem.equals(newItemName)) {
        SQLstring = "INSERT INTO EXAMPLE VALUES" + " ( '" + newItem +
            "' ,'" + content +"' )";
    }
    else {
        SQLstring = "UPDATE EXAMPLE SET ID = '" + newItem + "' , CONTENT =
            '"+content+"' WHERE ID ='" + oldItem + "'";
    }
}

// Stisknuto tlačitko odebrat
else if (button.equals(removeButton) && !newItem.equals("")) {
    SQLstring = "DELETE FROM EXAMPLE WHERE ID = '" + oldItem + "'";
}

if (!SQLstring.equals("")) {
    try {
        sqlUtility.execute(user, passwd, dbname, SQLstring);
    } catch (SQLException e) {
        e.printStackTrace();
    }
}

// Uložení vybrané položky do portlet.xml
sessionBean.setToShow(toShow);
request.getPortletSettings().setAttribute("toshow", toShow);
try {
    request.getPortletSettings().store();
}
catch (IOException e) {
}

// po akci zustane portlet v aktualnim modu
try {
    request.setModeModifier(Portlet.ModeModifier.CURRENT);
}
catch (AccessDeniedException ade) {
}
}

```

Do proměnné `locale` načteme lokalizaci uživatele. Přeložíme názvy všech tlačítek, abychom mohli rozpoznávat tlačítko, které požadavek vyvolalo. Načteme veškerá data, která přišla. Jména v uvozovkách značí název objektu ve formuláři. Pokud požadavek přišel na podnět tlačítka `submitButton` a jedná se o novou položku, přidá se do databáze přes SQL dotaz `INSERT`. Nejedná-li se o novou, modifikuje existující položku tabulky SQL dotazem `UPDATE`.

Pochází-li podnět od tlačítkem `removeButton`, položka s odpovídajícím ID se z databáze odstraní.

Stisk tlačítka `selectButton` způsobí zápis hodnoty z proměnné `toShow` do `session` proměnné a do souboru `portlet.xml`. Poslední část nastaví následující mód portletu. V našem případě je vhodné ponechat mód nezměněn. V některých aplikacích je naopak výhodnější jako reakci na stisk tlačítka přejít do `View` módu.

C.8.2. SQLInsert.java

Je třeba upravit stávající metodu `execute()`, jelikož potřebujeme přenášet více parametrů. Zásah není příliš radikální.

```
import helloworld.HelloWorldPortlet;

public void execute(
    String userid,
    String password,
    String url,
    String command)
    throws SQLException {
    try {
        modify.setUrl(url);
        modify.setCommand(command);
        modify.setUsername(userid);
        modify.setPassword(password);
        modify.execute();
    }

    // Free resources of select object.
    finally {
        modify.close();
    }
}
```

C.8.3. SQLSelect.java

Tutěž metodu změníme i ve třídě `SQLSelect`. Třidu rozšíříme o dvě metody. Každá z nich má na starosti uchování jednoho výsledku po `SELECT` dotazu. Metoda `populateDataView()` ukládá data pro potřeby `View` módu, druhá pak pro potřeby `Configure` módu. Pro úplnost musíme importovat hlavní třídu projektu `HelloWorldPortlet`.

```
import helloworld.HelloWorldPortlet;

public void execute(
    String userid,
    String password,
    String url,
    String command)
    throws SQLException {
    try {
        select.setUrl(url);
        select.setCommand(command);
        select.setUsername(userid);
        select.setPassword(password);
        select.execute();
    }
    // Free resources of select object.
    finally {
        select.close();
    }
}
```

```
    }  
  }  
  public void populateDataView>HelloWorldPortletViewBean viewBean) {  
    viewBean.setResultFromDatabase(select);  
  }  
  public void populateDataConfig>HelloWorldPortletConfigBean configBean) {  
    configBean.setResultFromDatabase(select);  
  }  
}
```

C.8.4. HelloWorldPortletViewBean.java

Do *Bean* souboru *View* módu doplníme globální proměnnou pro práci s výsledkem SQL dotazu, metody *set* a *get* pro její čtení/zápis a importujeme knihovnu `com.ibm.db.beans.DBSelect`.

```
import com.ibm.db.beans.DBSelect;  
  
private DBSelect resultFromDatabase;  
  
public DBSelect getResultFromDatabase() {  
    return resultFromDatabase;  
}  
public void setResultFromDatabase(DBSelect resultFromDatabase) {  
    this.resultFromDatabase = resultFromDatabase;  
}
```

Poznámka: Metody pro obsluhu globální proměnné je možné nechat generovat vývojovým prostředím automaticky. Při práci v rozsáhlejší projektů toto zjednodušení opravdu oceníte. Stačí pravým tlačítkem kliknout kamkoliv do editačního okna a vybrat položku `Source$rightarrow$Generate Getter and Setter`. Předem však musí být globální proměnná deklarována.

C.8.5. HelloWorldPortletConfigBean.java

Do třídy importujeme knihovnu `com.ibm.db.beans.DBSelect`. Přidáme globální proměnnou pro práci s výsledkem SQL dotazu `resultFromDatabase` a proměnnou `formActionURI` pro přenášení ID akce formuláře. Opět vygenerujeme, nebo napíšeme, obslužné metody *get* a *set*.

```
import com.ibm.db.beans.DBSelect;  
  
private DBSelect resultFromDatabase;  
private String formActionURI = "";  
  
public DBSelect getResultFromDatabase() {  
    return resultFromDatabase;  
}
```

```
}

public void setResultFromDatabase(DBSelect resultFromDatabase) {
    this.resultFromDatabase = resultFromDatabase;
}

public String getFormActionURI() {
    return formActionURI;
}

public void setFormActionURI(String formActionURI) {
    this.formActionURI = formActionURI;
}
```

C.8.6. HelloWorldPortletSessionBean.java

Tuto třídu rozšíříme o globální proměnnou `toShow` typu `String` a obslužné metodu `getToShow()` a `setToShow()`. Do proměnné se bude ukládat ID položky, která se ve *View* módu zobrazí.

```
private String toShow = "";

public String getToShow() {
    return toShow;
}

public void (String toShow) {
    this.toShow = toShow;
}
```

C.8.7. HelloWorldPortletView.jsp

JSP soubory prakticky všechny přepíšeme. Zdrojové kódy nebudou příliš dlouhé. `HelloWorldPortletView.jsp` je dynamicky generovaný HTML kód *View* módu portletu. První tři řádky importují důležité knihovny pro práci s Java soubory projektu, standardními metodami Portlet API a lokalizací. Následuje vytvoření objektu `viewBean`. Povede-li se vytvořit objekt, do proměnné `results` načteme výsledek SQL dotazu volaného v metodě `doView()` třídy `HelloWorldPortlet`. Zjednodušeně lze říci, že je `results` dvojrozměrné pole indexované od souřadnic `[1,1]`. Ve dvou vzájemně vnořených cyklech pole procházíme a vypisujeme obsah buněk. Prakticky ale `results` obsahuje pouze jednu položku.

```
<%@ page contentType="text/html" import="java.util.*, helloworld.*"%>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<%@ taglib uri="http://java.sun.com/jstl/fmt" prefix="fmt" %>
<portletAPI:init/>
```

```
<%
HelloWorldPortletViewBean viewBean = (HelloWorldPortletViewBean)
    portletRequest.getAttribute(HelloWorldPortlet.VIEW_BEAN);
%>
<HTML>
<BODY>
<DIV style="margin: 6px">
<%
if (viewBean == null) {
%>
Chyba v komunikaci s DB.
<% } else {

com.ibm.db.beans.DBSelect results = viewBean.getResultFromDatabase();

    for(int row=1; row<=results.getRowCount(); row++) {
        for(int col=1; col<=results.getColumnCount(); col++) {
            %>
<%=results.getCacheValueAt(row, col)%>

<%
        }
    }
} %>
</DIV>
</BODY>
</HTML>
```

C.8.8. HelloWorldPortletConfig.jsp

Configure mód má nejrozsáhlejší zdrojový soubor ze všech JSP. Hlavička je prakticky stejná jako v předešlém případě. Přibyl zde JavaScript, jenž umožní dynamicky pracovat s daty na straně uživatele. Z hlediska programování portletů však není podstatný a proto ho nebudeme detailněji popisovat. Podstatná je ukázka způsobu, jak ošetřit názvy proměnných a funkcí kvůli nežádoucí duplicitě. Názorná je též ukázka vnitřní struktury formuláře, zejména úvodní tag <FORM>, kde se jako action parametr vkládá URI, pod kterým je požadavek formuláře rozpoznáván (viz. kapitola 3.8.1.5.). Pozornost věnujte generování názvů tlačítek. Využívání funkcí pro lokalizaci textů znepráhledňuje zdrojový kód. Proto je lépe zabývat se touto problematikou až na úplný závěr programování.

```
<%@ page contentType="text/html" import="java.util.*, helloworld.*"%>
<%@ taglib uri="/WEB-INF/tld/portlet.tld" prefix="portletAPI" %>
<%@ taglib uri="http://java.sun.com/jstl/fmt" prefix="fmt" %>
<portletAPI:init/>

<fmt:setBundle basename="nls.NLS"/>
<%
HelloWorldPortletConfigBean configBean =
    (HelloWorldPortletConfigBean)portletRequest.
        getAttribute(HelloWorldPortlet.CONFIG_BEAN);
HelloWorldPortletSessionBean sessionBean =
    (HelloWorldPortletSessionBean)portletRequest.getPortletSession().
        getAttribute(HelloWorldPortlet.SESSION_BEAN);

com.ibm.db.beans.DBSelect results = configBean.getResultFromDatabase();
```

Příloha C. Tutorial: Vývoj portletů pro WebSphere Portal

```
String item = "";
%>

<SCRIPT language="JavaScript">
  <portletAPI:encodeNamespace value="item"/> = new Array();
  <portletAPI:encodeNamespace value="content"/> = new Array();

  <%
  for(int row = 1; row <= results.getRowCount(); row++) {
  %>
  <portletAPI:encodeNamespace value="item"/>[<%out.print(row);%>] =
    "<%=results.getCacheValueAt(row, 1)%>";
  <portletAPI:encodeNamespace value="content"/>[<%out.print(row);%>] =
    "<%=results.getCacheValueAt(row, 2)%>";
  <%}%>

  function <portletAPI:encodeNamespace value="output" />(formular) {

    if (formular.item.value != "new") {
      formular.new_item.value = formular.item.value;
      for (var i = 1; i <= <%=results.getRowCount()%>; i++) {
        if (<portletAPI:encodeNamespace value="item" />[i] ==
          formular.item.value) {
          formular.new_item.value = formular.item.value;
          formular.content.value =
            <portletAPI:encodeNamespace value="content"/>
            [i];
        }
      }
    }
    else {
      formular.new_item.value = "";
      formular.content.value = "";
    }
  }
</SCRIPT>
<HTML>
<BODY>
<H3 style="margin-bottom: 3px"><fmt:message key="configTitle"/></H3>
<DIV style="margin: 6px">
<FORM method="post"
  action="<portletAPI:createReturnURI><portletAPI:URIAction
  name='<%=configBean.getFormActionURI()%>'/>
  </portletAPI:createReturnURI>">
<TABLE border="0">
  <TBODY>
  <TR>
  <TD valign="top" valign="top">
    <FIELDSET><LEGEND><fmt:message key="editTitle"/></LEGEND>
    <DIV style="margin: 6px">
      <SELECT size="8" name="item"
        onChange=
" <portletAPI:encodeNamespace value="output" />(this.form)">
        <OPTION value="new" selected><fmt:message key="newItem" />
</OPTION>
        <OPTION value="new">-----</OPTION>
        <%
        for(int row = 1; row <= results.getRowCount(); row++) {
        %>
        <OPTION value="<%
          item = "" + results.getCacheValueAt(row, 1);
          out.print(item.replace('_', ' '));
        %>">
          <% out.print(row + ". ");%> <%
          out.print(item.replace('_', ' '));
        %></OPTION>
```

```

    <%}%>
    </SELECT><BR>
    <fmt:message key="editItem" /> :<BR>
    <INPUT type="text" name="new_item" size="20"
maxlength="20" value="">
    <BR>
    <TEXTAREA rows="8" cols="20" name="content"></TEXTAREA>
    </FIELDSET>
    <INPUT class="wpsButtonText"
    value="<fmt:message key="submitButton"/>" name="button"
    type="submit">
    <INPUT class="wpsButtonText"
    value="<fmt:message key="removeButton"/>" name="button"
    type="submit">
    </DIV>
    </TD>
    <TD valign="top">
    <FIELDSET><LEGEND><fmt:message key="selectTitle"/></LEGEND>
    <DIV style="margin: 6px">
    <%
    for(int row = 1; row <= results.getRowCount(); row++) {
    %>
    <INPUT type="radio" name="toShow"
    value="<%
    item = "" + results.getCacheValueAt(row, 1);
    out.print(item.replace('_',' '));
    %>"
    <%
    if (item.equals(sessionBean.getToShow()))
    out.print("checked");
    %>>
    <%out.print(row + ". " + item.replace('_',' ') +
" <BR>"); %> <%}%>
    </FIELDSET>
    <INPUT class="wpsButtonText"
    value="<fmt:message key="selectButton"/>" name="button"
    type="submit">
    </DIV>
    </TD>
    </TR>
    </TBODY>
</TABLE></DIV>
</BODY>
</HTML>

```

C.9. Vytvoření WAR souboru a instalace

V konečné fázi vyexportujeme projekt do souboru HelloWorld.war přes nabídku Files→Export WAR file. Standardně se exportují i zdrojové soubory. Tuto možnost lze případně zrušit odznačením možnosti Export source files.

Instalace na portál je velmi snadná. Soubor lze jednoduše *uploadovat* přímo přes stránky portálu. Musíte mít však práva privilegovaného uživatele.

Náhledy konečné podoby portletu si můžete prohlédnout na obrázcích C.8 a C.9.

C.10. Možné problémy

Práce s databází je mírně komplikovaná a proto dbejte několika zásad. Při testování portletu musí být přerušeno spojení Developer Studia s databází. Portál má jinak problém s navázáním vlastního spojení. Není třeba nastavené připojení v okně DB Servers definitivně odstranit. Pouze přes menu, vyvolaném pravým tlačítkem myši nad tímto spojením, vyberte možnost `Disconnect`. Pokud plánujete využívat databáze jiného typu, předem si zkontrolujte, zda portál obsahuje ovladače k této databázi. Například Oracle databáze vyžaduje ovladač `classes12.zip`¹. Z tohoto důvodu byl celý ukázkový příklad vysvětlován nad databází Cloudscape, která je standardně portálem podporována.

C.11. Rozdíly databází Cloudscape a Oracle

Rozdílů by sa jistě našlo mnoho. Zásadním problémem v případné migraci mezi databázemi jsou různé datové typy. U Cloudscape se pro ukládání časových značek používá typ `TIMESTAMP` ve formátu `yyyy-MM-dd HH:mm:ss`. Databáze sama nemá prostředky pro transformaci na jiný formát pro výstup (např. český tvar data). U Oracle pro ukládání času a data slouží typ `DATE`. Jeho formát je však složitější. Uložit lze buď datum, čas, nebo obojí.

Ukázky SQL dotazů:

zápis časové značky:

```
INSERT INTO TABULKA VALUES ({ts '2004-10-10 12:11:00'})
```

zápis času:

```
INSERT INTO TABULKA VALUES ({t '2004-10-10 12:11:00'})
```

zápis data:

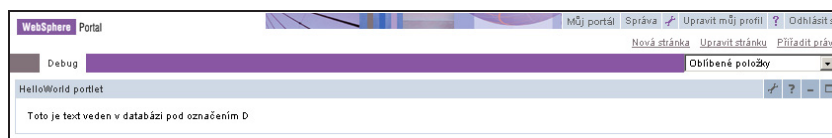
```
INSERT INTO TABULKA VALUES ({d '2004-10-10 12:11:00'})
```

Načítání v českém formátu :

```
SELECT TO_CHAR(CASOVA_ZNACKA, 'DD. MM. YYYY HH24:MI:SS') FROM TABULKA
```

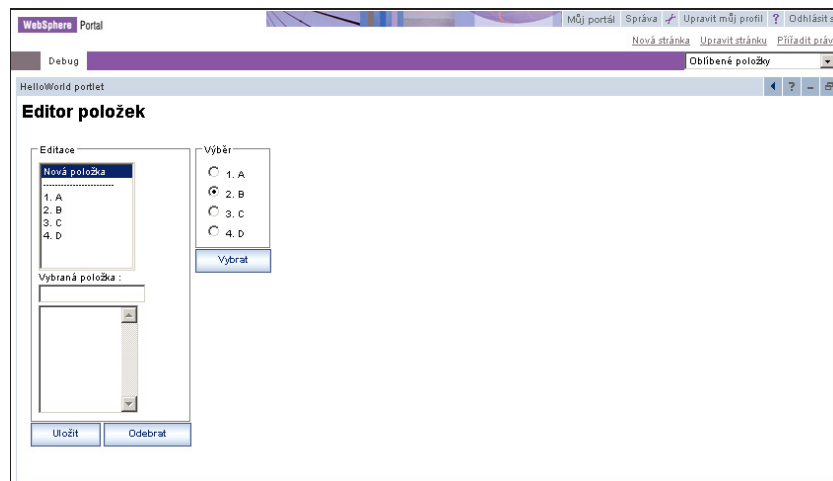
Poznánka:

Pokud nepoužijeme při načítání funkci `TO_CHAR`, vrátí databáze pouze datum v anglickém formátu.



Obrázek C.8: View režim HelloWorld portletu

¹Soubor je obsažen na příloženém CD-ROM



Obrázek C.9: Configure režim HelloWorld portletu

Skripta pro distanční vzdělávání

D.1. Účel

Skripta slouží k zaškolení pracovníků a studentů OSU pro obsluhu portálu. Jako vhodná forma bylo zvoleno distanční vzdělávání (eLeraning). V roce 2004 vyšlo první ověřovací vydání v nákladu 40 kusů, na první polovnu roku 2005 je plánováno druhé rozšířené vydání o nákladu cca 150 až 200 kusů.

D.2. Obsah

1. Portál jako univerzální pomocník
2. Základy práce s portálem
3. Portál z pohledu zaměstnance
4. Portál z pohledu studenta
5. Portlety pro úzké skupiny uživatelů
6. Budoucí vize a vývoj

D.3. Bibliografické informace

Název:	Portál Ostravské university
Autoři:	Mgr. Jiří Šimonek
Vydání:	první, 2004
Počet stran:	33
Náklad:	40 ks
Tisk:	Ediční středisko CIT OU
Vydavatel a tisk:	Ostravská univerzita v Ostravě

Příloha **E**

Zprávy z konferencí

E.1. IBM WebSphere Technical Conference

Innsbruck AT, listopad 2003

Jenalo se o první účast na konferenci IBM. Účastníkem byl J. Valdman a protože se konference konala ještě před schválením projektu, byla hrazena ze zdrojů ZČU.

Na této konferenci jsme získali množství podnětů a materiálů týkajících se WebSphere Application Server a WebSphere Portal, zejména v nedávno uvedené verzi 5. Seznámili jsme se s novým systémem administrace WebSphere Application Server, vývojem aplikací pomocí WSAD, podporovanými standardy J2EE. Díky této konferenci jsme získali přehled a zejména odkazy na další zdroje informací v široké problematice WebSphere Portal.

V diskusi s experty IBM (Keys Botsum, David Eyrman, Jim Plaistrant, Ron Tell) byly diskutovány aspekty napojení WebSphere Portal na prostředí ZČU, zejména Kerberos5 a OpenLDAP. Místo odpovědí jsme buď dostali jen další otázky. User registry WebSphere Application Server a WebSphere Portal se ukázalo příliš složité a nedokumentované, aby šlo hladce udělat vlastní custom implementaci. Původní záměr vytvořit TAI moduly nám byl rovněž rozmluven.

E.2. IBM WebSphere Portal and Lotus Technical Conference

Stuttgart DE, březen 2004

Účastníky této konference, jejímž hlavním tématem byl kromě WebSphere Portal také Lotus Workplace, byli Jan Valdman a Jiří Šimonek. V té době již ZČU i OSU měly instalovaný WebSphere Portal 5.0, čímž se nám rozšířil okruh problémů na otázky správy WebSphere Application Server, vývoje portletů (IBM API vs JSR168), využívání technologií WebSphere Portal a jeho správná konfigurace a jiné. Stranou nezůstal ani Lotus Workplace, resp. ILWWCM, který je jedním z portálových nástupců Lotus Collaborative Components.

Mimo jiné se (opět) ukázalo, že konference IBM jsou určeny hlavně pro IBM partnery a nikoli zákazníky. Přestože se zde (a pouze zde!) člověk dostane k nepostradatelným informacím nad rámec dostupné dokumentace, s odpověďmi na konkrétní problémy je to horší. Nicméně většina přednášek je velmi přínosná, protože takto strávených 30 minut vydá za studium 1000stránkového redbooku.

Společnou několikadenní cestu jsme také využili k vzájemné intenzivní výměně informací mezi ZČU a OSU.

E.3. Pan European Portals Conference – PEPC2004

Nottingham UK, červenec 2004

Konference se zúčastnil Jan Valdman. Jedná se o jedinou známou akademickou portálovou konferenci. Byly zde prezentovány projekty jednotlivých škol; komerční prezentace prakticky chyběly.

Poučné bylo zjištění, že anglické školy nejsou o mnoho dále než české, a že i jinde na portálu pracují 2–3 lidé, kteří portál dostanou do pilotního provozu typicky po 2 letech.

Anglické školy obvykle volí opensourcový produkt uPortal (který neumí zatím česky) podporující JSR168. uPortal nezle po stránce infrastruktury srovnávat s WebSphere

Portal, je zaměřen více jako redakční systém než jako integrátor aplikací – jeho implementace je ale také mnohem jednodušší.

Velmi ceně byly prezentované zkušenosti škol (chybami se člověk nejrychleji učí) a také ukázky struktury univerzitních portálů.

E.4. Directory and Single Sign-On in European Universities

Strasbourg FR, listopad 2004

Této konferenci se zúčastnili Jiří Šimonek, Maxmilián Otta, Robert Bohoněk a Michal Švamberg. Jejím cílem byla výměna zkušeností s budováním adresářových služeb a registrů uživatelů za účelem implementace *Single Sign-On (SSO)*. Byly zde prezentovány i výsledky průzkumu provedených *CSIESR* na více než patnácti evropských univerzitách, jakým způsobem jsou na nich tyto technologie nasazeny a používány. Snahou bylo najít především odpovědi na otázky zda používat více adresářových služeb či jediné meta-slужby, jaké atributy a jak detailní informace mají obsahovat, z jakých zdrojů mají být plněny a k jakým účelům jsou zpravidla používány. Zajímavé či možná spíše zákonité je, že univerzity přes tyto technologie dospívají k portálovým řešením svých webových aplikací. Diskutována byla i možnost integrace adresářových služeb mezi univerzitami, která by usnadnila administrativu spojenou s mobilitou studentů a zaměstnanců evropských univerzit.

E.5. Univerzitní informační systémy – konference UNINFOS 04

Bratislava SK, listopad 2004

Konference UNINFOS je tridenní univerzitní konferencí pořádanou slovenskou odnoží sdružení EUNIS, na které vysoké školy prezentují své úspěchy a plány v oblasti IT a IS.

Na konferenci přenesl Jan Valdman příspěvek na téma "Úloha portálu v projektu eUNIVERZITA", který popisoval postup zavádění portálu na ZČU. Organizátoři konference bohužel nezvládli zařazení příspěvků do sekcí, takže příspěvek vyslechlo "nesprávné" publikum.

Z hlediska řešení grantu tato konference nepřinesl aprakticky nic nového (kromě několika podnětů pro rozvoj IS na ZČU) a i z hlediska prezentace dosažených výsledků nesplnila očekávání.

Příloha **F**

Různé

F.1. Hlášení chyby do laboratoří IBM

We have successfully installed the IBM WebSphere Portal 5.0 and migrated it to use the Oracle 8i database. During the migration, we encountered the following problem: the migration tool used a JDBC feature (particularly the `java.sql.Connection.setSavepoint()` method) which is not available in the JDK version (1.3) shipped with the WebSphere Application server. So we were not able to complete the migration. We solved this by compiling the `java.sql` package taken from Sun's JDK 1.4 with the shipped IBM JDK 1.3 and then we replaced the IBM-JDK 1.3 `java.sql` package with it.

The migration went O.K., but we were not able to log into the portal. After going through the WebSphere logs, we realized, that an ORA-01858 error was the cause. We then replaced the Oracle JDBC classes with the Oracle classes with enabled debugging and tracing (`classes12_g.zip`) and found, that there is the following SQL statement sent to the Oracle database causing the ORA-01858 :

```
-----
SELECT - 100 AS VALUEINDEX,
       0 AS DATATYPE,
       WMMDBMBR.WMMDBMBR_ID AS WMMDBMBR_ID,
       MEMBER_DN,
       MEMBER_UNIQUE_ID,
       MEMBER_TYPE,
       0 AS WMMDBATR_ID,
       '' AS NAME,
       '' AS STRINGVALUE,
       0 AS INTEGERVALUE,
       0 AS DOUBLEVALUE,
       0 AS BIGINTVALUE,
       TO_DATE ( '04-FEB-68' ) AS TIMESTAMPVALUE
FROM WMMDBMBR
WHERE 1 = 1
      AND WMMDBMBR.WMMDBMBR_ID IN ( SELECT WMMDBMBR.WMMDBMBR_ID AS WMMDBMBR_ID
                                     FROM WMMDBMBR,
                                     WMMDBASVAL S1
                                     WHERE 1 = 1
                                     AND S1.WMMDBMBR_ID = WMMDBMBR.WMMDBMBR_ID
                                     AND S1.WMMDBATR_ID = :1
                                     AND ( S1.STRINGVALUE LIKE :2 ESCAPE '\ ' )
                                     AND WMMDBMBR.MEMBER_TYPE = 'P' )

UNION SELECT 9 AS VALUEINDEX,
           0 AS DATATYPE,
           WMMDBMBR.WMMDBMBR_ID AS WMMDBMBR_ID,
           MEMBER_DN,
           MEMBER_UNIQUE_ID,
           MEMBER_TYPE,
           WMMDBATR.WMMDBATR_ID AS WMMDBATR_ID,
           NAME,
           STRINGVALUE,
           0 AS INTEGERVALUE,
           0 AS DOUBLEVALUE,
           0 AS BIGINTVALUE,
           TO_DATE ( '04-FEB-68' ) AS TIMESTAMPVALUE
FROM WMMDBATR,
     WMMDBASVAL,
```

```
WMMDBMBR
WHERE WMMDBMBR.WMMDBMBR_ID = WMMDBASVAL.WMMDBMBR_ID
AND WMMDBASVAL.WMMDBATR_ID = WMMDBATR.WMMDBATR_ID
AND HAS_CONTEXT = 0
AND WMMDBATR.WMMDBATR_ID IN ( 10003,
                               10000,
                               10008,
                               10001,
                               10011 )
AND WMMDBMBR.WMMDBMBR_ID IN ( SELECT WMMDBMBR.WMMDBMBR_ID AS WMMDBMBR_ID
                               FROM WMMDBMBR,
                               WMMDBASVAL S1
                               WHERE 1 = 1
                               AND S1.WMMDBMBR_ID = WMMDBMBR.WMMDBMBR_ID
                               AND S1.WMMDBATR_ID = :3
                               AND ( S1.STRINGVALUE LIKE :4 ESCAPE '\ ' )
                               AND WMMDBMBR.MEMBER_TYPE = 'P' )

ORDER BY WMMDBMBR_ID,
         WMMDBATR_ID,
         VALUEINDEX;
```

The expression TO_DATE ('04-FEB-68') uses a default NLS_DATE_FORMAT, that differs from the format expected on the Oracle server side.

After searching for a fix-pack, we found the iFix APAR PQ77683 which fixes problem number 57287 - after database transfer to Oracle 9i, cannot log in to WebSphere Portal. This is very close related to our problem.

We applied the patch and the TO_DATE in the above shown SQL statement change to:

```
TO_DATE ( '04-FEB-68', 'DD-MON-YY' )
```

This is a better solution, but it's still not sufficient. The WebSphere, after it connects to the Oracle database, executes the following set of commands:

```
DRVR OPER Enabled logging (moduleMask 0x0fffffff, categoryMask
0x0000008f)
DRVR OPER OracleConnection.setAutoCommit(autoCommit=true)
DRVR OPER OracleConnection.setAutoCommit(autoCommit): return
DRVR OPER OracleConnection.getDefaultFixedString() returning false
DRVR SQLS SQL: "ALTER SESSION SET NLS_LANGUAGE = 'CZECH'"
DRVR SQLS Input SQL: "ALTER SESSION SET NLS_LANGUAGE = 'CZECH'"
DRVR OPER OracleConnection.getAutoCommit() returned true
DRVR OPER OracleStatement.cleanupForBatching()
DRVR SQLS SQL: "ALTER SESSION SET NLS_TERRITORY = 'CZECH REPUBLIC'"
DRVR SQLS Input SQL: "ALTER SESSION SET NLS_TERRITORY = 'CZECH
REPUBLIC'"
DRVR OPER OracleConnection.getAutoCommit() returned true
DRVR OPER OracleStatement.cleanupForBatching()
DRVR OPER OracleStatement.executeQuery(sql)
DRVR SQLS SQL: "SELECT VALUE FROM NLS_INSTANCE_PARAMETERS WHERE
```

```
PARAMETER = 'NLS_DATE_FORMAT' "  
DRVR SQLS Input SQL: "SELECT VALUE FROM NLS_INSTANCE_PARAMETERS  
WHERE PARAMETER = 'NLS_DATE_FORMAT' "
```

In Oracle, the date format depends also on the current locale. That means, that - for example - '04-FEB-68' becomes '04-ÚNO-68'. We solved finally the problem by changing the parameter 'locale.default.language=cs' to 'locale.default.language=en' in the file '/opt/WebSphere/PortalServer/shared/app/config/services/LocalizerService.properties'.

The WebSphere now uses the american locale:

```
DRVR OPER Enabled logging (moduleMask 0x0fffffff, categoryMask  
0x0000008f)  
DRVR OPER OracleConnection.setAutoCommit(autoCommit=true)  
DRVR OPER OracleConnection.setAutoCommit(autoCommit): return  
DRVR OPER OracleConnection.getDefaultFixedString() returning false  
DRVR SQLS SQL: "ALTER SESSION SET NLS_LANGUAGE = 'AMERICAN' "  
DRVR SQLS Input SQL: "ALTER SESSION SET NLS_LANGUAGE = 'AMERICAN' "  
DRVR OPER OracleConnection.setAutoCommit() returned true  
DRVR OPER OracleStatement.cleanupForBatching()  
DRVR SQLS SQL: "ALTER SESSION SET NLS_TERRITORY = 'AMERICA' "  
DRVR SQLS Input SQL: "ALTER SESSION SET NLS_TERRITORY = 'AMERICA' "  
DRVR OPER OracleConnection.setAutoCommit() returned true  
DRVR OPER OracleStatement.cleanupForBatching()  
DRVR OPER OracleStatement.executeQuery(sql)
```

... for this is the expression TO_DATE ('04-FEB-68', 'DD-MON-YY')
O.K. and we can
now log into the WebSphere Portal.

To definitively fix this problem, it is necessary to change it
either to:

```
TO_DATE ( '04-FEB-68', 'DD-MON-YY', 'NLS_DATE_LANGUAGE = American' )
```

or use a numerical representation of the month (i.e. 'DD-MM-YY'),
or - if the date doesn't matter - to use the SYSDATE variable.

F.2. Nabídka na služby Groupsoft

Přepis nabídky firmy Groupsoft na služby poptávané ZČU. Ceny byly odstraněny.

Nabídka konzultace a korekce nastavení nástroje
WebSphere Portal Server

pro

Západočeská univerzita
Univerzitní 8
Plzeň, 306 14

F.3. Nabídka Cleverbee – portlet FileExplorer

Ukázka kalkulace nákladů na vývoj portletu v komerčních cenách ilustrovaná na příkladu portletu FileExplorer (viz strana 73).



Cleverbee s.r.o.
Nádražní 56, 150 00 Praha 5, Czech Republic
tel. +420 2 5110 6110, fax +420 2 5110 6111

3 Project costs

3.1 Time consumption

Development:

Item	Work time
Research, concept	2 Mandays
Backend impementation	1 Mandays
Tree and list compoment	3 Mandays
Other function	
upload	1 Mandays
download	1 Mandays
email	1 Mandays
preview	2 Mandays
refresh, delete	0,5 Mandays
Edit mode	1 Mandays
Design	1 Mandays
Corrections based on errorreport	2 Mandays
Total development days	15,5 Mandays

Totals:

Item	Work time	Cost
Software development	15,5 Mandays	7 285 EUR
Testing/QM	2 Mandays	940 EUR
Unexpected tasks (development, testing/QM)	0 Mandays	
Project leadership	3 Mandays	1770 EUR
Total all project tasks	18,5 Mandays	9995 EUR

F.4. Další zdroje informací

- IBM portal support
<http://www-306.ibm.com/software/genservers/portal/support>
- IBM developerworks portal zone
<http://www-106.ibm.com/developerworks/websphere/zones/portal>
- IBM passport advantage
<http://www.ibm.com/software/passportadvantage>
- IBM WebSphere Developer Technical Journal
<http://www-106.ibm.com/developerworks/websphere/techjournal/>
- Looksmart archiv WebSphere Developer's Journal
http://www.findarticles.com/cf_0/m0MLX/mag.jhtml
- JSR168 Portlets & Java Portals
<http://groups.yahoo.com/group/portlets/>
- Opensource Portlets
<http://portlet-openscr.sourceforge.net>
- JSPOLYMPUS
<http://www.jspolympus.com>
- IBM portlet catalog
<http://www-106.ibm.com/developerworks/websphere/~zones/portal/catalog/>
- diskusní fóra IBM
<news://news.software.ibm.com>
<http://groups-beta.google.com/group/ibm.software.websphere.~.portal-server>
- WebSphere.ORG
<http://www.websphere.org>
- Bitpipe KnowledgeAlert
<http://ka.bitpipe.com>
- Portlas Magazine
<http://www.portalsmag.com>
- IBM Scholars Program
<http://www.developer.ibm.com/university/scholars>
- IBM PartnerWorld
<http://www.ibm.com/partnerword>
- Web Server Talk – WebSphere Portal Server
http://www.webservertalk.com/WebSphere_Portal_Server_81.htm