

RELACIJSKE PODATKOVNE BAZE IN SVETOVNI SPLET

Sebastian Lahajnar
PRIS Inženiring, Ljubljana, Slovenija
sebastian.lahajnar@pris-inz.si

Povzetek

Področje razvoja spletnih aplikacij za dostopanje do relacijskih podatkovnih baz je relativno nov segment informacijske tehnologije, saj je večji pomen dobilo šele v zadnjih štirih letih. Namen članka je kratka obravnava najpomembnejših tehnologij in jezikov za izgradnjo spletnih aplikacij, predstavitev treh novejših tehnik za izgradnjo spletnih podatkovnih baz ter predložitve ustreznega odločitvenega modela za njihovo ovrednotenje.

Abstract

The area of developing Web applications for accessing relational databases is a relatively new segment of information technology as it has become more important in the last four years. This article briefly covers the most important technologies and languages used today for building Web applications, it describes three recent techniques for implementing Web databases and finally it proposes an appropriate decision model for their evaluation.



1. Uvod

Hiter razvoj Interneta, predvsem storitve svetovnega spleta v devetdesetih letih, je presenetil marsikoga, tudi največje vizionarje na področju informacijske tehnologije. V kratkem času je postal nepogrešljiv del našega vsakdana, brez katerega si ni mogoče zamisliti moderne komunikacije in prenosa podatkov vseh vrst. Internet je posegel na različna področja našega življenja in v marsičem spremenil vsakdanjo prakso, navade ter način zabave in dela.

Predstavitev podatkov na svetovnem spletu oziroma v okviru intranetov znotraj podjetij neposredno iz podatkovnih baz ima veliko pozitivnih učinkov: povečanje funkcionalnosti spletnih predstavitev, poenostavitev predstavitev velikih dokumentnih zalog podatkov, centraliziranje dostopa do različnih podatkovnih baz z uporabo enotnega uporabniškega vmesnika, povečanje interaktivnosti spletnih predstavitev itd. Podjetja danes uporabljajo podatkovne baze v spletnih predstavitev predvsem za vzdrževanje podatkov o poslovnih partnerjih, za baze znanja o tehničnih karakteristikah, cenah proizvodov, za statistike dostopov do spletnih strani, prodajnih trendov, za vzdrževanje podatkov o zalogah, za poslovne dogovore in pravila, za navodila povezana z naročanjem, dobavo in plačilom proizvodov itd.

2. Tehnologije in jeziki za razvoj spletnih aplikacij

Dostop do podatkovnih baz v svetovnem spletu je omogočen preko spletnih aplikacij, za katerih razvoj se je sprva uporabljala predvsem kombinacija označevalnega jezika HTML na odjemalčevi strani spletne povezave in programov CGI na strani spletnega strežnika. Pojav novih tehnologij v preteklih letih pa je omogočil preprostejše in hkrati funkcionalno zmogljivejše dostope do podatkovnih baz in s tem odprla vrata za njihovo vsesplošno vključevanje v spletne predstavitve in elektronsko trgovanje. Najpomembnejše podporne tehnologije, na katerih dandanes temeljijo tehnike za dostop do relacijskih podatkovnih baz prek svetovnega spleta, so:

HTML: Temelj storitve svetovnega spleta so večpredstavni dokumenti oziroma hiperbesedila, zapisana v obliki HTML. Čeprav mnogi obravnavajo HTML kot programski jezik, se večina nagiba k mnenju, da njegova sintaksa, predvsem pa pomanjkanje določenih elementov, ki kategorizirajo druge jezike predvsem tretje generacije (podatkovni tipi, zanke, odločitveni stavki itd.), onemogočajo postavitev HTML-ja ob bok drugim programskim jezikom. S stališča izgradnje spletnih aplikacij je najpomembnejši element jezika obrazec HTML. Obrazci omogočajo

opremljanje spletnih strani z elementi (vnosno polje, seznam, gumb itd.), ki so temelj vsakega dobrega uporabniškega vmesnika. Uporabnik se pri svojem delu neposredno sreča le s tem delom informacijskega sistema, saj ostali procesi delujejo v ozadju in so zanj nevidni. Uporabnost in prijaznost informacijske rešitve tako v največji meri ocenjujemo po izgledu in funkcionalnosti uporabniškega vmesnika, ki jo zagotavljajo grafični elementi objektnega modela HTML.

CGI (Common Gateway Interface): vmesniške specifikacije CGI omogočajo spletnim strežnikom izvajanje strežniških programov in vključevanje rezultatov obdelave v odgovore brkljalnikom. Določajo namreč standardni vmesnik za komunikacijo med spletnimi strežniki in zunanji programi, ki so lahko zaradi tehnološke neodvisnosti specifikacij razviti z uporabo najrazličnejših programskih jezikov in orodij. Določene pomanjkljivosti programov CGI (počasnost izvajanja, velika potratnost sistemskih virov, predvsem pomnilnika itd.), so narekovele pojav nekaterih novih tehnologij kot so ISAPI (Internet Server Applications Programming Interface) podjetja Microsoft in NSAPI (Netscape Server Applications Programming Interface) podjetja Netscape, ki v marsičem rešujejo navedene probleme, vendar le za spletne strežnike posameznih podjetij, saj niso uveljavljene kot standard.

Java: Java je visoko nivojski, objektno usmerjen, platformsko neodvisen programski jezik podjetja Sun Microsystems, ki vključuje različne pripomočke, predvsem v obliki knjižnic razredov, za delo z Internetom. Je hkrati prevajan in tolmačen programski jezik. Z uporabo prevajalnika se izvorna koda najprej prevede v preproste binarne ukaze, podobne procesorski strojni kodi. Prevedena javanska koda, imenovana tudi J-code, se nato izvaja s pomočjo ustreznega tolmača. Le-ta je relativno majhen in preprost, implementirati ga je mogoče na vseh platformah. V večini sistemov je napisan v programskih jezikih C ali C++. Ponavadi obstaja kot samostojna aplikacija, lahko pa je umeščen tudi v druge programske produkte, predvsem v spletne brkljalnike (prav to omogoča izvajanje programčkov na spletnih straneh). Glavna naloga tolmača je, da sproti prilagaja vmesno kodo lastnostim posameznega mikroprocesorja z vzpostavitev navideznega stroja v pomnilniku.

Najbolj znan razred programskega jezika Java je razred Applet. S svojimi lastnostmi in metodami omogoča izgradnjo programov, ki se izvajajo neposredno na spletnih straneh in s tem vpeljuje v svetovni splet novo dimenzijo. Za razliko od programov CGI, ki se izvajajo na spletnem strežniku, na

odjemalčev sistem pa se prenesejo zgolj rezultati, se spletni programčki v jeziku Java izvajajo neposredno pri odjemalcu. Java tako omogoča razvoj programskih rešitev za svetovni splet, ki v ničemer ne zaostajajo za klasičnimi aplikacijami, delujočimi v okviru posameznega operacijskega sistema. Z uporabo razredov knjižnice AWT (Abstract Windowing Toolkit) je možno zgraditi kompleksen uporabniški vmesnik, ki se samodejno prilagaja grafičnim značilnostim različnih spletnih brkljalnikov in platform. V današnjem času, ko v okviru informacijskih sistemov pridobivajo na pomenu predvsem skladišča podatkov, je življenjskega pomena za obstoj Jave zmožnost njenega povezovanja s podatkovnimi bazami. Knjižnica razredov JDBC (Java Database Connectivity), ki je danes kot tudi ves programski jezik, v nenehnem razvoju, je namenjena prav neposrednemu dostopu do različnih podatkovnih baz. JDBC skupaj s spletnimi programčki omogoča nov pogled na dostopanje do podatkovnih baz prek svetovnega spleta, ki je bil doslej v glavnem v domeni programov CGI.

JavaScript/VBScript: skriptna jezika podjetij Netscape in Microsoft dodajata spletnim stranem povsem nove zmožnosti. Omogočata namreč takojšen prikaz in obdelavo podatkov brez čakanja na spletni strežnik. Skriptni programi so vključeni v kodo HTML spletne strani, njihovo izvajanje pa omogočajo v brkljalnike vgrajeni tolmači. JavaScript in VBScript vsebujeta več razredov in metod, ki zagotavljajo kakovostnejšo interakcijo med uporabniki in spletnimi stranmi. Med najbolj uporabne sodi vsekakor sistem za razpoznavanje dogodkov, ki omogoča odzivanje na vnose podatkov v obrazce, premike miške itd. Za ustrezen odziv poskrbijo vnaprej pripravljene funkcije brez dodatnega prenosa podatkov po omrežju. Poleg izvajanja skriptnih jezikov na strani odjemalca se danes vse bolj uveljavlja tudi njuna uporaba na spletnih strežnikih v okviru izgradnje strežniških skriptnih programov (na primer tehnologiji ASP in LiveWire).

ActiveX: Tehnologija ActiveX je rezultat razvoja in prenove starejših Microsoftovih tehnologij VBX (Visual Basic Controls) in OCX (OLE Custom Controls). Komponente programske opreme, ki so razvite v skladu s tehnologijo ActiveX, nosijo naziv gradniki ActiveX (ActiveX controls). Gradniki ActiveX predstavljajo danes enega najpreprostejših načinov za razširitev funkcionalnosti spletnih strani, saj poskušajo združevati pripravnost spletnih programčkov v jeziku Java ter zmogljivost in funkcionalnost Netscapeovih dodatkov plugin. Podobno kot spletni programčki se lahko gradniki ActiveX samodejno prenesejo na odjemalčev sistem v primeru, ko tam še niso nameščeni, oziroma je nameščena še stara različica.

Podobnost z dodatki plugin pa je v tem, da ko so gradniki ActiveX enkrat nameščeni, ostanejo na razpolago brkljalniku ves čas.

Za večino uporabnikov je integracija gradnikov ActiveX transparentna, ker se le-ti samodejno aktivirajo ob zagonu brkljalnika, v pomnilnik pa se naložijo šele ob odpiranju spletne strani. Gradniki ActiveX se nahajajo v pomnilniku le takrat, ko je to potrebno za prikaz vsebine spletne strani, in se odstranijo takoj potem, ko se uporabnik premakne na drugo spletno stran, ki gradnikov ne uporablja. Širša uporabnost gradnikov preprečuje predvsem njihova omejenost na okolja Windows, kar je velika slabost v primerjavi s platformsko neodvisnimi javanskimi programčki.

3. Predstavitev tehnik

V nadaljevanju so podrobneje predstavljene in analizirane tri novejšje tehnike za dostop do relacijskih podatkovnih baz prek svetovnega spleta. Tehnike je moč razdeliti v dve skupini in sicer na tiste, pri katerih se večina aktivnosti izvaja na strani odjemalca (Java - JDBC), in tiste, pri katerih se procesiranje izvaja na strani strežnika (ASP-ADO, Oracle Application Server PL/SQL). Vse omenjene tehnike sem uporabil na konkretnih primerih, saj sem želel preseči okvire teoretične obravnave. V računalništvu in informatiki smo vajeni, podobno kot tudi v drugih panogah, da proizvajalci povečujejo svoje rešitve, tako da je prava, kritična presoja možna šele po njihovi uporabi v praksi.

ASP - ADO

Tehnologija ASP (Active Server Pages) je rezultat Microsoftove vizije izgradnje spletnih aplikacij z uporabo kombinacije skriptnih jezikov in objektov ActiveX na strani spletnega odjemalca in strežnika. ASP sestavljata dve temeljni sestavini:

- **Skriptni jezik (JavaScript, VBScript):** omogoča interakcijo med strežniškimi objekti, manipulacijo z njihovimi lastnostmi in metodami, izvajanje funkcij in procedur, krmiljenje izvajanja programskega koda itd.
- **Strežniški objekti:** tvorijo jedro tehnologije ASP in zagotavljajo programerjem izrabo vseh njenih zmogljivosti. Strežniški objekti in skriptni jezik tvorijo skupaj z označevalnim jezikom HTML izvorni kod posamezne aktivne spletne strani, katerih medsebojno povezovanje omogoča izgradnjo dinamičnih spletnih aplikacij. Pri izdelavi spletnih aplikacij s tehnologijo ASP se uporabljajo metode in lastnosti predvsem naslednjih objektov: Request (uporabniška zahteva), Response (nadzor oblikovanja izhoda), Server (dostop do funkcij spletnega strežnika) in Application (stanje spletne aplikacije).

Ena glavnih prednosti izvajanja skriptnih programov na spletnem strežniku je možnost njihovega povezovanja s podatkovnimi bazami. V ta namen tehnologija ASP najbolje sodeluje z objektnim modelom ADO (ActiveX Data Objects). Sestavlja ga nabor objektov ActiveX, ki omogočajo dostop do podatkov in manipulacijo z njimi. ADO za razliko od drugih Microsoftovih podatkovnih modelov, ki uporabljajo tehnologijo ODBC (Open Database Connectivity), temelji na tehnologiji OLE DB. Največja razlika med omenjenima tehnologijama je vmesnik, za katerega se v primeru ODBC uporabljajo gonilniki v obliki knjižnic DLL (Dynamic Link Libraries), medtem ko ima gonilnike OLE DB realizirane kot implementacijo objektov ActiveX. To naj bi v teoriji pospašilo izvajanje aplikacij zaradi preprostejšje strukture gonilnika.

Preprosta uporaba ob vrsti ponujenih možnosti je ena glavnih prednosti tako tehnologije ASP kot tudi podatkovnega modela ADO. V obeh primerih je hierarhija objektov sila preprosta in omejena le na najpomembnejše. Prava moč se skriva v številnih metodah in lastnostih, ki pogojujejo in nadzorujejo njihovo obnašanje. ASP omogoča poleg izgradnje dinamičnih spletnih strani na podlagi rezultatov poizvedb tudi razvoj dinamičnih vhodnih obrazcev za vnos pogojev poizvedb. Tehnika ASP-ADO se tako že močno približuje dinamičnosti aplikacij razvitih s tehnologijo ActiveX ali Java, čeprav ju verjetno ne bo nikoli dosegla. Razlog se nahaja v njeni zasnovi, ki zahteva, da se celotna vsebina oblikuje že pred pošiljanjem spletne strani k odjemalcu. Kombinacija tehnologije ASP z objekti ADO za dostop do podatkovnih baz v okolju svetovnega spleta ima vsekakor več prednosti kot slabosti, seveda če se sprijaznimo z omejenostjo na Microsoftov spletni strežnik.

Java - JDBC

Ena od začetnih pomanjkljivosti Jave je vsekakor bila ta, da programski jezik ni vseboval neposredne podpore za delo s podatkovnimi bazami. Za odpravo omenjenega problema so različna podjetja začela pošiljati na tržišče svoje rešitve. Pri tem je šel najdlje Microsoft, ki je v svojem razvojnem okolju Visual J++ zagotovil povezavo Jave s podatkovnimi bazami preko COM (Component Object Model) objektnih podatkovnih modelov DAO (Data Access Objects) in RDO (Remote Data Objects). Vsekakor gre za zanimivo kombinacijo dveh tehnologij, ki ob določenih omejitvah (za marsikoga sicer nesprejemljivih) zagotavlja ustrezno povezavo. Se pa omenjena rešitev bistveno oddaljuje od same filozofije programskega jezika Java (platformska neodvisnost, prenosljivost itd.), zato v javanskih krogih ni bila sprejeta z navdušenjem.

Standard JDBC je bil razvit prav zaradi vse pogostejših zahtev po zagotovitvi standardiziranega vmesnika

za dostop do relacijskih podatkovnih baz. JDBC je rezultat sodelovanja podjetja Sun in nekaterih najbolj znanih proizvajalcev SUPB. Njihov glavni cilj je bila izgradnja podatkovnega vmesnika na nižjem nivoju, ki podpira dostop do podatkov z uporabo poizvedovalnega jezika SQL. Kot osnovo za novi vmesnik so uporabili vmesnik X/Open SQL CLI (Call Level Interface), ki predstavlja tudi temelje Microsoftovega vmesnika ODBC. Problem neposredne uporabe vmesnika ODBC je njegova arhitektura, ki je prilagojena programskemu jeziku C (klasični postopkovni programski jezik tretje generacije) in s tega stališča ni najbolj primerna za uporabo v Javi (popolnoma objektni programski jezik).

Vmesnik JDBC je sestavljen iz nabora razredov, shranjenih v javanskem paketu SQL. Njihove metode in lastnosti omogočajo nalaganje gonilnika za izbran SUPB, vzpostavljajo povezavo s podatkovno bazo, kreiranje in izvajanje stavkov SQL ter oblikovanje in prikaz rezultatov. Najpomembnejša značilnost JDBC je njegova neodvisnost od SUPB. Programer pri izgradnji aplikacije vedno uporablja le javanski vmesnik in se ne ozira na vrsto in karakteristike izbranega SUPB.

Za vzpostavitev povezave aplikacije, napisane v Javi, s podatkovno bazo je potrebno poleg vmesnika JDBC zagotoviti še ustrezen gonilnik za izbran tip SUPB. Dandanes je večina podjetij, ki se ukvarjajo z razvojem SUPB, že zagotovila gonilnike za svoje izdelke, pojavljajo pa se tudi številni izdelki neodvisnih proizvajalcev. Izbor primerne gonilnika je ključnega pomena pri izdelavi spletnih programčkov za dostop do podatkovnih baz, saj so njihove pristojnosti zaradi varnostnih zahtev zelo omejene.

Java tako vse od različice razvojnega kompleta JDK 1.1 omogoča izgradnjo nadvse dinamičnih in fleksibilnih programskih rešitev za dostop do relacijskih podatkovnih baz. Omenjena dinamičnost je posledica zmogljivosti, ki jih zagotavlja Java pri izgradnji spletnih programčkov za Internet. Le-ti se po funkcionalnem izgledu, dinamiki izvajanja operacij in uporabniškem vmesniku skoraj ne ločijo od klasičnih aplikacij. To je vsekakor velik napredek v primerjavi z drugimi tehnikami, ki za izdelavo uporabniškega vmesnika uporabljajo preprost objektni model HTML. Navkljub možnosti izgradnje kompleksnih obrazcev s pomočjo objektov HTML ima taka programska rešitev še vedno priokus spletne strani ne pa prave aplikacije.

Z vidika dinamičnosti in fleksibilnosti spletnih aplikacij Java v kombinaciji z vmesnikom JDBC dandanes nima prave konkurence. Še najbližja ji je Microsoftova tehnologija gradnikov ActiveX, ki pa ima kar nekaj pomanjkljivosti in omejitev povezanih z varnostjo odjemalčevega računalniškega sistema (gradniki ActiveX lahko za razliko od javanskih programčkov dostopajo do vseh resursov odjemalčevega sistema) in

prenosljivostjo objektov (namenjeni so predvsem okoljem Windows). V luči navedenih slabosti konkurenčnih tehnologij in ob dejstvu, da edino Java zagotavlja izgradnjo platformsko neodvisnih aplikacij, je potrebno analizirati tudi prednosti in slabosti tehnike Java-JDBC.

Razredi paketa SQL tvorijo razmeroma dobre temelje za vzpostavitev, vzdrževanje in nadzorovanje povezave s podatkovnimi bazami. Omogočena je uporaba vseh standardnih stavkov poizvedovalnega jezika SQL, kar zagotavlja zadovoljivo mero funkcionalnosti pri dostopu in obdelavi podatkov. Vmesnik JDBC vsebuje tudi podporo izvajanju transakcij, kar zagotavlja varen prenos podatkov ter konsistentnost podatkovne baze. Navkljub predhodno poudarjenimi zmožnostmi pa je v nekaterih segmentih kar precej opazna mladost, s tem pa tudi nedorečenost samega vmesnika. Predvsem se moja kritika nanaša na implementacijo podatkovnega kazalca, saj vsebuje le eno metodo (next), ki omogoča premikanje med podatkovnimi zapisi v smeri od začetka do konca množice rezultatov poizvedbe. Na ta način postane problematična že sama izvedba premika na predhodni zapis, ki jo je potrebno reševati z uporabo drugih prijemov. Obstaja torej še kar nekaj pomanjkljivosti, ki jih bodo morali snovalci vmesnika JDBC in njegovih razredov odpraviti. Vsa tehnologija, povezana s programskim jezikom Java, predvsem pa njegov paket za povezavo s podatkovnimi bazami, je šele v začetni fazi razvoja in pričakovati je veliko sprememb in izboljšav že v bližnji prihodnosti. Čeprav se dandanes funkcionalnost vmesnika JDBC še ne more primerjati z nekaterimi drugimi podatkovnimi modeli, so njegovi temelji (tehnologija programskega jezika Java) garant nadaljnega uspešnega razvoja in uporabe.

Oracle Application Server - modul PL/SQL

Oracle Application Server (leta 1998 je doživel svojo četrto različico) je temelj Oracleove vizije izgradnje tri nivojskih sistemov skladno z arhitekturo omrežnega (internetnega) računalništva. Jedro arhitekturnega modela tvori aplikacijski strežnik, ki nadzoruje izvajanje aplikacij in zagotavlja prenos podatkov med lahkim odjemalcem in podatkovnim strežnikom (preostala dva nivoja). Aplikacijski strežnik vsebuje več tehnik za dostop do relacijskih podatkovnih baz, ki večinoma temeljijo na programskih jezikih PL/SQL in Java.

Oracleov aplikacijski strežnik je sestavljen iz naslednjih komponent:

- Spletni poslušalec (Web Listener): vzdržuje povezavo med spletnimi odjemalci in aplikacijskim strežnikom z uporabo standardnih internetnih protokolov. Spletni poslušalec posluša prihajajoče zahteve in jih predaja v nadaljnjo obdelavo spletnemu posredovalcu zahtev.

- Spletni posredovalec zahtev (Web Request Broker): gre za najbolj inovativen del aplikacijskega strežnika, ki je zasnovan na tehnologiji CORBA. Uporablja se za obdelavo in preusmerjanje zahtev skladno z identificiranimi parametri. Po določitvi tipa zahteve Web Request Broker prepusti njeno nadaljnje izvajanje izbranemu programskemu modulu (cartridge). Poleg samega preusmerjanja zahtev je naloga spletnega posredovalca zahtev tudi zagotavljanje štirih temeljnih servisov (transakcijski, avtentifikacijski, medmodularni in skladiščni) vsem modulom aplikacijskega strežnika.
- Programski Moduli (cartridges): predstavljajo izvršne programske module, implementirane na aplikacijskem strežniku, ki zagotavljajo z izvajanjem določenih servisov in funkcij ustrezno servisiranje prispelih zahtev. Oracle Application Server vsebuje več vnaprej definiranih modulov: PL/SQL (omogoča izvajanje spletnih aplikacij na podlagi dinamičnih podatkov iz podatkovnih strežnikov z neposredno uporabo shranjenih procedur in paketov v programskem jeziku PL/SQL), Java (vgrajen tolmač za izvajanje spletnih aplikacij v programskem jeziku Java), Perl (tolmač za CGI programe v programskem jeziku Perl) itd.

Modul PL/SQL je dandanes prav gotovo najpogostejše uporabljena tehnika aplikacijskega strežnika Oracle Application Server za dostop do SUPB podjetja Oracle. Izgradnja spletnih strežniških aplikacij poteka s pomočjo programskega jezika PL/SQL v obliki razvoja shranjenih postopkov in paketov. Prav možnost uporabe jezika PL/SQL je največja prednost omenjene tehnike pred drugimi (Java cartridge, Perl cartridge itd.). PL/SQL je namreč osnovni programski jezik Oracleovih podatkovnih strežnikov in razvojnih okolij (Developer/2000, Designer/2000), v katerem je realiziranih večina aplikacij tipa odjemalec-strežnik. Programerji Oracleovih podatkovnih baz so tako z modulom PL/SQL dobili idealno razvojno orodje za izdelavo dinamičnih spletnih aplikacij, saj zahteva le minimalno dodatno izobraževanje.

Shranjene procedure predstavljajo programske konstrukte v jeziku PL/SQL, shranjene neposredno na podatkovnem strežniku. Skupaj z odzivnimi procedurami (prožilci) tvorijo temelje Oracleove metodologije izgradnje aplikacij za dostop do podatkov. Izvajajo se neposredno na podatkovnem strežniku in s tem zmanjšujejo promet v omrežju. V primeru arhitekture odjemalec-strežnik se shranjene procedure kličejo neposredno z odjemalca, v primeru omrežne arhitekture pa je to naloga aplikacijskega strežnika. Programski jezik PL/SQL vsebuje vrsto vnaprej pripravljenih procedur in funkcij, namenjenih predvsem zajemanju in obdelavi podatkov v Oracleovih podat-

kovnih strežnikih. Aplikacijski strežnik Oracle Application Server prinaša kar nekaj novih paketov PL/SQL, ki omogočajo izgradnjo podatkovno aktivnih spletnih aplikacij: HTP in HTF (procedure in funkcije za implementacijo označb jezika HTML), OWA_COOKIE (servisi za delo s tako imenovanimi HTTP kolački - omogočajo hranjenje podatkov v okviru ene seje), OWA_UTIL (uporaba sistemskih spremenljivk CGI, večvrednostnih parametrov) itd.

Pri delu s standardnimi paketi programskega jezika PL/SQL pogrešamo predvsem večjo podporo podatkovnim kazalcem, kar je bil tudi glavni očitak javanskemu vmesniku JDBC. PL/SQL sicer vsebuje programski konstrukt kazalec, vendar se le-ta uporablja le za zajem podatkov iz tabel in še to zgolj enosmerno. Kazalci v shranjenih procedurah tako ne omogočajo dodajanja novih zapisov, brisanja in spreminjanja obstoječih, iskanja po izbranih kriterijih itd. Oracle je v ta namen že pred časom razvil nekaj dodatnih paketov PL/SQL in jih vključil v razvojno okolje Developer/2000. Žal omenjenih paketov ni moč uporabiti za razvoj shranjenih procedur na podatkovnem strežniku in skladno s tem tudi ne za izgradnjo spletnih aplikacij z uporabo modula PL/SQL aplikacijskega strežnika.

Predstavljeni modul aplikacijskega strežnika je le ena od številnih tehnik, ki jih podjetje Oracle ponuja za dostop do svojih podatkovnih strežnikov. Namenjen je predvsem programerjem z izkušnjami pri uporabi programskega jezika PL/SQL, medtem ko bodo drugi verjetno raje posegli po modulih Java in Perl, oziroma uporabili druge produkte kot so SUPB Oracle 8i ali strežnike Oracle Forms/Reports iz družine Developer/2000.

4. Model ocenjevanja tehnik

Za informacijsko tehnologijo velja, podobno kot za večino drugih področij, da nikoli ne obstaja en sam zmagovalc, pa naj gre za strojno ali programsko opremo. Obstajajo le izdelki, ki so v določenem trenutku, v določenih pogojih uporabe in za določen krog uporabnikov, za korak pred ostalimi. V kolikor izvedemo KAJ-ČE analizo in spremenimo pogoje uporabe oziroma krog uporabnikov, se rezultati ponavadi zelo spremenijo. Navedeno spoznanje velja tudi v primeru ocene tehnik za dostop do relacijskih podatkovnih baz prek svetovnega spleta.

Izbor primerne tehnike je v praksi vse prej kot svoboden, saj nanj vpliva veliko dejavnikov, povezanih predvsem z obstoječo informacijsko tehnologijo v podjetju. Dandanes ima že vsaka večja organizacija svoj lasten informacijski sistem, ki je tudi osnova za izgradnjo spletnih aplikacij. Le-te so tako v večini primerov zgolj nadgradnja obstoječih informacijskih sistemov, kar zelo omejuje možnosti izbora tehnik.

S stališča informacijske tehnologije so najpomembnejši naslednji dejavniki:

- **Operacijski sistem:** pogojuje izbor vse preostale programske opreme, od SUPB do spletnih in aplikacijskih strežnikov.
- **SUPB:** čeprav dandanes vmesnik ODBC omogoča dostop do najrazličnejših podatkovnih virov, omenjena rešitev ni optimalna (potreba po transformaciji poizvedb). Prav zavoljo tega nekatera podjetja ponujajo rešitve v obliki posebnih gonilnikov, prilagojene določenemu tipu podatkovnega strežnika (gonilniki JDBC za različne podatkovne baze, prilagojenost Oracleovega aplikacijskega strežnika podatkovnim strežnikom istega podjetja itd.). Vse to zožuje izbor potencialnih tehnik, pri čemer pa je še vedno na voljo dovolj manevrskega prostora.
- **Spletni (aplikacijski) strežnik:** predstavlja najpomembnejši dejavnik pri izboru ustrezne tehnike. Vsi pomembnejši spletni strežniki podpirajo izvajanje programov CGI ter spletnih programčkov v jeziku Java, poleg tega pa pogosto vsebujejo še lastne tehnologije za izdelavo strežniških programov (NSAPI, ISAPI itd.). Zaradi vse večjih potreb po povezovanju s podatkovnimi bazami spletni strežniki že vsebujejo tudi vrsto individualnih tehnik za dostop do podatkov (ASP-ADO, Internet Database Connector podjetja Microsoft, LiveWire podjetja Netscape, modul PL/SQL podjetja Oracle itd.), kar po eni strani v marsičem olajša delo programerjem, po drugi strani pa zožuje izbiro na dve ali tri možnosti.

Obstoječi informacijski sistem lahko torej znatno omeji možnost izbora najprimernejše tehnike. Posledice tega so različne in se kažejo v daljšem razvoju, neoptimalni izrabi programske in strojne opreme ter v funkcionalni omejenosti spletnih aplikacij. Priložnost za vpeljavo novih tehnologij in rešitev, povezanih s svetovnim spletom, se ponuja predvsem ob prenovi informacijskih sistemov, ki se izvaja danes in se bo tudi v bližnji prihodnosti izvajala v velikem številu podjetij. Vsaka prenova informacijskega sistema zahteva uvedbo novih programskih rešitev, kar je povezano z nabavo nove programske in strojne opreme. Prav tu vidi mo idealno priložnost za neodvisen pristop k izboru najprimernejše tehnike, saj je vpliv različnih dejavnikov najmanjši.

V nadaljevanju je podan primer odločitvenega modela za ocenjevanje in izbor primerne tehnike, pri čemer odločanje ni pogojeno z nekim vnaprej definiranim stanjem informacijske infrastrukture. Tehnike so obravnavane in medsebojno primerjane z različnih vidikov, ki pripadajo eni od dveh osnovnih skupin: tehnični ali razvojni. Tehnični vidiki se nanašajo na tehnične karakteristike posameznih tehnik, medtem ko razvojne obravnavajo tehnike s stališča programerjev

spletnih programskih rešitev za dostop do relacijskih podatkovnih baz.

Tehnični vidiki:

- **Funkcionalnost:** obravnava število, kakovost podatkovnih objektov, popolnost in celovitost njihovih metod. Funkcionalni vidik ugotavlja, katere možnosti nudi posamezna tehnika za implementacijo načrtovanih spletnih programskih rešitev.
- **Dinamičnost:** ocenjuje zmožnost tehnike za izgradnjo čim bolj dinamičnih, odzivnih in uporabniško prijaznih spletnih aplikacij oziroma njihovih uporabniških vmesnikov. Vidik dinamičnosti obravnava predvsem zunanji izgled spletnih programskih rešitev in njihovo podobnost z ekvivalentnimi programi tipa odjemalec-strežnik.
- **Prenosljivost:** ugotavlja prenosljivost spletnih aplikacij, uresničenih z obravnavanimi tehnikami med različnimi operacijskimi sistemi in SUPB. Poleg neodvisnosti programskih rešitev na strani strežnika, zajema vidik prenosljivosti tudi obravnavo odjemalca, predvsem v smislu zahtevane programske opreme (spletni brkljalniki) za uporabo specifične tehnike.

Razvojni vidiki tehnik:

- **Zahtevnost:** ukvarja se z oceno, kako težavna je izdelava spletnih aplikacij z obravnavano tehniko. V tem okviru ocenjujemo zgradbo in povezanost podatkovnih objektov, zahtevnost uporabe njihovih metod in lastnosti, obsežnost potrebnega programskega koda, razpoložljivost čarovnikov ali drugih orodij za hiter razvoj aplikacij itd.
- **Uporabljenost:** še tako funkcionalne, dinamične, preproste tehnike imajo lahko posamezne pomanjkljivosti, gledano predvsem s stališča nacionalnih posebnosti (jezik, datumski, urini formati itd.), ki omejujejo ali povsem onemogočajo njihovo uporabo v praksi.

Oceno posamezne tehnike sem izračunal s pomočjo utežene vsote vidikov, pri čemer se posamezni vidik ocenjuje z oceno od 1 do 10:

$$\text{Rezultat} = 30\% * (30\% * \text{zahtevnost} + 70\% * \text{uporabljenost}) + 70\% * (50\% * \text{funkcionalnost} + 30\% * \text{dinamičnost} + 20\% * \text{prenosljivost})$$

Izbor uteži daje vedeti, da sem dal večjo težo tehničnim ne pa razvojnim vidikom. Pri tem sem imel v mislih predvsem razvoj kompleksnih spletnih aplikacij, kjer sta najpomembnejši prav funkcionalnost in dinamičnosti, ne pa hiter in preprost razvoj. Iz formule je nadalje razvidno, da dajem veliko večji pomen uporabnosti tehnik kot pa njihovi preprostosti ter da je vidik funkcionalnosti osrednji tehnični vidik. Tehnično

manj podkovani uporabniki bi verjetno dali prednost preprosti uporabi tehnik, medtem ko izkušeni programerji tega vidika sploh ne bi upoštevali. Sam sem naredil kompromis, ki se kar najbolje ujema z mojimi pogledi na razvoj spletnih aplikacij za dostop do podatkovnih baz.

Tehnike/Vidiki	Zahtevnost	Uporabnost	Funkcionalnost	Dinamičnost	Prenosljivost	Skupaj
ASP-ADO	5	9	9	7	2	7.24
Java-JDBC	3	9	6	9	9	7.41
Oracle AS PL/SQL	5	9	6	7	4	6.47

Slika 1: Tabela rezultatov ocenjevanja

Tabela rezultatov (slika 1) nazorno prikazuje, da imajo tehnike Java-JDBC, ASP-ADO in Oracle Application Server PL/SQL podobno skupno oceno, medtem ko se ocene po posameznih vidikih kar precej razlikujejo. Še najboljše se je odrezala tehnika Java-JDBC, katere edini pomanjkljivosti sta slabša funkcionalnost in visoka zahtevnost razvoja aplikacij. Na drugi strani je največja ovira tehnike ASP-ADO za pohod na vrh omejenost na Microsoftov spletni strežnik Internet Information Server in manjša dinamičnost v primerjavi s programskim jezikom Java. PL/SQL modul Oracle Application Server-ja se je prav tako izkazal kot nadvse zanimiva in uporabna tehnika, v kolikor za shranjevanje podatkov uporabljamo podatkovni strežnik podjetja Oracle.

Primerjava tehnik za dostop do relacijskih podatkovnih baz prek svetovnega spleta je pokazala, da navkljub različnim tehnološkim izhodiščem v splošnem ni neke tehnike, ki bi bistveno odstopala od ostalih. To po eni strani zagotavlja širok krog možnih kandidatov, po drugi strani pa otežuje samo odločitev, predvsem v luči razvoja tehnike v prihodnosti. Se pa lahko rezultati bistveno spremenijo, če v ospredje postavimo zgolj nekatere vidike tehnik (na primer funkcionalnost, prenosljivost itd.), pri čemer se preostali zanemarijo oziroma se jim nameni minimalna utež. V

tem primeru se izbor ponavadi zoži na dve ali tri tehnike, kar pomeni dobro podlago za končno odločitev.

5. Zaključek

Prihodnost razvoja spletnih aplikacij je v integriranih razvojnih okoljih, ki ne bodo ločevala izdelave spletnih od klasičnih programskih rešitev. Razvojni proces bo potekal na višjem nivoju, predvsem v obliki povezovanja vnaprej pripravljenih objektov. Ti bodo vsebovali različne vmesnike za dostop do podatkovnih baz, ki bodo abstrahirali vse podrobnosti in na ta način zagotovili hitro in učinkovito izgradnjo aplikacij. Čeprav je v zadnjem času opaziti kar nekaj poskusov v tej smeri, so tehnike za dostop do relacijskih podatkovnih baz prek svetovnega spleta šele na začetku svoje razvojne poti.

Literatura

1. Beveridge Tony, McGlashan Paul: High Performance ISAPI/NSAPI Web Programming. Albany: Coriolis Group Books, 1997.
2. Dynamic Information Systems: Oracle Web Application Server Handbook. Berkeley: McGraw-Hill, 1998.
3. Fleet Dina: Teach Yourself Active Web Database Programming in 21 Days. Indianapolis: Sams.net, 1997.
4. Greenwald Rick: Using Oracle Web Application Server 3. Indianapolis: Que Corporation, 1997.
5. Gundavaram Shishir: CGI Programming on the World Wide Web. Bonn: O'Reilly, 1996.
6. Lester S. Christopher: Database Programming with Visual J++ . Indianapolis: Que Corporation, 1997.
7. Mesojedec Uroš: Java, programiranje za Internet. Ljubljana: Pasadena, 1996.
8. Swank Mark, Kittel Drew, Spenik Mark: World Wide Web Database Developer's Guide with Visual Basic 5. Indianapolis: Sams.net, 1997.
9. Taylor Art: JDBC Developer's Resource: Database Programming on the Internet. Menlo Park: Prentice Hall, 1997.

Mag. Sebastian Lahajnar je diplomiral leta 1997 na Fakulteti za računalništvo in informatiko Univerze v Ljubljani. Po diplomiji je vpisal podiplomski študij na Ekonomski fakulteti, smer Informacijsko upravljalne vede, in leta 1999 zagovarjal magistrsko delo z mentorjem prof. dr. Borko Jerman Blažič. Zaposlen je kot razvijalec v podjetju PRIS Inženiring, kjer se ukvarja z razvojem poslovnih informacijskih sistemov.