

IZAZOV SINKRONIZACIJE PODATAKA IZMEĐU MOBILNIH I STANDARDNIH BAZA PODATAKA

THE CHALLENGE OF DATA SYNCHRONIZATION BETWEEN MOBILE AND STANDARD DATABASES

Zlatko Stapić, prof. dr. sc. Neven Vrček

SAŽETAK

Trendovi u razvoju programskih proizvoda naglašavaju sve češću primjenu i implementaciju mobilnih i web rješenja kao elemenata kompleksnog višekorisničkog sustava. Proizvođači različitih rješenja prepoznaju važnost primjene mobilnih aplikacija, ali se susreću i s problemom implementacije sinkronizacije podataka između mobilnih baza i standardnih sustava za upravljanje bazom podataka. Ovaj rad sistematski prikazuje pristup odabiru najprikladnijeg načina sinkroniziranja podataka između spomenutih sustava uzimajući u obzir sve važne parametre kao što su smjer, brzina i učestalost sinkronizacije, količina podataka koju je potrebno sinkronizirati ali i druge funkcionalne zahtjeve kao što su mogućnost nepovezanog (eng. offline) rada i slično. Rad također kratko opisuje i osvrće se na različite pristupe sinkronizaciji podataka primjenom Microsoft i drugih često korištenih tehnologija.

ABSTRACT

Trends in developing software products emphasize more frequent use and implementation of mobile and web solutions into complex multi-user systems. Software manufacturers recognize the importance of implementation of mobile applications, but are faced with the problem of implementing data synchronization between mobile database and standard database management system. This paper presents a systematic approach to selecting the most appropriate way of synchronizing data between these systems, taking into account all relevant parameters such as direction, speed and frequency of synchronization, the amount of data that needs to be synchronized et cetera. On the other hand, functional requirements such as the application ability to work offline and alike are also taken into consideration. The paper also briefly describes and reflects on different approaches to data synchronization using Microsoft and other commonly used technologies.

1. UVOD

Upotreba i korištenje mobilnih uređaja novijih generacija (pametnih telefona, dlanovnika – u daljnjem tekstu PDA uređaja) posljednjih nekoliko godina raste velikom brzinom. Eksponencijalno poboljšanje performansi spomenutih uređaja po pitanju brzine, razlučivosti, ugrađene ili proširive memorije, ugrađenih dodataka i podrške različitim protokolima, učinilo je uključivanje mobilnih uređaja u različite sfere osobnog, društvenog ali i profesionalnog života. Od osnovne funkcionalnosti upravitelja osobnim informacijama (eng. Personal Information Manager - PIM), PDA uređaji se sve više koriste u svrhu izvođenja različitih aplikacija koje se mogu besplatno ili relativno jeftino preuzeti s Interneta. PDA uređaji se također koriste u svrhu izvođenja aplikacija koje su sastavni dio kompleksnijih sustava koji su podrška poslovanju. Iako je pojam *Personal Digital Assistant (PDA)* osmislio predsjednik kompanije Apple Computer John Sculley tek početkom 1992. godine kako bi opisao računalo veličine dlana kojim se upravlja pomoću igle (male olovke, eng. stylus) i LCD zaslona [13], povijest nastanka i primjene PDA uređaja seže u 80-te godine prošlog stoljeća, kada je nastala ideja kreiranja malog računala koje bi se držalo na dlanu te služilo za pohranu adresa, brojeva telefona i dnevnih obaveza vlasnika. Kronološki, povijest nastanka i razvoja PDA uređaja je izgledala ovako [14], [13], [4]:

- 1980 – Psion osmislio koncept PDA uređaja
- 1983 – Prvi PDA uređaj (CASIO PF-3000)
- 1984 – Psion predstavio prvi organizator
- Psion 1, Psion 2

- Psion 3a – prvi uređaj koji je imao mogućnost povezivanja s stolnim računalom i prijenosa podataka (sinkronizacije)
- Psion 5 – još bolje performanse i nastavak uspjeha 3a serije
- 1993 – Apple predstavio Newton MessagePad – među prvim uređajima koji su imali ekrane osjetljive na dodir i ugrađen softver za prepoznavanje rukopisa
- 1996 – US Robotics predstavlja Palm Pilot; Microsoft otkriva plan razvoja Windows CE operacijskog sustava za dlanovnike
- 1997 – Apple predstavlja eMate ali zbog nedovoljno brzog i pouzdanog softvera za prepoznavanje rukopisa, od 1998 Apple zaustavlja razvoj PDA uređaja
- 1998 – Microsoft predstavlja Palm PC koji pokreće Windows CE 2.0 operacijski sustav; na tržištu se pojavljuju i proizvođači Sony, Philips, Sharp, HP i drugi.
- 1999 – Palm VII podržava bežični pristup Internetu.
- 2000 – Microsoft izdao Pocket PC
- 2002 – Predstavljen BlackBerry; predstavljen Pocket PC 2002
- 2004 – Predstavljen Windows CE 5
- 2005 – Microsoft predstavio Windows Mobile 5 operacijski sustav
- 2007 – Apple predstavlja iPhone; predstavljen Windows Mobile 6; osnovano udruženje Open Handset Alliance; predstavljen Android – operacijski sustav baziran na otvorenom kodu
- 2008 – Predstavljen iPhone 3G; Windows Mobile 6.1

- 2010 – Google predstavlja Nexus One; PDA uređaji bazirani na Androidu postaju tržišni lideri već u prvom kvartalu godine.

Prikazana kratka povijest razvoja PDA uređaja predstavlja samo dio burnog tržišnog natjecanja između nekoliko desetaka proizvođača koji su prepoznali potencijal koji ima sveprisutna primjena mobilnih uređaja. Iako su osobne aplikacije, ali i aplikacije informativnog ili zabavnog sadržaja postale podatkovno zahtjevne, te zahtijevaju redovitu sinkronizaciju podataka s drugim uređajima (stolnim računalima, web servisima i slično), ovaj rad je prvenstveno fokusiran na izazove implementacije sinkronizacije podataka između mobilnih i standardnih baza podataka u poslovno orijentiranim sustavima.

Promatrajući primjenu PDA uređaja s poslovne perspektive, možemo reći da je njihova najvažnija prednost [2] što omogućuju korištenje ili kreiranje kopije ili dijela potrebnih poslovnih podataka pri čemu korisnikova mobilnost nije uskraćena. Osnova spomenute funkcionalnosti je mogućnost jednostavne replikacije i spajanja podataka između PDA uređaja i izvora podataka koji su podrška cjelokupnom sustavu.

Budući da implementacija sinkronizacije podataka između spomenutih sustava nije jednostavan i trivijalan zadatak proizvođača softvera, u ovom radu ćemo pokušati sistematizirati pristupe i načine sinkronizacije podataka, uzimajući u obzir sve važne parametre kao što su smjer, brzina i učestalost sinkronizacije, količina podataka koju je potrebno sinkronizirati ali i druge funkcionalne zahtjeve kao što su mogućnost nepovezanog (eng. offline) rada i slično. Rad također kratko opisuje i osvrće se na različite pristupe sinkronizaciji podataka primjenom Microsoft i drugih često korištenih tehnologija.

2. POTREBA ZA SINKRONIZACIJOM

Prilikom razvoja sustava koji će uključivati i mobilnu komponentu, proizvođači se često suoče s poslovnim izazovima kao što su [9]:

- Kako osigurati ažurne podatke za radnike koji dislocirano obavljaju poslovne zadatke?
- Kako rezultate dislociranog posla učitati na glavni server i učiniti dostupnima u sustavu?
- Kako osigurati ažurnost prilikom promjene relacijskih shema, aplikacija i nadogradnje softvera?
- Kako pratiti i nadgledati tko posjeduje koji mobilni uređaj u velikom poslovnom sustavu?

Navedena pitanja su različito rješavana tijekom posljednjih nekoliko godina, ali danas je općeprihvaćeno rješenje primjena male, ugrađene, mobilne baze podataka koja bi imala mogućnosti sinkronizacije podataka i osiguranje mogućnosti upravljanja uređajima (eng. Device Management Capability) za administratore.

Isto tako, jedno od rješenja je arhitektura sustava u kojoj su i mobilni uređaji neprestano povezani na poslovnu mrežu te imaju mogućnost dohvata i promjene poslovnih podataka u realnom vremenu.

Ipak, mnogi proizvođači baza podataka prepoznali su probleme koji se javljaju u poslovnom svijetu, a koji rezultiraju potrebom mobilnog nepovezanog (eng. offline) rada, te potom i potrebom sinkronizacije podataka između mobilnih i standardnih baza podataka. Tako na primjer Oracle navodi [12] kako uvijek nije moguće implementirati idealni poslovni scenarij u kojem su mobilne aplikacije preko postojeće Internet veze neprestano spojene na glavnu bazu podataka. Oracle navodi kako je stalna povezanost na Internet u različitim poslovnim scenarijima [12]:

- *nemoguća*. Mnoge poslovne aktivnosti (na primjer servisiranje kupaca) odvijaju se u područjima gdje bežični pristup mreži nije moguć ili je nepouzdan.
- *nep praktična*. Iako propusnost današnjih bežičnih mreža zadovoljava različite poslovne scenarije, još uvijek je nedovoljna za podatkovno jako zahtjevne operacije koje se mogu provoditi na mobilnim uređajima kao što je na primjer definiranje kompleksnih prodajnih kvota.
- *nepoželjna*. U različitim scenarijima u kojima mobilna aplikacija koristi osjetljive podatke, sigurnosne politike nalažu da je stalni pristup Internetu nepoželjan ili čak zabranjen. Također, neisplativo je koristiti stalnu Internet vezu u poslovnim scenarijima u kojima se poslovni podaci neznatno mijenjaju tijekom nepovezanog rada.

Različiti drugi poslovni scenariji, koji zahtijevaju rad s podacima na više udaljenih i promjenjivih lokacija istovremeno, ili zahtijevaju rad s podacima koje nije moguće pohraniti u bazu podataka također mogu rezultirati potrebom manje ili više učestalog sinkroniziranja podataka između mobilnih uređaja ili između mobilnog uređaja i standardne baze podataka.

3. PARAMETRI SINKRONIZACIJE

Prilikom definiranja dizajna i arhitekture te razvoja programskih proizvoda potrebno je obratiti pozornost na nekoliko parametara budućeg programskog proizvoda o kojima će odabir sinkronizacijskog mehanizma i implementacija sinkronizacije podataka u velike ovisiti.

Ukoliko je u pitanju razvoj programskog proizvoda koji koristi mobilnu komponentu koja će imati stalan, neograničen, siguran i pouzdan pristup Intranetu ili Internetu te na taj način i korporativnim podacima, najjednostavnije i najefikasnije rješenje je implementacija direktnog pristupa mobilne aplikacije na standardnu bazu podataka. Spomenuta implementacija će rezultirati ažurnim podacima kod svih korisnika u realnom vremenu.

S druge strane, ukoliko je sinkronizacija potrebna (pogledati prethodno poglavlje), u obzir treba uzeti nekoliko važnih parametara:

- *smjer sinkronizacije podataka* – u različitim poslovnim scenarijima nije uvijek potrebno vršiti sinkronizaciju podataka u oba komunikacijska smjera. Podaci ponekad moraju biti ažurirani samo na mobilnom uređaju, ponekad (ali rijetko) samo na glavnom serveru, a ponekad se poslovni događaji koji rezultiraju izmijenjenim stanjem događaju i unutar poslovnog sustava (na glavnoj bazi), ali i u okviru dislociranih poslovnih procesa (na mobilnoj bazi). Prema navedenom, odabir sinkronizacijskih mehanizama može ovisiti o smjeru sinkronizacije.
- *učestalost sinkronizacije* – potreba za učestalošću sinkronizacije u pojedinim poslovnim sustavima može varirati u intervalima od tjedne ili mjesečne sinkronizacije do dnevne ili satne sinkronizacije. Ukoliko se sinkronizacija odvija rijetko, dužina trajanja sinkronizacije nije ključan čimbenik, ali ukoliko se sinkronizacija odvija češće, potrebno je implementirati onaj sinkronizacijskih mehanizam koji će biti zadovoljavajuće brzine.
- *brzina sinkronizacije* – različite implementacije sinkronizacijskih mehanizama imaju značajnu razliku u trajanju sinkronizacije. Na primjer, sinkronizacija koja je implementirana na standardnim web servisima traje nekoliko puta duže nego sinkronizacija koja je implementirana pomoću direktne TCP veze između dva uređaja. Isto tako, sinkronizacija koja uključuje razmjenu cjelokupnih i

opsežnih podataka (tablica) ne može biti provedena u istom vremenu kao sinkronizacija koja se bazira na razmjeni izmijenjenih podataka.

- **zahtjevi za sigurnošću podataka** – mobilni uređaji najčešće pristupaju Internetu ili Intranetu pomoću različitih bežičnih rješenja. Zahtjevi za sigurnošću podataka mogu rezultirati zabranom sinkronizacije ukoliko je mobilni uređaj spojen pomoću nesigurnog komunikacijskog kanala, ili mogu biti implementirani na način da prema potrebi uključe dodatne sigurnosne mehanizme prilikom prijenosa podataka.
- **uključeni uređaji u sinkronizaciju** – važan parametar odabira sinkronizacijskih mehanizama je njihova ovisnost o drugim infrastrukturnim uređajima kao što su web ili mobilni serveri, agenti ili slično. Potreba za direktnom sinkronizacijom između dvaju mobilnih uređaja, bez posredovanja servera, s liste mogućih rješenja će maknuti većinu implementacija postojećih proizvođača mobilnih baza.

Iz navedenog možemo zaključiti da nije uvijek moguće koristiti sinkronizacijske mehanizme koje pružaju proizvođači baza podataka, te je potrebno implementirati vlastitu sinkronizacijsku logiku, koja će zadovoljiti specifične uvjete sustava po pitanju količine prenesenih podataka, brzine prijenosa, pouzdanosti, uređaja koji su uključeni u sinkronizaciju i slično.

4. SINKRONIZACIJSKI MEHANIZMI

4.1 Vlastita rješenja sinkronizacije

Vlastiti sinkronizacijskih mehanizmi se najčešće implementiraju uz uporabu direktne veze između sustava koji razmjenjuju podatke, te kako bi se osigurala takozvana *push* sinkronizacija. Rješenja variraju od onih koja koriste TCP ili UDP sokete do onih koja se baziraju na uporabi specifično izgrađenih web servisa.

U svakom slučaju, potrebno je implementirati vlastitu sinkronizacijsku logiku, vlastite mehanizme oporavka od pogreške te vlastite mehanizme rješavanja konflikata. U nastavku je prikazan primjer sinkronizacijske logike koja bi se mogla implementirati prilikom izrade vlastitog algoritma za sinkronizaciju podataka [2].

PDA zapis	Server zapis	Akcije pri sinkronizaciji
Dodan	Ne postoji	Dodati zapis na server
Ne postoji	Dodan	Dodati zapis na PDA
Obrisan	Bez promjene	Obrisati zapis sa servera
Bez promjene	Obrisan	Obrisati zapis sa PDA
Ne postoji	Obrisan	Bez akcija
Obrisan	Ne postoji	Bez akcija
Obrisan	Obrisan	Bez akcija
Obrisan	Promijenjen	Vratiti i izmijeniti na PDA
Promijenjen	Obrisan	Vratiti i izmijeniti na server
Promijenjen	Bez promjene	Promijeniti zapis na server
Bez promjene	Promijenjen	Promijeniti zapis na PDA
Promijenjen	Promijenjen	Kreirati dvije kopije koje prikazuju promjene nastale na PDA i na serveru, te obavijestiti korisnika o konfliktu.

Tablica 1. Sinkronizacijska logika

U scenariju u kojem se sinkronizacija provodi između više različitih uređaja (na primjer: više mobilnih uređaja i jednog servera ili jednog mobilnog uređaja i više servera), sinkronizacijski mehanizmi se znatno kompliciraju. Jedno od najčešćih rješenja primjenjivanih prilikom izrade vlastitih rješenja ovog problema više

krajnjih točaka, je primjena koncepata *brze i spore sinkronizacije* [2].

U ovoj implementaciji sinkronizacijski metapodaci moraju sadržavati ID posljednjeg uređaja s kojim je provedena sinkronizacija koji se prilikom iniciranja sinkronizacije provjerava. Ukoliko je posljednja sinkronizacija provedena s istim uređajem, onda se provodi *brza sinkronizacija* u kojoj se sinkroniziraju promjene nastale nakon posljednje sinkronizacije. U drugom slučaju, ukoliko sinkronizacija nije provedena s istim uređajem, potrebno je provesti *sporu sinkronizaciju* u kojoj je potrebno pronaći sve promjene koje su nastale nakon posljednje sinkronizacije između ova dva uređaja i njih sinkronizirati. U ovom scenariju, sinkronizacijska logika je znatno složenija i mora uzeti u obzir način rješavanja različitih konflikata.

Ipak, preporučljivo je graditi vlastite sinkronizacijske mehanizme isključivo u slučajevima kada postojeća profesionalna rješenja ne zadovoljavaju po pitanju funkcionalnosti ili isplativosti.

4.2 Microsoft sinkronizacijski mehanizmi

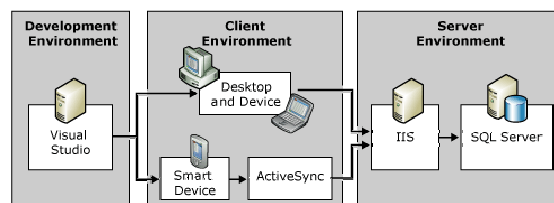
Microsoft je desetak godina prisutan na tržištu s rješenjima za mobilne baze podataka. Počevši od *SQL Servera za Windows CE*, preko *SQL Server Mobile*, Microsoft je izdao *Microsoft SQL Server Compact* izdanje koje je besplatno za korištenje i distribuciju u mobilnim aplikacijama.

SQL Server Compact je mala, u proces ugrađena, samostalna baza podataka koja podržava širok podskup standardnih SQL tipova i naredbi. Između ostalih može ju se koristiti u mobilnim i ugrađenim uređajima i aplikacijama, a iz perspektive sinkronizacije i korištenja podataka sa standardnim uređajima nudi tri različite implementacijske mogućnosti:

- Merge Replication
- Remote Data Access
- Sync Services za ADO.Net

Merge Replication i Sync Services za ADO.Net su implementacije koje se koriste za sinkronizaciju podataka [3]. Merge Replication sinkronizacijski mehanizam je dizajniran i orijentiran za uporabu u poslovnim aplikacijama, dok je Sync Service sinkronizacija orijentirana prema programerima, razvoju manjih aplikacija i drugim neposlovnim rješenjima, ali se na razvijenom Sync Service okviru (eng. framework) može implementirati skoro sve što podržava Merge Replication mehanizam [3].

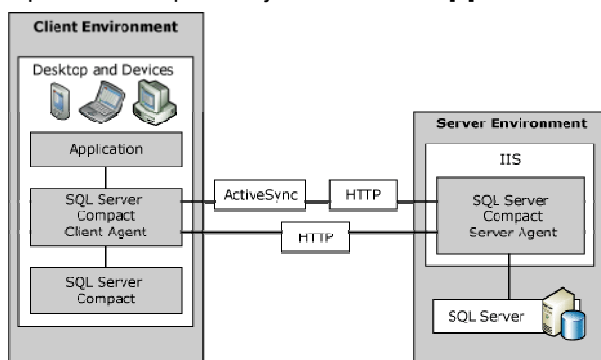
Arhitektura za razvoj aplikacija se tipično sastoji od tri okruženja koja implementiraju nekoliko bitnih alata i komponenti [7].



Slika 1. Microsoft arhitektura

Na razvojnoj strani se tipično koristi Visual Studio. Iako je MS SQL SCE tipično građen za implementaciju u aplikacijama razvijanim u Microsoft razvojnom okruženju, ono, iako preporučeno, nije eksplicitno nužno. Klijenti u MS okruženju su standardni uređaji i aplikacije te mobilni uređaji i aplikacije. Važna komponenta koja pruža mogućnost spajanja mobilnih uređaja na Microsoft operacijskim sustavima je ActiveSync. Za kraj, okruženje

na serverskoj strani je tipično upogonjeno web serverom i serverom za upravljanje bazom podataka. Replikacija i sinkronizacija podataka je kod Microsofta implementirana prema sljedećem modelu [7]:



Slika 2. Microsoft sinkronizacijska infrastruktura

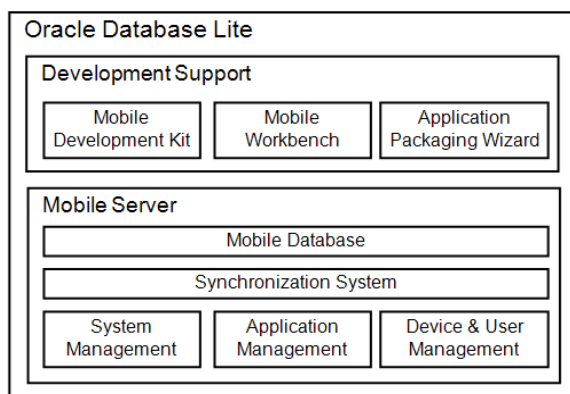
SQL Server Compact Client Agent je osnovna komponenta koja pruža mogućnost spajanja i sinkronizacije. Client Agent implementira tri objekta: replikacijski objekt (eng. Replication object), objekt za dohvat udaljenih podataka (eng. RemoteDataAccess object – RDA object) i *engine* objekt (eng. Engine object) [7]. Koristeći spomenute objekte moguće je programski kontrolirati vezu prema SQL serveru.

S druge strane, SQL Server Compact Server Agent upravlja HTTP zahtjevima pristiglim od agenta sa strane klijenta. Server agent dohvaća podatke sa SQL Servera te ih zajedno sa podatkovnom shemom vraća klijentu putem HTTP-a.

IIS web server pruža integrirane mogućnosti web servera, te pruža mogućnosti spajanja i prijenosa podataka prilikom korištenja RDA ili replikacijskih tehnologija.

4.3 Oracle sinkronizacijski mehanizmi

Kao i drugi proizvođači baza podataka i Oracle nudi rješenje mobilne baze za razvoj mobilnih aplikacija pod nazivom *Oracle Database Lite*, koje ima ugrađenu replikacijsku funkcionalnost sa *Oracle Database EE* bazom podataka [10].

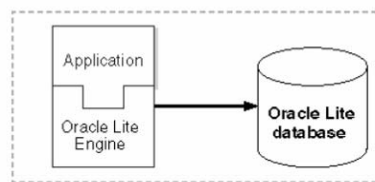


Slika 3. Oracle Database Lite komponente

Kako je prikazano na slici iznad, dva su osnovna skupa komponenti od kojih se sastoji Oracle Database Lite. U prvom redu tu su alati namijenjeni razvoju aplikacija, dok je druga komponenta Mobilni Server koji služi skalabilnom razmještanju (eng. deployment) i upravljanju aplikacija, korisnika i uređaja [11].

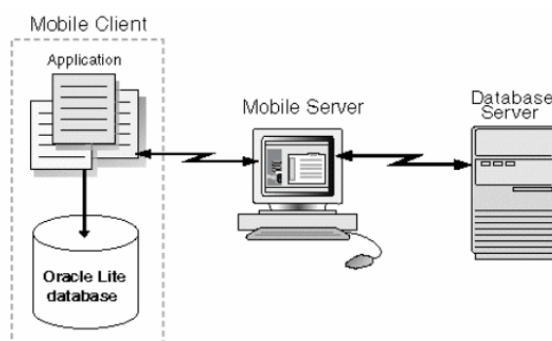
Aplikacijski proces na mobilnom uređaju uključuje samu aplikaciju, pokreće osnovne klase za rad s Oracle Lite bazom (eng. engine) i ujedno manipulira podacima u

Oracle Lite bazi. Prednost ovakvog ugrađenog pristupa je mogućnost upravljanja bazom, logikom i aplikacijom s jednog mjesta.



Slika 4. Aplikacijski proces na mobilnom uređaju

Današnje Oracle rješenje sinkronizacije [9] se temelji na infrastrukturi koja za sinkronizaciju koristi web server (Mobile Server) i repozitorij (Mobile Repository) koji se nalazi na serveru baze podataka (Database Server) i koji služi za povezivanje mobilnih i glavnih podataka. Spomenuta infrastruktura se naziva *Publication/Subscription* infrastruktura i prikazana je na slici ispod.

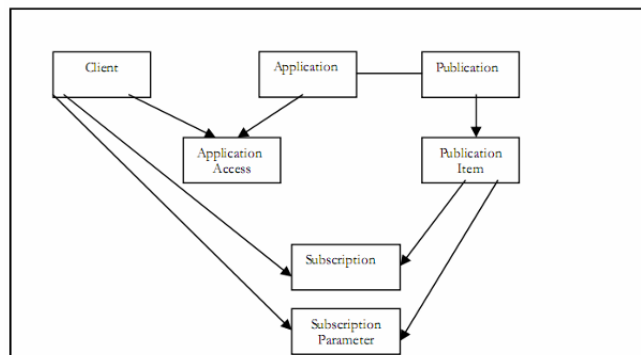


Slika 5. Oracle sinkronizacijska infrastruktura

Prilikom prve sinkronizacije, kopije tablica s glavne baze se automatski prenose na mobilnu bazu i spremaju se u obliku *snapshot* tablica kojima je dodana mogućnost praćenja promjena. Tijekom budućih sinkronizacija, ove promjene se u obliku log datoteka šalju na server (Mobile Server) i potom se pohranjuju u bazu podataka.

Sinkronizacija i očuvanje konzistentnosti podataka su dakle implementirani slanjem transakcijskih log datoteka s mobilnog klijenta. Nakon što je server integrirao promjene nastale na klijentu, ažurni podaci se šalju natrag na klijenta. Server poništava sve transakcije nastale na klijentu koje su bile u konfliktu s transakcijama nastalim na glavnom serveru, pri čemu se ne koriste nikakvi specifični algoritmi za rješavanje konflikata [2].

Replikacijska shema uključuje osnovne tablice i administratorske tablice, a osnovni replikacijski objekti su prikazani na sljedećoj slici [10].



Slika 6. Osnovni replikacijski objekti

Ovakav način sinkronizacije se može pokazati nepraktičnim, na primjer kod sinkronizacije višestrukih promjena nastalih na istom zapisu. Ipak, važna mogućnost i prednost spomenute mobilne infrastrukture je mogućnost manipuliranja postavkama i shemom baze na mobilnim uređajima s centraliziranog administrativnog računala.

5. USPOREDNA ANALIZA

Postoji više desetaka različitih proizvođača mobilnih baza podataka, no na današnjem tržištu ih stabilno pozicioniralo samo desetak. Prema nekim izvorima najveći tržišni udio ima *Sybase SQL Anywhere* baza podataka, no međutim istaknutu ulogu u tržišnoj utrci još drže IBM-ova *DB2 Everyplace* baza, spomenute *Microsoft SQL Server Compact* i *Oracle Database Lite*. Od manjih proizvođača treba izdvojiti *SQLBase* (Gupta Technologies LLC), *HandBase* (DDH Software Inc.) i *Database Viewer Plus* (Cellica Corporation NY). Naravno za kraj, tu je i *SQLite* baza podataka koja je sve češće korištena u razvoju mobilnih aplikacija.

Implementirane mogućnosti sinkronizacije i rada s mobilnim uređajima kod različitih proizvođača baza podataka	Microsoft DB2	Microsoft SQL Server Compact	SQL Anywhere	Oracle DB Lite	SQLite
Besplatno rješenje	+	+	-	-	+
Neovisno rješenje o okruženju, bez instalacije	-	-	-	-	+
Auto sinkronizacija u smjeru MSUBP	+	+	+	+	-
Auto sinkronizacija u smjeru SUBP	+	+	+	+	-
Auto dvosmjerna sinkronizacija između M- i SUBP	+	+	+	+	-
Auto sinkronizacija između više MSUBP	-	-	+/	-	-
Moguća implementacija vlastitih sinkronizacijskih mehanizama	+	+	+/	+	+
Korištenje komercijalnih rješenja sinkronizacije	+	+	+	+	+
Centralizirano administriranje sinkronizacije	-	+	+	+	-
Podaci na mobilnoj bazi mogu biti osigurani	+	+	+	+	+
Auto sinkronizacija može biti pokrenuta ručno	+	+	+	+	-
Auto sinkronizacija konfigurirana pravilima	+	+	+	+	-
Moguća direktna sinkronizacija (bez posrednika) sa sustavom za upravljanje bazom podataka	-	-	+/	-	-
Napredni algoritmi za rješavanje konflikta	-	+/	-	-	-
Konfigurabilna pravila za rješavanje konflikta	+	+	+	+	-
Podrška JDBC razvojnom sučelju	+	+	+	+	+
Podrška ODBC razvojnom sučelju	+	+	+	+	+
Podrška ADO.Net razvojnom sučelju	+	+	+	+	+
Mogućnost samostalnog oporavka od pogreške	+	+	+	+	-
Podrška pohranjenim procedurama i trigerima	-	+	+	+	-
Implementacija neovisna o platformi	-	+	+	+	+

Tablica 2. Usporedba mogućnosti sinkronizacije

Uzimajući u obzir navedene činjenice o opisanim bazama podataka, te razmatrajući mogućnosti drugih spomenutih baza podataka za mobilne uređaje [1],[5],[6],[8],[15] napravili smo usporednu analizu

Literatura:

- Chong R., Hakes I., Ahuja R.: *Getting Started With DB2 Express-C*, 3. izdanje, IBM Corp., 2009.
- Denny M., Franklin M. J.: *Edison: Database-Supported Synchronization for PDAs; Distributed and Parallel Databases*, 15. pp. 95-116, 2004.

implementiranih mogućnosti. Definirali smo skup kriterija koji se odnose na mogućnosti sinkronizacije i rada s mobilnim uređajima i usporedili ih na *Microsoft SQL Server Compact*, *IBM DB2 Everyplace*, *Sybase SQL Anywhere*, *Oracle Database Lite* i *SQLite* mobilnim bazama. Rezultati su prikazani u obliku tablice.

Prikazan tablica nas dovodi do zaključka da na tržištu nema znatne diversifikacije mogućnosti koje pružaju najveći proizvođači baza za mobilne uređaje. Prema svim ključnim kriterijima, spomenuti proizvođači nude slična rješenja koja se samo u nijansama razlikuju.

Ono što treba primijetiti jesu činjenice da se sva komercijalna rješenja temelje na komponentama koje je potrebno instalirati, da direktna sinkronizacija (bez posredničkih servera, servisa i agenata) nije moguća, isto kao što nije moguća direktna sinkronizacija između više mobilnih aplikacija. Isto tako, kod svih proizvođača još uvijek ima mjesta za poboljšanje funkcionalnosti po pitanju naprednih koncepata za rješavanje sinkronizacijskih konflikata.

Od pozitivnih elemenata treba izdvojiti podršku jednosmjernoj i dvosmjernoj automatiziranoj sinkronizaciji, podršku za primjenu kod različitih razvojnih sučelja, dobru sigurnost pohranjenih podataka i podataka u razmjeni, te rješavanja osnovnih konflikata i oporavka od pogreške.

Mobilna baza koja se prema analizi najviše izdvaja je *SQLite* baza, koja još uvijek po pitanju sinkronizacije nije konkurentna spomenutim bazama velikih proizvođača. Ipak, njena prednost je jednostavnost, brzina i otvorenost, te se nameće kao lider u rješenjima implementiranja vlastitih sinkronizacijskih mehanizama.

6. ZAKLJUČAK

Ovaj rad je pokušao prikazati cjelokupnu sliku rješavanja problema sinkronizacije podataka između mobilnih uređaja ili između mobilnih uređaja i standardnih sustava za upravljanje bazom podataka.

Problem sinkronizacije podataka treba shvatiti kao izazov kojem treba posvetiti pozornost te ga uzeti u obzir od početnih faza konceptualnog modeliranja programskog proizvoda, preko definiranja arhitekture i dizajna sustava sve do same implementacije.

Iako na tržištu postoji relativno puno proizvođača koji nude različita rješenja za primjenu kod razvoja mobilnih aplikacija, naša analiza je pokazala da su ona relativno slična, te da se samo u manjoj mjeri razlikuju. To rezultira činjenicom da postoje poslovni scenariji korištenja mobilnih aplikacija koji također trebaju ažurne poslovne podatke, a korištenjem spomenutih mehanizama sinkronizacije ih nije moguće realizirati. U konačnici, za takve poslovne scenarije je potrebno pisati vlastite sinkronizacijske mehanizme.

Rad je na sistematičan način prikazao važne parametre za implementaciju i odabir najbolje sinkronizacijskog algoritma, osvrnuo se na različite pristupe sinkronizaciji podataka primjenom *Microsoft* i *Oracle* tehnologija, te u konačnici pružio tabličan prikaz mogućnosti koja pružaju na tržištu dostupna komercijalna ili besplatna rješenja.

- 3 Goteti B. U.: Merge replication vs. Sync Services for Compact, SQL Server Compact Team Blog, Microsoft Corp., 2010.
- 4 Hill S.: History of Android: First Applications, Prototypes & Other Events; Bright Hub, 2010. Izvor: <http://www.brighthub.com/mobile/google-android/articles/18260.aspx> (učitano, 08.4.2010.)
- 5 Microsoft Corp., Differences Between SQL Server Compact and SQL Server, MSDN Library, 2010. Izvor: <http://msdn.microsoft.com/en-us/library/bb896140.aspx>
- 6 Microsoft Corp., Microsoft SQL Server 2005 Compact Edition Overview, White Paper, 2005. Izvor: <http://download.microsoft.com/download/A/4/7/A47B7B0E-976D-4F49-B15D-F02ADE638EBE/SSCEOverview.doc> (učitano: 25.03.2010.)
- 7 Microsoft Corp., Overview of SQL Server Compact, MSDN Library, 2010. Izvor: <http://msdn.microsoft.com/library/ms172448.aspx>
- 8 Mullins C. S.: DB2 Everyplace: A Mobile DB2, 2002. Izvor: http://www.craigsmullins.com/dbu_1202.htm (učitano 20.3.2010.)
- 9 Oracle Corp., Adding Mobile Capability to an Enterprise Application With Oracle Database Lite, An Oracle White Paper, 2007. Izvor: <http://www.oracle.com/technology/products/lite/OLOverview.pdf> (učitano: 4.5.2010.)
- 10 Oracle Corp., Oracle Database Lite InDepth – Oracle Database Lite Replication, An Oracle White paper, 2003. Izvor: http://www.oracle.com/technology/products/lite/lite_technical_InDepth.pdf (učitano: 4.5.2010.)
- 11 Oracle Corp., Oracle Database Lite Overview, An Oracle feature overview paper, 2006. Izvor: http://www.oracle.com/technology/products/lite/lite_datasheet_10g.pdf (učitano: 4.5.2010.)
- 12 Oracle Corp., Oracle Database Lite 10gR3, An Oracle Business White Paper, 2007. Izvor: http://www.oracle.com/technology/products/lite/BWP_Lite_10gR3.pdf (učitano: 4.5.2010.)
- 13 Polsson K.: Chronology of Handheld Computers, 2010. Izvor: <http://handheldtimeline.info/>, (učitano, 15.04.2010.)
- 14 Shroff S.: History of PDA, Izvor: http://www.medindia.net/pda/pda_history.htm, (učitano, 11.04.2010.)
- 15 Sybase Inc., SQL Anywhere: Mobile Database Management Software and Embedded Data Management Systems, 2010. Izvor: <http://www.sybase.com/products/databasemanagement/sqlanywhere> (učitano: 20.03.2010.)

Podaci o autorima:**Zlatko Stapić, mag. inf.**

zlatko.stapic@foi.hr

Fakultet organizacije i informatike

Pavlinka 2

42000 Varaždin

tel: +385 42 390 853

fax: +385 42 213 413

Zlatko Stapić, mag. inf. je od 2006. godine asistent na Katedri za razvoj informacijskih sustava na Fakultetu organizacije i informatike u Varaždinu, te polaznik poslijediplomskog doktorskog studija Informacijske znanosti na istom fakultetu. Njegova nastavna aktivnost je prvenstveno usmjerena na kolegije koji se odnose na programsko inženjerstvo, analizu i razvoj programa, modeliranje poslovnih procesa i razvoj informacijskih sustava, te značajne napore ulaže u radu sa studentima, za što je 2008. godine dobio i posebno priznanje.

Iz znanstvenog i stručnog rada treba izdvojiti voditeljstvo višegodišnjeg projekta za razvoj programskog sustava za proračun potrebne topline kod stambenih i nestambenih objekata, te voditeljstvo Modula za upravljanje internacionalnim projektima u dvogodišnjem internacionalnom Tempus projektu „Enhancing Absorption Capacity of EU Programmes in Croatia“.

U spomenutom razdoblju je objavljivao znanstvene i stručne radove u području razvoja programskih proizvoda, rudarenja podataka i sigurnosti, što su mu ujedno i područja od primarnog interesa. Zlatkov detaljniji životopis, s popisom svih radova, projekata i nagrada, te drugih važnih podataka može se pronaći na njegovoj osobnoj web stranici, na <http://www.foi.hr/nastavnici/stapic.zlatko/index.html>.

Prof. dr. sc. Neven Vrček

neven.vrcek@foi.hr

Fakultet organizacije i informatike

Pavlinka 2

42000 Varaždin

tel: +385 42 390 827

fax: +385 42 213 413