UNIVERZITET SINGIDUNUM

FAKULTET ZA INFORMATIKU I RAČUNARSTVO

Baza podataka Progress
- fakturisanje u JKP BVK -

- Diplomski rad -

Mentor:

Prof. dr Mladen Veinović

Student:

Snežana Petrović

Br. indeksa: E-8/2006

Beograd, 2010.

FAKULTET ZA INFORMATIKU I RAČUNARSTVO

UNIVERZITET SINGIDUNUM

FAKULTET ZA INFORMATIKU I RAČUNARSTVO

Beograd, Zorana Đinđića 44

Kandidat: Snežana Petrović

Broj indeksa: E-8/2006

Smer: projektovanje i programiranje

Tema: Baza podataka Progress fakturisanje u JKP BVK

Zadatak:.

Datum odobrenja rada:
Beograd, ___.___._____.

MENTOR

_____

Prof. dr Mladen Veinović

DEKAN

_____

Prof. dr Mladen Veinović

Sažetak:

U ovom diplomskom radu prikazana je baza podataka Progress i diskutovan je način kako se ona koristi. U tu svrhu, kao osnovni izvor podataka, korišćeni su podaci JKP BVK.

Upotreba baze podataka Progress značajno pojednostavljuje rukovanje podacima, a samim tim i fakturisanje.

U radu se prikazuje informatička podrška i efikasno poslovanje svih delova Sektora prodaje (od registracije potrošača, pogona zamene vodomera, evidencije o vodomerima, spojevima, potrošačima, rasturu vode do fakturisanja utroška vode) povezanih u jedinstvenu bazu podataka. Sektor prodaje je povezan u jedinstven informacioni sistem sa ostalim delovima JKP BVK, što je značajno doprinelo efikasnijem poslovanju celokupnog sistema.

.

Abstract:

This undergraduate paper deals with the Progress database and shown how it is used.

For this purpose, as the main source of data, used the data JKP BVK. Using the Progress database, greatly simplifying data handling and consequently invoicing.

The intention was to show the support of information technology and efficient operations in all parts of the sector sales (from customer registration, facility replacement meter, record the water meter, connections, consumers, leading to the break up of water consumption for invoicing) connected to a single database and connection information from the Department JKP BVK other areas, which significantly contributed to the efficient operations of the entire system.

SADRŽAJ

1 Uvod ..... - 3 -

2 Baze podataka ..... - 4 -

2.1 Personalne i Serverske baze podataka ..... - 4 -

2.2 Objekti baze podataka i prikazi ..... - 4 -

2.2.1 Tabele ..... - 5 -

2.2.2 Forme ..... - 5 -

2.2.3 Upiti ..... - 5 -

2.2.4 Izveštaji ..... - 5 -

2.3 Relacione baze podataka ..... - 6 -

3 Opis problema ..... - 7 -

3.1 Formiranje cene ..... - 7 -

3.2 Fakturisanje ..... - 7 -

3.3 Polazne osnove uvođenja baze podataka Progress u fakturisanje ..... - 8 -

3.4 Fakturisanje i nova rešenja u Progress-u ..... - 8 -

3.5 Projektovanje rešenja fakturisanja ..... - 9 -

3.6 Planovi za dalji razvoj ..... - 10 -

4 Nastanak Progress Software Corporation ..... - 11 -

4.1 Progress OpenEdge instalacija i konfiguracija na UNIX ..... - 12 -

4.2 Administracija sistema tabela ..... - 12 -

4.3 Progress i Unix i njihovo mrežno okruženje ..... - 14 -

4.4 Rutiranje ..... - 14 -

4.4.1 Broj porta ..... - 15 -

4.5 Mrežni servisi ..... - 15 -

4.6 Korišćenje baze podataka Progress na Unix sistemu ..... - 16 -

4.7 Komponente NIS sistema ..... - 17 -

4.7.1 Programi koji se koriste za ažuriranje i distribuiranje NIS baze su: ..... - 18 -

4.7.2 Sistemske datoteke koje ulaze u sastav NIS baze ..... - 18 -
4.8 Standardi baze podataka Progress ..... - 19 -
4.9 Štampa ..... - 20 -
5 Modeliranje procesa ..... - 23 -
5.1 Idef 0-funkcionalno modeliranje ..... - 23 -
5.2 Stablo aktivnosti ..... - 27 -
5.3 Modeliranje idef1x metodologijom ..... - 27 -
5.4 Logical model ..... - 28 -
5.5 Phyalcal model ..... - 29 -
6 4 GL ..... - 30 -
4GL - Fourth (4th)Generation Language ..... - 30 -
6.1 Tabela ..... - 32 -
6.2 Tipovi podataka ..... - 34 -
6.3 Index ..... - 36 -
6.4 Starting the AppBuilder ..... - 38 -
6.5 Dobijanje pomoći na mreži ..... - 40 -
6.6 Izrada formi, upita i izveštaja ..... - 40 -
6.7 Trigeri u relacionim bazama podataka ..... - 44 -
6.7.1 Razvrstavanje naredbi u trigerima ..... - 45 -
6.7.2 Načini udruživanja trigera ..... - 46 -
6.7.3 Optimalni redosled definisanja trigera ..... - 46 -
6.8 Procedure ..... - 47 -
7 Izrada fakture ..... - 49 -
8 Izgled računa za vodu za potrošača: ..... - 61 -
9 Zaključak ..... - 62 -
Literatura ..... - 63 -

1 Uvod

U ovom radu opisuje se Progress OpenEdge, odnosno upotrebe ove tehnologije u JKP Beogradski vodovod i kanalizacija. Opisan je i praktičan primer koriscenja baze podataka Progress u fakturisanju.

Menadžment preduzeća je donoseći ovu odluku očekivao da pre svega poveća efikasnost i unapredi poslovanje, kao i da eliminiše uočene propuste. Progress-om se objedinjuje i upotpunjuju podaci koji su vođeni u pet fizički odvojenih baza podataka (od kojih se jedna nalazila u potpuno drugom preduzeću),praćenje celokupnog sistema vodosnadbevanja, analiziranja potrošnje i utvrđivanja postojanja gubitaka, povezanost baze podataka i aplikativnog software-a sa ostalim delom IS preduzeća, uparenost podataka na nivou preduzeća, onemogućavanje duplih evidencija vodovodnih priključaka , evidenciju o svim potrošačima na jednom mestu (bez obzira o kojoj vrsti potrošača se radi), evidenciju o svim vodomerima i mestima gde se oni trenutno nalaze,stvaranje osnova službi za registrovanje nelegalnih priključaka za efikasniji rad,intuitivniji interface* i grafički prikaz podataka, mogućnost automatskog transfera podataka na/sa ručnih terminala i mogućnost rada sa uređajima za daljinsko čitanje, povezanost sa knjigovotstvenim podatcima i pristup podatcima o naplati, povezanost sa komandno-kontrolnim centrom i uvid u evidenciju o kvarovima, povezanost sa kontakt (CALL) centrom

2 Baze podataka

Baza podataka (eng. *database*) je računarski izraz za zbirku podataka koji se odnose na određeni pojam ili koji pripadaju određenoj poslovnoj aplikaciji. Baze podataka nam pomažu da podatke organizujemo u logičke celine radi jednostavnosti pristupa i učitavanja.

Računarska baza podataka je automatizovana verzija funkcija skladištenja i pronalaženja podataka u klasičnom sistemu za skladištenje dokumenata. Računarske baze podataka skladište podatke u struktuiranom formatu koji programer definiše. Podaci se mogu čuvati u raznim oblicima, od običnog reda teksta, do složenih struktura podataka koji sadrže slike, zvučne i video zapise. Čuvanje podataka u preciznom unapred poznatom formatu omogućava sistemu za upravljanje bazom podataka (eng. *Database Management System, DBMS*) da sirove podatke pretvara u korisne informacije putem raznih oblika ulaznih zahteva, kao što su obrasci i izveštaji. Upiti izdvajaju samo tražene podatke iz tabela, dok izveštaji prikazuju podatke na ekranu ili ih šalju na štampač.

Baza podataka sadrži jednu ili više tabela (tj. logičkih grupisanja srodnih podataka). Rad sa više tabela pojednostavljuje unošenje podataka i pravljenje izveštaja tako što umanjuje broj suvišnih podataka. Aplikacija koja radi sa više obično efikasnije radi s podacima nego što bi to činila kada bi svi podaci bili u jednoj tabeli.

2.1 Personalne i Serverske baze podataka

Personalne baze podataka su projektovane za ograničen broj korisnika i izvršavaju se na personalnim računarima. One su jeftinije rešenje u slučajevima kada je neophodno koristiti bazu podataka. Personalne baze podataka su lake za upotrebu, jer za obavljanje operacija nad bazom podataka nije neophodno koristiti složene SQL upite. Personalne baze podataka obično imaju grafički korisnički interfejs koji se lako razume i koristi.

Serverske baze podataka su posebno projektovane da istovremno opslužuju veći broj korisnika i imaju osobine koje omogućavaju efikasno upravljanje velikim količinama podataka, tako što istovremeno opslužuju više korisnika. Primeri serverskih baza podataka su Microsoft SQL Server, Oracle, ..Serverske baze podataka su projektovane tako da se mogu koristiti na različitim platformama, da odgovaraju na zahteve koje postavlja veći broj korisnika baze podataka i da optimalnom brzinom obavljaju svaki posao upravljanja bazom podataka. Serverske baze podataka su namenjenje velikim kompanijama, što znači da moraju biti dostupne 24 časa dnevno. Da bi stalno bile dostupne, serverske baze podataka imaju posebne osobine, kao što su preslikavanje i dnevničenje.

2.2 Objekti baze podataka i prikazi

Objekti baze podataka su:

1. Tabele
2. Upiti

3. Obrasci
4. Izveštaji

2.2.1 Tabele

Osnovni elemenat svake baze podataka je tabela. Svaka tabela sarži kolone i redove a u njihovom preseku nalazi se polje ili ćelija u koje Access smešta karakterističan podatak. Red tabele predstavlja jedan slog koji sa svojim poljima daje grupu podataka čiji je karakter određen kolonama tabele. Definisanjem kolone, odnosno karaktera podatka koji će se slagati u koloni, polja u koloni se definišu i svrstavaju u određen tip, koji može biti: tekst, broj, datum/vreme, logičko polje, memo polje (polje u koga se može uneti tekst proizvoljne veličine), polje za unos brojeva u novčanom formatu, posebno polje u kome se automatski po započetom unosu sloga inkrementira broj (polje AutoNumber), hiperlink polje (pokazivač na adresu na Internetu), OLE objekt polje (polje sa objektom povezanim iz drugog programa) i Look Up polje (polje čiji prikazani sadržaj se nalazi u drugoj tabeli).

Pri kreiranju baze podataka veoma je bitno postaviti tip polja, jer su pre svega, organizovanost podataka i veličina same baze u direknoj vezi sa ispravno postavljenim tipom polja.

2.2.2 Forme

Svi podaci koji se unose u bazu podataka završavaju u tabeli i spremni su za obradu. Pomalo je zamorno unositi podatke direktno u tabelu, te se kreiraju posebne maske tzv. forme kako bi se lakše unosili podaci. Forma omogućava stvaranje stranice kako bi se kasnije unosili željeni podaci na najjednostavniji način. Progress povezuje formu i tabelu pa podaci koji se unesu u formu neposredno se slažu u tabelu formirajući novi slog tabele. Forma za unos podataka može sadržati slike, komandne tastere (pritiskom na njih se aktivira neka komanda,...), može sadržati podformu itd.

2.2.3 Upiti

Upit je način skrivanja informacija koje se žele videti, a da bi se informacije koje se žele prikazati videle što jasnije. U tom smislu bitno je iz mnoštva podataka izdvojiti samo one koji su neophodni ne bi li se dobila egzaktna informacija. Tako se uz pomoć upita mogu filtrirati podaci kako iz jedne tako i iz mnoštva drugih tabela u okviru baze podataka.

Pored toga moguće je i ustanoviti kriterijume po kojima će se vršiti pretraga i sortiranje podataka. Rezultat svakog upita je, praktično jedna nova tabela sa novoformiranim slogovima po željenim kriterijumima.

2.2.4 Izveštaji

Izveštaji predstavljaju elektronski oblik klasičnih papirnih obrazaca koji se mogu gledati na ekranu ili štampati na papir. Podaci prikazani u izveštaju mogu biti izvedeni iz tabela ili iz upita (filtrirani podaci). Kreiranje izveštaja je veoma slično dizajniranju formi.

2.3 Relacione baze podataka

Termin *relaciona baza podataka* dolazi od činjenice da ona koristi relaciju (odnos) umesto datoteke. Cilj relacione baze podataka je da se što brže i jednostavnije dođe do traženog podatka. šta čini jednu tabelu relacione baze podataka.

Atributi

- Svaka kolona relacije sadrži vrednost jednog atributa, entiteta ili veze koja se prikazuje relacionom (tablicom);
- Atribut ima svoje **ime** kojim se razlikuje od ostalih atributa;
- Vrednost jednog atributa su podaci **iste vrste ili tipa**;
- **Skup vrednosti** jednog atributa zovemo **domenom** atributa;
- Vrednost atributa treba biti **jednostruka i jednostavna,** ne sme da se rastavlja na delove;
- Toleriše se ponekad da vrednost atributa nije definisana ili upisana (ne ostaje);
- Broj atributa je **stepen** relacije.

n-torke

- Jedan red relacije predstavlja primerak entiteta ili beleži vezu između dva ili više primeraka entiteta;
- Primerak zovemo **n-torka**;
- U jednoj relaciji **ne mogu** biti dve jednake n-torke;
- Broj n-torki je **kordinalnost** relacije;
- Relacija ne propisuje redosled svojih n-torki ili atributa. Permutacijom kolona ili reda dobija se drugi zapis, ali **ista** relacija.

Ključ

Ključ čine kolone čija je vrednost jednobrazna. Pomoću ključa se identifikuju atributi. Moguće je da postoji više skupova atributa koji zadovoljavaju navedena svojstva, pa se tada jedan od njih proglašava **primarnim ključem.** Skup atributa koji čine primarni ključ, zovu se **primarni atributi**. Vrednost primarnog atributa ne bi ni u jednoj n-torki smela biti neupisana (nedefinisana)!

Entitet

Entiteti su ključni objekti iz realnog sveta.

- Svaki tip entiteta prikazuje se jednom relacijom;
- Atributi entiteta postaju atributi relacije;
- Pojedini primerci entiteta prikazuje se n-torkama relacije;
- Primarni ključ entiteta postaje primarni ključ relacije;
- Postojanje veza s drugim entitetima može zahtevati dodavanje još nekih atributa; Npr. entitet STUDENT postaje relacija: STUDENT (indeks, ime, adresa, pol, ...).

3 Opis problema

U JKP Beogradski vodovod i kanalizacija postoji tehničko-tehnološka i organizaciona složenost preduzeća. Izmedju sektora i Službi, kao i izmedju zaposlenih postoje vertikalne, hijerarhijske i horizontalne komunikacije, a sve u cilju dnevnog funkcionisanja sisema.

Jedna od organizacionih jedinica je i Sektor prodaje u čijem se sastavu nalazi 5 Služi: Služba prodaje RS, Služba prodaje RJ, Služba zamene vodomera, Služba kontrole potrošača i Služba čitanja vodomera.

Tema ovog diplomskog rada je jedan jako mali deo sistema JKP BVK ali izuzetno bitan za funkcionisanje i dohodovanje preduzeća – FAKTURISANJE za utrošenu vodu i usluge odvodjenja otpadnih voda (kanalizacije). U sektoru prodaje vrši se fakturisanje vode za svih 16 Beogradskih opština.

3.1 Formiranje cene

Gradonačelnik grada Beograda, na osnovu člana 27. Zakona o javnim preduzećima i obavljanju delatnosti od opšteg interesa ("Službeni glasnik RS", br. 25/00, 25/02, 107/05, 108/06 - ispravka), član 23. Zakona o komunalnim delatnostima ("Službeni glasnik RS", br. 16/97 i 42/98) i člana 52. tačka 6. Statuta grada Beograda ("Službeni list grada Beograda", broj 39/08), donosi Rešenje o davanju saglasnosti javnom komunalnom preduzeću "Beogradski vodovod i kanalizacija" na Odluku o ceni vode i odvođenju otpadnih voda. Preduzeće Beogradski vodovod i kanalizacija određuje cenu za isporučen kubni metar vode, kao i za odvođenje otpadnih voda primenom pravila za obrazovanje cene komunalnih usluga u skladu sa Zakonom o komunalnim delatnostima uz saglasnost Skupštine grada Beograda koja je osnivač komunalnog preduzeća.

3.2 Fakturisanje

Svaki potrošač ima svoj jedinstveni broj pod kojim se vodi u BVK (šifra potrošača). Za svako potrošačko mesto postoji jedinstven broj vodomera. Količina vode koju isporuči preduzeće vodovoda određuje se očitavanjem vodomera i individualnog (pojedinačnog) vodomera, na osnovu razlike stanja na brojilu između dva očitavanja. Očitavaju se samo cele jedinice mere za koju je utvrđena cena.

Potrošači imaju obavezu da ovlašćenim radnicima Beogradskog vodovoda dozvole pristup objektu i omoguće očitavanje vodomera.Ako očitavanje vodomera nije omogućeno, radnik Beogradskog vodovoda, koji je ovlašćen za očitavanje, ima obavezu da ostavi pisano obaveštenje o danu i satu naknadnog očitavanja vodomera. U slučaju da ni naknadno očitavanje nije omogućeno, količina isporučene vode obračunava se po prosečnoj dnevnoj potrošnji vode između prva dva očitavanja vodomera, po poslednjoj izvršenoj zameni vodomera. Prilikom fakturisanja Sektor prodaje je u obavezi da primenjuje zakonska akta Osnivača: Odluku o prečišćavanju i distribuciji vode, Zakon o komunalnim delatnostima, Zakon o PDV-u.

Vrsta potrošača:	Vrsta cenovnika	Vrsta čitanja
0 Direktan priključak	11 Privreda	20 Ocitan vodomer
1 Individualni vodomer	12 Beneficirani potrošač privrede	21 Na osnovu dnevnog proseka
2 Procentualni potrošač	13 Domacinstvo	22 Zamena vodomera
3 Fiksno mesecno zaduženje		

3.3 Polazne osnove uvođenja baze podataka Progress u fakturisanje

Osnovni ciljevi koji su postavljeni pred tim stručnjaka – programera koji su ovaj zadatak trebali da sprovedu bili su:

- uvođenje jedinstvenog šifarnika sistema na nivou preduzeća, pri čemu će se koristiti zvanični šifarnici Gradskog zavoda za statistiku (tamo gde postoje)
- primena modularnog koncepta*, tako da se pojedini delovi aplikacije mogu i nezavisno koristiti
- zadovoljene principa fleksibilnosti
- mogućnost praćenja celokupne mreže vodosnadbevanja i definisanje nadležnosti za unos i ažuriranje podataka na nivou podsistema i elemenata mreže
- potpuna evidencija svih mernih uređaja (vodomera) po proizvođačima, kalibrima i mestima gde se oni nalaze (u mreži, magacinu, remontu...)
- potpuna evidencija o svom potrošačima razvrstanim po vrsti priključka, sistemu fakturisanja, različitim atributima koje određuje sam korisnik
- potpuna povezanost sa ostalim delovima IS JKP BVK
- potpun uvid u to ko je, šta i kada uneo, odnosno izmenio, od podataka
- bolje informisanje i efikasan odziv na upite i reklamacije potrošača (CRM)
- razvoj WEB portala – kako za komunikaciju sa potrošačima, tako i za otvaranje odgovarajućih servisa za poslovodstvo i radnike JKP BVK
- mogućnost exporta podataka u MS Excel i njihov grafički prikaz

3.4 Fakturisanje i nova rešenja u Progress-u

Menadžment preduzeća je donoseći ovu odluku očekivao da pre svega poveća efikasnost i unapredi poslovanje, kao i da eliminiše uočene propuste.

- objedinjavanje i upotpunjavanje podataka koji su vođeni u pet fizički odvojenih baza podataka (od kojih se jedna nalazila u potpuno drugom preduzeću)
- praćenje celokupnog sistema vodosnadbevanja, analiziranja potrošnje i utvrđivanja postojanja gubitaka

- povezanost baze podataka i aplikativnog software-a sa ostalim delom IS preduzeća
- uparenost podataka na nivou preduzeća
- onemogućavanje duplih evidencija vodovodnih priključaka
- evidenciju o svim potrošačima na jednom mestu (bez obzira o kojoj vrsti potrošača se radi)
- evidenciju o svim vodomerima i mestima gde se oni trenutno nalaze
- stvaranje osnova službi za registrovanje nelegalnih priključaka za efikasniji rad
- intuitivniji interface* i grafički prikaz podataka
- mogućnost automatskog transfera podataka na/sa ručnih terminala i mogućnost rada sa uređajima za daljinsko čitanje
- povezanost sa knjigovotstvenim podatcima i pristup podatcima o naplati
- povezanost sa komandno-kontrolnim centrom i uvid u evidenciju o kvarovima
- povezanost sa kontakt (CALL) centrom

3.5 Projektovanje rešenja fakturisanja

Da bi se osigurala jednoznačnost podataka na nivou celokupne baze, bilo je neophodno uočiti grupe jedinstvenih informacija koje opisuju sve osnovne atribute elemenata informacionog sistema Sektora, pri čemu se posebno vodilo računa da se svi podatci za komunikaciju sa ”spoljnim” svetom uklope u već postojeće, zvanične šifarnike gradskih institucija. U tom cilju pristupilo se formiranju jedinstvenog šifarskog sistema celokupnog vodovoda-koji se sastojao iz dve celine:

a) šifarnika specifičnih vodovodskih pojmova

b) šifre gradskog zavoda za statistiku (GZS)

Šifarnici vodovodskih pojmova su grubo podeljeni na one koji se odnose na mrežu vodosnadbevanja (visinske zone, objekti vodosnadbevanja, reoni, vrste kvarova...), one koji se odnose na merne uređaje (vodomeri, proizviđači vodomera, mesta ugradnje, statuse...) i šifarnike koji se odnose na očitavanje mernih uređaja i raspodelu utrošene vode.

U cilju praćenja i analize potrošnje vode i postojećih gubitaka (usled kvarova na mreži i/ili bespravnih priključaka), u celokupnoj mreži, ili njenom delu, bilo je neophodno razviti novi koncept podređenih i nadređenih objekata vodosnadbevanja.

Ceo koncept je primenjen na različite vrste potrošača – bilo da se njihova potrošnja očitava direktno sa vodomera, bilo da su u pitanju procentualni ili paušalni potrošači. Za što vernije prikazivanje na ekranu, izabran je hijerarhijski prikaz objekata putem grafičke OCX kontrole tipa ”drveta”**.

Kao jedinstven podatak-"ključ" po kome će se nedvosmosleno prepoznavati svaki spoj na vodovodnoj mreži usvojen je registarski broj spoja, umesto do tada korišćene šifre spoja koja se sastojala od šesnest govorećih cifara (ulica, kućni broj, redosled u nizu...).

Na ovaj način je dobijen jednostavniji ključ i onemogućeno samovoljno uvođenje potrošača u evidenciju obzirom da je određeno jedno odeljenje koje će biti zaduženo za otvaranje novih registara.

Osnovu za očitavanje vodomera predstavlja čitački nalog umesto čitačkih knjiga. Sam nalog je zamišljen kao elektronski nalog za ručni terminal koji čitača vodi po čitačkom hodu, pri čemu se očitana stanja unose u terminal i nakon završenog čitanja elektronski se prebacuju u bazu podataka. Prilikom unosa postoji kontrola koja upozorava čitača da još jednom proveri uneto stanje, ukoliko postoji bitnije odstupanje od prethodnog stanja. Korišćenje ručnih terminala omogućava skraćenje vreme obrade podataka jer se ne zahteva unos istih od strane referenta, čime se dobija više vremena za kvalitetniju analizu potrošnje. U slučaju da ne postoje ručni terminali, moguće je odštampati nalog u kom čitač unosi samo stanje na vodomeru, čime će se značajno umanjiti mogućnost unošenja fiktivnog stanja na bazi ranijih proseka – tzv. "tipovanje".

Radni nalog za zamenu vodomera pored standardnih podataka o izdavaocu naloga, datumu izdavanja, razlogu izdavanja..., mora da sadrži i mogućnost vezivanja digitalne fotografije skinutog vodomera.

Obzirom da se naplata utrošene vode i usluga kanalizacije za domaćinstva u stambenim zgradama obavlja preko drugog preduzeća, bilo je neophodno da se napravi jasan transfer podataka, kao i uspostavljanje veće kontrole nad zaduženjima i kontroli naplate. U tom cilju napravljen je koncept po kome JKP BVK šalje sva zaduženja na nivou vodomera (u kubicima i novčanim iznosim), dok Infostan vraća podatke o naplaćenom.

Ceo programski paket je spojen sa knjigovotstvenim podatcima o naplati i sa podatcima komandno-kontrolnog centra o kvarovima na vodovodnoj mreži, kako bi referenti imali što više kvalitetnih podataka za odgovarajuće analize. Pored toga postoji i veza sa kontakt (CALL) centrom, čime je omogućeno da potrošači dobiju sve željene informacije na jednom mestu.

3.6 Planovi za dalji razvoj

U narednom periodu se planira:

- uvođenje baze podataka Oracle
- uvođene u upotrebu ručnih terminala
- uvođenje daljinskog čitanja (putem radio signala) za određeni broj najvećih potrošača
- potpuno povezivanje Sektora u jedinstvenu bazu JKP BVK
- puštanje u rad modula za CRM (reklamacije potrošača) i u okviru njega aplikacije za šaltere
- uključenje Web portala u aktivan rad

- puštanje u rad modula MIS (Menagment Information System)

4 Nastanak Progress Software Corporation

Progress Software Corporation (Nasdaq: PRGS), nekad Data Language Corporation, je američka softverska kompanija koja prodaje poslovne softverske aplikacije infrastrukture. Njezin najpoznatiji proizvod je OpenEdge ABL (ranije poznat kao Progress 4GL), koji je razvijen u ranim 1980-ih. Najpoznatija aplikacija napisana u Progress-u je MFG / PRO (od QAD Inc.) Progress je najveća softverska firma sa sedištem u Massachusettsu; I predstavništvom u Bedfordu. Kompanija se sastoji od četiri poslovne jedinice: Progress OpenEdge Division, Sonic Software, DataDirect tehnologije, i Progress Real Time Division.

Progress Software su osnovali (pod imenom Data Corporation Language) nekoliko MIT (Massachusetts Institute of technology) studenata, uključujući i Joseph W. Alsop, 1981 [1]. Kompanija je preimenovan u 1987. Progress tehnologije i proizvodi se koriste u više od 60.000 organizacija u 140 zemalja, uključujući 90% od Fortune 500. Napredak tehnologije osigurava infrastrukturu za razne aplikacije iz oblasti ERP i financijskog trgovanja, preko industrije kao i maloprodaje (Smartstores), proizvodnje, telekomunikacija, financijskih usluga, te vlade.

U oktobru 2002 Progress Software korporacija je otkupio eXcelon iz Bostona (Nasdaq: EXLN, bivši Object Design, Nasdaq: ODIS) za oko US $ 24 milijuna. Njihova baza podataka XML tehnologija je sada dio Sonic softvera i njihovih visokih performansi Object Database tehnologije a ObjectStore, je sada dio napretka Real Time Division.

4.1 Progress OpenEdge instalacija i konfiguracija na UNIX

UNIX je stabilan operativni sistem visokih performansi, pogodan za izvršavanje velikog broja različitih aplikacija. Kao takav, dostupan je već trideset godina za brojne računarske arhitekture. Većina velikih svetskih proizvodjača računara razvija sopstvenu komercijalnu varijantu UNIX operativnog sistema Sun Solaris.

UNIX je stabilan, moćan i fleksibilan operativni sistem visokih performansi pogodan za izvršavanje kriti_nih aplikacija od visoke važnosti. UNIX je čvrsto povezan sa mrežnim servisima TCP/IP protokola, čime je u potpunosti promenjena slika UNIX servera i radnog okruženja iz prošlosti. Umesto servera sa klasičnim serijskim terminalima UNIX server se nalazi u mreži, pri čemu sa radnim stanicama ostvaruje vezu preko LAN/WAN mreže i TCP/IP skupa protokola. Većina velikih svetskih proizvođača računara razvija specifičnu varijantu UNIX operativnog sistema, što ukazuje na njegov kvalitet, popularnost i rasprostranjenost. Većina UNIX sistema, poput IBM AIX i Sun Solaris, je komercijalna korisnik mora da plati licencu za korišćenje, a izvorni kod nije rasploživ. To su razlozi narastajućće popularnosti Linux sistema koji zadržava većinu dobrih osobina UNIX sistema, a dodatno se odlikuje raspoloživim izvornim kodom i praktično besplatnim korišćenjem. Zbog toga danas većina proizvođača računara osim sopstvene komercijalne verzije UNIX sistema nudi i podršku za Linux. Linux se najčešće koristi kao operativni sistem na radnim stanicama ili serverima u manjoj ili srednjoj klasi servera, a jedna od oblasti dominantne primene, u kojoj veliki broj korisnika podržava i promoviše Linux kao bazični server, su Internet servisi.

Na UNIX sistemima postoje dve vrste korisnika:

- sistemski korisnici, koji nastaju prilikom instalacije operativnog sistema i služe za specijalne namene, a ne za prijavljivanje na sistem. Jedini sistemski korisnik koji se može prijaviti na sistem je superuser root. Root ima sve privilegije i služi isključivo za administraciju sistema;
- regularni korisnici, koji služe za prijavljivanje na sistem. Regularne korisnike kreira superuser.

4.2 Administracija sistema tabela

Prilikom instalacije ili nadogradnje operativnog sistema potrebno je utrošiti izvesnu Količinu vremena na administraciju diskova i sistema datoteka. Sistemi datoteka su neophodni za instalaciju operativnog sistema i čuvanje datoteka na sekundarnim memorijama. Postupci koji su ovde objašnjeni najčešće se izvode pre instalacije operativnog sistema, ali se mogu ponoviti ukoliko se ukaže potreba za podešavanjem parametara sistema datoteka ili ukoliko se na sistem dodaje novi disk.

četiri osnovna zadatka iz oblasti administracije diskova su:

- formatiranje diska na niskom nivou, čime se disk priprema za korišćenje. Većina diskova koji se danas proizvode fabrički su preformatirani, tako da je administrator sistema oslobođen tog dela posla;

- podela diska na particije. Disk se deli na particije ukoliko je na računar potrebno instalirati više operativnih sistema ili u cilju razdvajanja sistemskih i korisničkih datoteka, čime se pojednostavljuju postupci backupa i arhiviranja;
- kreiranje sistema datoteka na particijama diska. Korisnici svoje podatke čuvaju kao datoteke koje sa hijerarhijskom strukturom direktorijuma u kojima se nalaze čine sistem datoteka;
- aktiviranje sistema datoteka, montiranjem (mounting) na odgovarajuće direktorijume, čime se formira struktura aktivnog direktorijumskog stabla. Ovaj postupak se obavlja ili automatski, prilikom podizanja sistema, ili ručno.

Progress programi rade sa sistemima datoteka a mali broj programa sa particijama – ti programi neće raditi ukoliko sistem datoteka nije kreiran. Kada se podaci snimaju na disk, ne snimaju se direktno na particije, već strogo u sisteme datoteka. S druge strane, programi za rad s particijama (kao što su fdisk i programi za kreiranje sistema datoteka - npr. mkfs) su destruktivni po pitanju podataka koji na particijama postoje. Na primer, brisanjem jedne particije brišu se svi sistemi datoteka koji se na njoj nalaze, a samim tim i podaci na njima.

Da bi disk ili particija mogli da se iskoriste za skladištenje podataka potrebno je na njima kreirati sisteme datoteka. Sistem datoteka predstavlja način organizacije datoteka na sekundarnim memorijskim medijumima, tj. sistem datoteka je skup metoda i struktura podataka koje operativni sistem koristi za čuvanje datoteka. Sistem datoteka čine: zaglavlje (u kome se nalazi najmanje podataka, ali su ti podaci neophodni za funkcionisanje sistema datoteka), strukture za organizaciju podataka na medijumu (metadata area) i sami podaci, odnosno datoteke i direktorijumi. Zaglavlje i meta-data area čine premašenje (overhead) sistema, ali bez njih sistem datoteka ne može da funkcioniše.

Sa korisničke tačke gledišta, zahvaljujući VFS-u, UNIX sve sisteme datoteka posmatra na isti način, bez obzira da li su istog ili različitog tipa, da li se nalaze na lokalnom disku ra_unara ili na mreži. UNIX posmatra svaki sistem datoteka kao nezavisnu hijerarhijsku strukturu objekata (direktorijuma i datoteka) na čijem se vrhu nalazi root direktorijum (/).

Objektima se pristupa pomoću relativne ili apsolutne putanje i imena objekta. U objekte UNIX sistema datoteka spadaju:

Po FHS standardu aktivno stablo čine sledeći sistemi datoteka:

- (root), čiji se koreni direktorijum poklapa sa korenim direktorijumom aktivnog UNIX stabla. Root sistem datoteka sadrži osnovnu direktorijumsku strukturu stabla i sve datoteke neophodne za podizanje sistema i dovođenje sistema u stanje u kom ostali sistemi datoteka mogu biti aktivirani;
- /usr, u kome se nalazi većina korisničkih komandi, biblioteke, dokumentacija (man pages) i ostale relativno nepromenljive datoteke neophodne za normalno funkcionisanje sistema;
- /var, u kome se nalaze promenljive datoteke, poput spool direktorijuma (npr. red za štampu), log datoteka i nekih privremenih datoteka;
- /home, u kome se nalaze korisnički podaci odnosno lični direktorijumi svih korisnika sistema. Realizacijom ovog direktorijuma u vidu posebnog sistema datoteka olakšavaju se postupci arhiviranja podataka.

4.3 Progress i Unix i njihovo mrežno okruženje

Računarska mreža je skup računara relativno visokog stepena autonomije. Korisnicima umreženih ra;unara dostupni su deljeni mrežni resursi poput štampača i direktorijuma, kao i mehanizmi centralizovane autentifikacije i administracije. UNIX je mrežni (networking) operativni sistem - integralni deo sistema _ini TCP/IP skup protokola. Softver za umrežavanje, čiji su razvoj započeli programeri sa Berkeley univerziteta u Kaliforniji (grupa koja je realizovala BSD UNIX), omogućava funkcionalnost operativnog sistema u LAN i WAN mrežama. Na savremenim varijantama UNIX i Linux sistema prisutan je relativno potpun skup mrežnih alata, koji su dostupni regularnim korisnicima sistema i sistem administratorima. Većina osnovnih servisa Linux sistema, poput sistema datoteka, štampanja i arhiviranja podataka može se realizovati pomoću mrežnih funkcija operativnog sistema. U ovom poglavlju objašnjene su osnove umrežavanja, TCP/IP protokola i postupaka za korišćenje i administraciju mrežnih servisa.

Računarska mreža omogućava komunikaciju međusobno povezanih autonomnih računara. Da bi se formirala računarska mreža potrebna su najmanje dva uređaja (radne stanice, štampa_i ili serveri) povezana bakarnim ili optičkim kablovima, ili bežičnom vezom, u cilju razmene informacija i/ili deljenja resursa. U resurse koji se mogu deliti ubrajaju se:

- izlazni uređaji (štampa_i, ploteri, ...),
- ulazni uređaji (skeneri, digitalne kamere, ...),
- masovni memorijski medijumi,
- modemi i internet konekcije,
- podaci i aplikacije, u cilju efikasnog iskorišćenja prostora na diskovima i bolje saradnje na višekorisničkim projektima

4.4 Rutiranje

Prava upotrebna vrednost IP mreže iskazuje se njenom mogućnošću da kontaktira druge IP mreže. Da bi se paket poslao hostu koji se nalazi na drugoj mreži, potrebno je da u mreži postoji uređaj koji zna kako i gde isporučiti paket. Ovaj oblik isporuke paketa je poznat kao rutiranje. U IP svaka mreža čuva informacije samo o prvim skokovima ka nodovima koji predstavljaju vezu sa ostatkom sveta. Ovi nodovi znaju sledeći skok paketa, tako da se paket prenosi do odredišta putem skokova u mreži (hop).

Postoji nekoliko tipova rutiranja:

- standardno rutiranje - svi paketi koji nisu namenjeni mreži šalju se na podrazumevani izlaz. Ovaj tip rutiranja je primenljiv ukoliko mreža ima samo jedan izlaz;
- statičko rutiranje - pomoću određene komande formiraju se statičke rute u tabeli. Ovaj način rutiranja upotrebljava se ukoliko iz mreže postoji nekoliko izlaza ka drugim mrežama, pri čemu se jedan koristi kao podrazumevani izlaz ka svim ostalim mrežama;
- dinamičko rutiranje - sistem “osluškuje” broadcast pakete rutiranje i automatski podešava tabele. Mnogi Internet ruteri koriste ovaj metod rutiranja.

Unix može obavljati funkciju mrežnog rutera. Rutiranje obavljaju razni daemon programi kao što su routed (Route Daemon) i mrouted (Multicast Route Daemon).

4.4.1 Broj porta

Osim IP adrese protokoli koriste i broj porta, odnosno 16 bitni kvantitet koji dozvoljava više istovremenih konekcija na jednom čvoru. Svaka konekcija se odnosi na određeni port, koji se dodeljuje određenom mrežnom servisu na korišćenje. Korisnici mogu otvoriti ili zatvoriti određeni port, čime se omogućava, odnosno onemogu_ava uspostavljanje konekcije ka nekom tipu servisa. Portovi _iji je broj manji od 1024 smatraju se povlašćenim i za njihovo otvaranje potrebne su privilegije koje ima root. Korisnici se svakom ra_unaru mogu obratiti putem IP adrese. To naravno znači da korisnici koji koriste servise 150 različitih računara moraju znati 150 IP adresa. Da bi se komunikacija pojednostavila koristi se sistem dodele logi_kih imena IP adresama. Na primer, korisnici se mogu obratiti računaru čija je IP adresa 166.60.10.15 imenom nicotine, ukoliko je ime nicotine dodeljeno toj IP adresi. Linux sistemi koriste sledeće metode za razrešavanje imena računara:

- razrešavanje imena na osnovu datoteka u /etc/direktorijumu (/etc/hosts,/etc/networks),
- razrešavanje imena pomoću NIS servisa (Network Information Service).

4.5 Mrežni servisi

Mrežni servisi (daemons) prihvataju zahteve za uspostavljanje konekcije na određenom portu. Imena servisa su odre_ena datotekom /etc/hosts, koja povezuje servis sa odgovaraju_im protokolom (TCP ili UDP) i brojem porta. Funkcionisanje većine servisa je pod kontrolom wrapper daemon programa, kao što su inetd (Internet dispatch Daemon) i xinetd.

Na Linux sistemima postoji veliki broj mrežnih servisa koji pružaju različite usluge korisnicima sistema:

- otvaranje udaljene interaktivne sesije (telnet). Zahtevi za otvaranjem telnet sesije šalju se na TCP port 23, a prihvataju ih telnetd ili odgovarajući wrapper programi (inetd ili xinetd);
- transfer datoteka (ftp - file transfer protocol). Zahtevi za otvaranjem ftp sesije šalju se na TCP port 20, a prihvataju ih ftpd ili odgovarajući wrapper programi;
- zahtev za razrešavanjem imena (DNS) - šalje se na UDP port 53;
- pristup Web stranicama preko http protokola (servis obezbeđuje hppd iz Apache paketa);
- pristup elektronskoj pošti (servis obezbe_uju Sendmail i Postfix);
- pristup mrežnom sistemu datoteka (NFS);
- centralizovanu autentifikaciju (NIS).

Kao i svi ostali aspekti računarskog sistema, mreža nije imuna na greške. Kako mreža predstavlja skup većeg broja računara, otklanjanje problema vezanih za mrežu je komplikovanije od otklanjanja problema vezanih za ostale aspekte računarskog sistema.

Osnovna pitanja koja administrator treba da postavi ukoliko mreža ne funkcioniše ispravno su:

- da li je mreža ispravno konfigurisana (komandom ifconfig se utvrđuje da li su mrežni interfejsi konfigurisani i aktivirani),
- da li je rutiranje ispravno konfigurisano (utvrđuje se komandom netstat),
- da li računar može da komunicira sa ostalim čvorovima u mreži (utvrđuje se komandama ping i traceroute).

Administrator na serveru može tačno da prati

ATTEMPT beleži se neuspeli pokušaj ostvarivanja konekcije (log_on_failure);

DURATION beleži se dužina trajanja konekcije (log_on_success);

EXIT beleži se izlazni status servisa (log_on_success);

HOST beleži se IP adresa ra_unara koji želi da ostvari konekciju (log_on_failure i log_on_success);

PID beleži se PID servisa (log_on_success);

RECORD beleže se informacije o sistemu ukoliko se servis ne može

pokrenuti. Ove opcije koriste specifični servisi, kao što je finger;

USERID beleži se ID korisnika udaljenog sistema za sve multi-threaded

4.6 Korišćenje baze podataka Progress na Unix sistemu

NIS (Network Information System) je UNIX mehanizam centralizovane autentifikacije i upravljanja konfiguracionim datotekama. Ukoliko se NIS ne koristi, svaki UNIX proces koji želi da pristupi konfiguracionim datotekama poput /etc/passwd mora da otvori datoteku sa lokalnog diska i iz nje pribavi odgovarajuće informacije. To znači da sve bitne konfiguracione datoteke moraju postojati na svim računarima u mreži koji koriste UNIX, odnosno Linux sistem. Takođe, ukoliko administrator želi da kreira novog korisnika ili promeni članstvo u grupama, moraće da modifikuje sadržaj datoteke /etc/passwd, odnosno/etc/group na svim ra_unarima u mreži. NIS zamenjuje lokalne konfiguracione datoteke centralizovanom bazom podataka koja sadrži potpuno iste informacije.

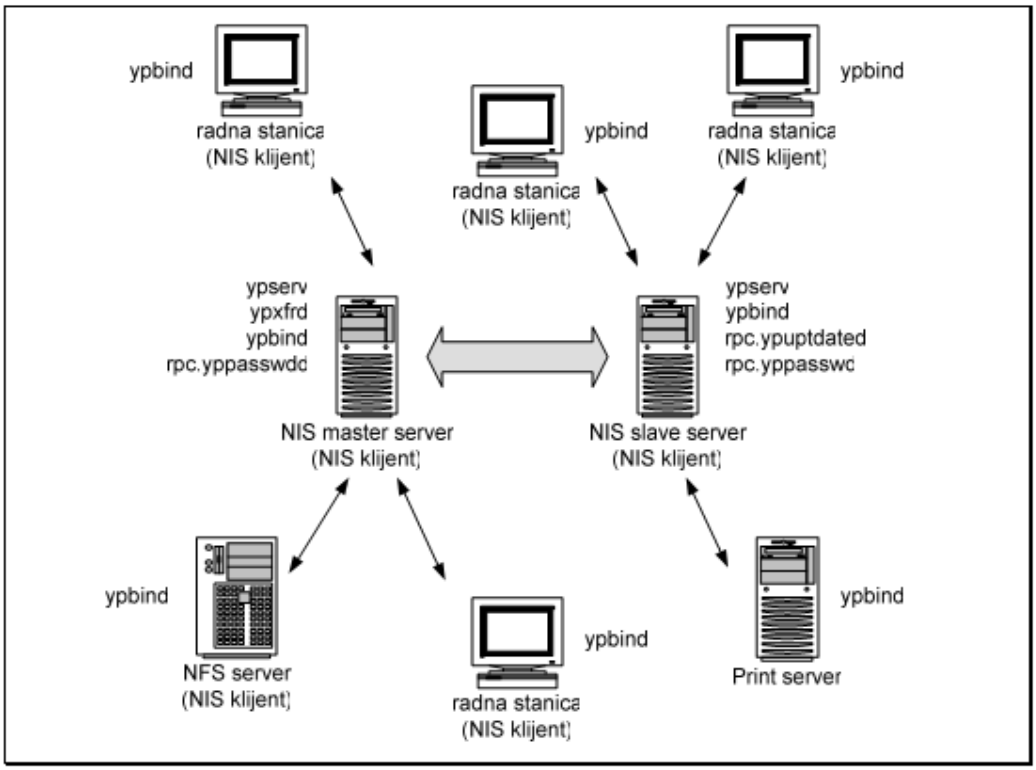
NIS grupiše računare u NIS domene. Svi računari u jednom NIS domenu koriste centralizovanu bazu pomoću koje se vrši autentifikacija korisnika i koja obezbeđuje konfiguracione datoteke za sve računare u tom domenu. U okviru jednog domena jedan računar obavlja funkciju NIS master servera. Na tom računaru se nalaze sve konfiguracione datoteke (passwd, group, hosts) od kojih se gradi baza. Dodatno, u jednom domenu postoje i slave serveri čiji se sadržaj ažurira sa master serverima. Ovi serveri odgovaraju NIS klijentima, odnosno svim ra_unarima u domenu koji vrše NIS lookup.

Prilikom podizanja operativnog sistema na klijent računaru u NIS domenu, pokreće se program ypbind (NIS lookup daemon) koji šalje broadcast upit tražeći domenski NIS

server. Svi serveri u domenu odgovaraju na upit, a sistem čiji odgovor prvi stigne postaje server datom klijentu. Server koji je opterećen odgovoriće sa zakašnjenjem, dok serveri sa lakšim opterećenjem odgovaraju brže, tako da se na ovaj način vrši raspodela opterećenja među serverima.

Korisnik klijent računara može odrediti koji računar obavlja funkciju njegovog NIS servera komandom ypwhich, dok se računar koji obavlja funkciju master servera može odrediti pomoću komande ypwhich -m.

Napomena: svaki sistem u NIS domenu, uključujući i servere, je NIS klijent. Na primer, master serveru se nakon traženja servera u domenu može dodeliti slave server. To je normalno i ne predstavlja nikakav problem



4.7 Komponente NIS sistema

NIS sistem _ini nekoliko komponenti:
.Progress baza podataka, odnosno skup DBM datoteka,

- NIS servisi (daemon),
- programi za administraciju baze.

Progress baza podataka se nalazi u /var/yp/domain direktorijumu na svakom serveru (domain je ime domena). Kreiranje NIS baze obuhvata sledeće postupke: kreiranje domena, konverziju datoteka u DMB format baze podataka, kreiranje alias tabele za

komandu sendmail, generisanje informacija za RPC sistem, povezivanje korisnika sa mrežnim grupama i uparivanje imena ra_unara sa MAC i IP adresama.

Na master serveru se tako_e nalaze i datoteke pomoću kojih se može regenerisati baza (Makefile). Kreiranje i regenerisanje NIS baze vrši se pokretanjem komande make u direktorijumu /var/yp.

NIS servisi koji obezbe_uju rad NIS mehanizma su:

- ypserv NIS server (master ili slave),
- ypbind proces kojim se klijentu dodeljuje NIS server,
- ypxfrd proces na master serveru koji upravlja prenošenjem ažuriranih
- podataka na slave server,
- rpc.ypupdated proces na slave serveru koji upravlja prenošenjem ažuriranih podataka sa master servera,
- rpc.yppasswdd rešava zahteve za promenu lozinke korisnika sa udaljenih računara.

Programi koji se koriste za administraciju NIS baze su:

- ypinit kreira novi NIS domen,
- makedbm kreira bazu, odnosno konvertuje konfiguracione datoteke u DBM format baze podataka,
- mkalias kreira alias tabelu za komandu sendmail,
- mknetid generiše informacije za RPC sistem,
- revnetgroup generiše datoteku kojom se korisnici povezuju sa mrežnim grupama,
- stdethers generiše datoteku kojom se uparuju imena računara sa MAC adresama,
- stdhosts generiše datoteku kojom se uparuju imena računara sa IP adresama.

4.7.1 Programi koji se koriste za ažuriranje i distribuiranje NIS baze su:

yppoll zahtev slave servera za ažuriranje baze sa master serverom,

yppush prosleđuje tabele sa master servera na slave servere,

ypxfr program koji vrši transfer tabela sa jednog servera na drugi,

ypset zahtev da sistem bude server procesu ypbin na klijent računaru.

4.7.2 Sistemske datoteke koje ulaze u sastav NIS baze

NIS baza podataka se nalazi u direktorijumu /var/yp/domain na serveru, gde je domain ime domena. Pre kreiranja baze administrator sistema treba da odredit koje će

konfiguracione datoteke u_i u bazu. U podrazumevanom stanju, u sastav NIS baze ulaze sledeće datoteke:

passwd svi korisnici sistema,

group sve korisničke grupe,

hosts parovi IP adresa i imena računara,

ethers parovi MAC adresa i imena računara ,

networks parovi IP adresa i imena mreža,

services imena IP portova,

protocols imena IP protokola,

netgroup pripadnost korisnika mrežnim grupama

4.8 Standardi baze podataka Progress

Bilo kakva automatska obrada informacija zahteva najmanje tri stvari: hardver koji će procesirati informacije (ra;unar), program za obradu informacija i same informacije. Da bi program mogao da se izvršava na ra_unaru i obradi informacije na onaj način na koji korisnici to očekuju, potrebno je programu obezbediti odgovarajuće "uslove za rad". Osim prostora na disku većina programa zahteva sledeće:

- informacije koje će procesirati, smeštene na tačno određenom mestu na sekundarnim memorijama (diskovi, trake), pod odgovarajućim imenom i u odgovarajućem formatu;
- jednu ili više konfiguracionih datoteka, pomoću kojih se kontroliše ponašanje programa, i donekle omogućava prilagođavanje programa korisniku. Konfiguracione datoteke se takođe moraju smestiti na diskove pod odgovarajućim imenima na odgovarajuća mesta;
- pravilno određena mesta na diskovima gde će program snimati rezultate i međurezultate obrade ulaznih informacija.

Dodatno, neki programi mogu zahtevati da na sistem budu instalirani i drugi softverski paketi, bez kojih upotreba samih programa nije moguća. Iako to nije obavezno, uz program se u najvećem broju slučajeva isporučuje i prateća dokumentacija koja omogućava korisnicima da lakše i brže nauče da koriste program.

To znači da proces instalacije programskog paketa obuvata veći broj postupaka od jednostavnog kopiranja datoteka. U opštem slučaju instalacioni proces obuhvata sledeće aktivnosti:

- proveru međuzavisnosti paketa (package dependencies), odnosno proveru postojanja prethodno instaliranih paketa koji su neophodni za izvršavanje konkretnog programa (na primer, bashburn zahteva da na sistemu bude instaliran paket cdrecord, dok Midnight Commander zahteva da na sistemu bude instaliran paket ncurses). Ti paketi se moraju instalirati pre instalacije samog programa;
- proveru slobodnog mesta u sistemima datoteka. Ukoliko na diskovima nema dovoljno prostora za smeštaj paketa, proces instalacije se prekida;
- kreiranje privremenih direktorijuma za kopiranje i raspakivanje paketa;

- kreiranje destinacionih direktorijuma za smeštaj izvršnih i konfiguracionih datoteka i dokumentacije;
- prevođenje i povezivanje izvornog koda programa;
- kreiranje ili kopiranje inicijalnih konfiguracionih datoteka;
- kopiranje dokumentacije;
- kreiranje linkova;
- brisanje privremenih datoteka i direktorijuma koje koristi instalacioni proces.

Skup aktivnosti koje obavlja instalacioni proces zavisi od formata programskog paketa. Na primer, ukoliko se program isporučuje u binarnom obliku (preveden i povezan), instalacioni proces ne uključuje korake prevođenja i povezivanja, a često ni kreiranja privremenih direktorijuma.

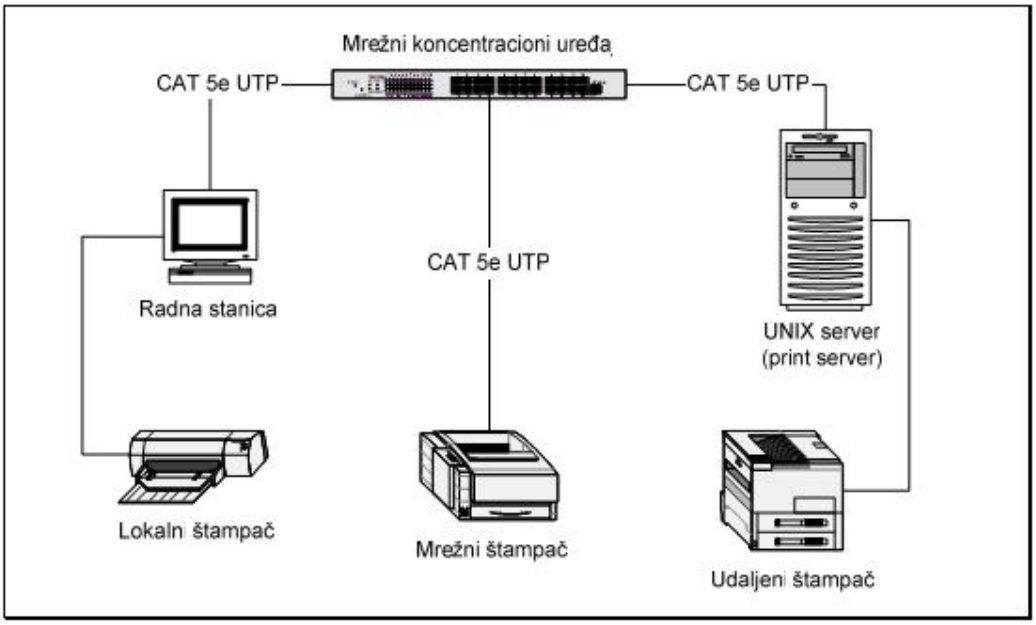
Ukoliko paket uključuje veći broj datoteka ovi koraci mogu biti veoma komplikovani, a ručno instaliranje se može izvesti samo ukoliko se prethodno pročita obimna dokumentacija. Da bi se postupak instalacije pojednostavio koriste se specijalni alati za instalaciju softverskih paketa - paket menadžeri (package managers). Ovi alati najčešće koriste specifičan format datoteke - paketa, kao što su RPM i DEB.

Svaki sistem za upravljanje paketima treba da poseduje mogućnost praćenja svakog dela softvera na operativnom sistemu, kao i da poseduje sledeće funkcije:

- instaliranje novih paketa,
- deinstaliranje (uklanjanje) starih ili neželjenih paketa,
- nadogradnja paketa novijim verzijama (update, odnosno upgrade),
- prikazivanje informacija o instaliranim paketima.

4.9 Štampa

Svi mrežni operativni sistemi, korisnicima omogu_ava štampanje dokumenata na lokalnim i udaljenim štampačima. To znači da korisnik može da koristi štampač koji je pomoću paralelnog ili USB kabla povezan na njegov računar, kao i štampače koji su povezani na drugi umreženi računar ili na neki mrežni uređaj. Da bi korisnik uopšte mogao da štampa na računaru mora biti instaliran i pokrenut odgovaraju_i servis (daemon) koji upravlja zahtevima za štampu. Za razliku od Microsoft Windows ili Mac OS operativnih sistema, UNIX nema standardni interfejs i sistem za podršku štampača, što dovodi do problema pri razvoju drajvera za štampače. Razli_ite vrste UNIX sistema koriste različite servise i skupove komandi kojima se upravlja redom za štampu, a dva najznačajnija rešenja koja se danas koriste deo su System V i BSD UNIX sistema. Ovi sistemi za štampu omogu_avaju štampanje teksta na linijskim štampačima i štampanje teksta i grafike na PostScript štampa_ima. Kasnijim proširenjima uvedena je podrška za veći broj štampača i formata datoteka koje se mogu štampati. Linux sistemi za štampu (lprNG - line printer new generation) i CUPS (common UNIX printing system) objedinjuju System V i BSD interfejse.



Kada korisnik pošalje zahtev za štampu, zahtev se najpre smešta u red za štampu. Red za štampu je specijalni direktorijum koji se nalazi na hard disku radne stanice ili servera. Zahtev čeka u redu sve dok se štampač ne oslobodi, nakon čega se štampa. Svakom štampaču se dodeljuje red za štampu na računaru na koji je fizički povezan.

Osnovne komponente UNIX okruženja za štampu su:

- štampači
- redovi za štampu
- serveri za štampu

Štampač je hardverski uređaj za stampu. Štampač je deljivi resurs - može ga koristiti jedan korisnik ili grupa korisnika. U odnosu na računar sa kog se šalje zahtev za štampu štampač može biti lokalni ili udaljeni - remote. Lokalni štampa_i su svi štampa_i povezani na paralelni ili serijski port računara sa kog se šalje zahtev za štampu. Udaljeni štampači su svi štampači povezani na drugi server ili radnu stanicu koji su dostupni sa računara sa kog se šalje zahtev za štampu. U mrežne štampače ugrađeni su NIC adapteri, odnosno mrežne kartice koje omogu_avaju direktno vezivanje štampa_a na mrežne ure_aje (Hub ili Switch), pri _emu se veza naj_eš_e ostvaruje UTP kablovima kategorije 5 i 5e. Slika 7.1 ilustruje lokalne, udaljene i mrežne štampače i način na koji se oni vezuju za računare, odnosno mrežne uređaje.

Server za štampu je računar na kome se nalazi red za štampu. Na serveru za štampu aktiviran je print servis (daemon) koji prihvata dolazeće zahteve, raspoređuje ih u red i šalje na štampu kad je štampač slobodan. Zahtevi za štampu nalaze se u redu sve dok se ne odštampaju, a nakon toga ih print servis briše iz reda.

Server za štampu može da bude radna stanica ili mrežni server. Štampači koji su priključeni na radnu stanicu naj_eš_e koriste isti mrežni server kao server za štampu, pri čemu se red sa svim zahtevima čuva lokalno. Štampači koji su priključeni na mrežni server (ili povezani direktno na mrežni koncentracioni uređaj) najčešće koriste isti mrežni server kao server za štampu. U tom slučaju se zahtevi upućeni sa drugih računara procesiraju

centralno. Serveri za štampu mogu da upravljaju većim brojem štampača i redova za štampu.

Sistem administratori postavljaju inicijalno okruženje za štampu tako što instaliraju štampače, definišu redove za štampu i odgovarajuću podršku u vidu servisa. Nakon toga, administrator rukovodi okolinom za štampanje: određuje dostupnost štampača, vrši kontrolu pristupa štampaču (određuje skup korisnika koji mogu da koriste štampač) i upravlja redovima za štampu.

5 Modeliranje procesa

5.1 Idef 0-funkcionalno modeliranje

Grafički jezik IDEF0 opisuje metodu funkcionalne dekompozicije preko skupa dijagrama, od kojih svaki predstavlja ograničenu količinu detalja definisanih odgovarajućom sintaksom i semantikom. Dijagrami su međusobno povezani tako da opisuju sistem, hijerarhijski, sa vrha naniže. Dijagrami se sastoje od pravougaonika koji predstavljaju neki deo celine. Povezani su međusobno usmerenim linijama koje predstavljaju veze između delova.

Dijagram konteksta za racun za vodu ili fakturu definisan je pravougaonikom koji predstavlja hipotetičke granice sistema koji se proučava. U i van ovog sistema teku informacije preko strelica. Dijagram konteksta je najviši nivo apstrakcije koji se dekompozicionim dijagramima prevodi u niži nivo apstrakcije. Na sledećoj slici prikazan je dijagram konteksta za fakturu gde se mogu videti koji tipovi informacija su potrebni a predstavljeni su kao nazivi strelica.

Definisanjem dijagrama konteksta, tj. granica sistema, uspostavljaju se okviri posmatranja i definiše okolina koja utiče na sistem.

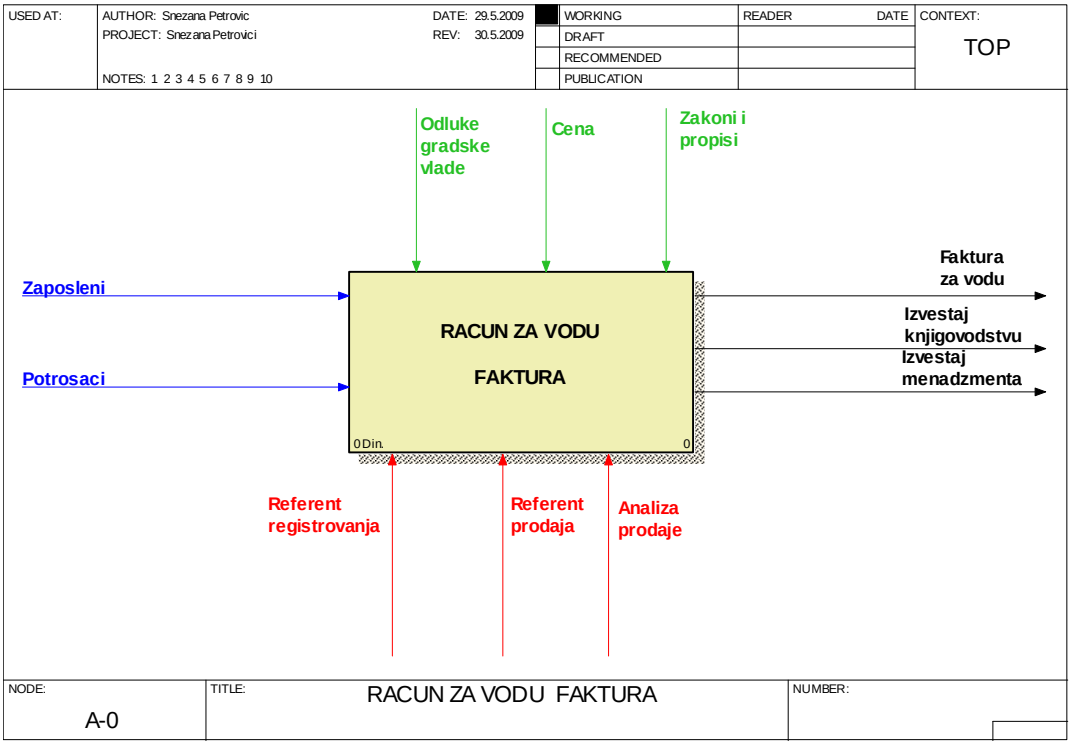
Strelice sa leve strane pravougaonika definišu se kao ulazi (Input) i definišu se kao: Potrošači, Zaposleni

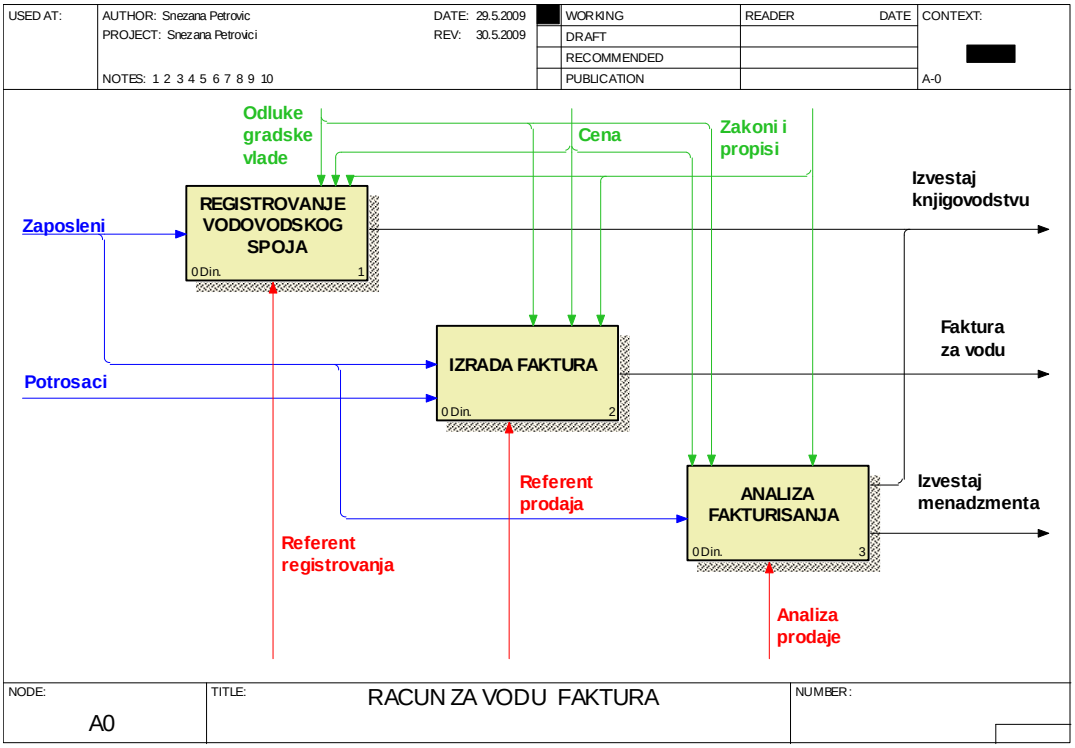
Strelice koje ulaze u pravougaonik odozgo definišu se kao kontrole (Control) i definisane su kao: Cena, Odluka gradske vlade i Zakoni i propisi.

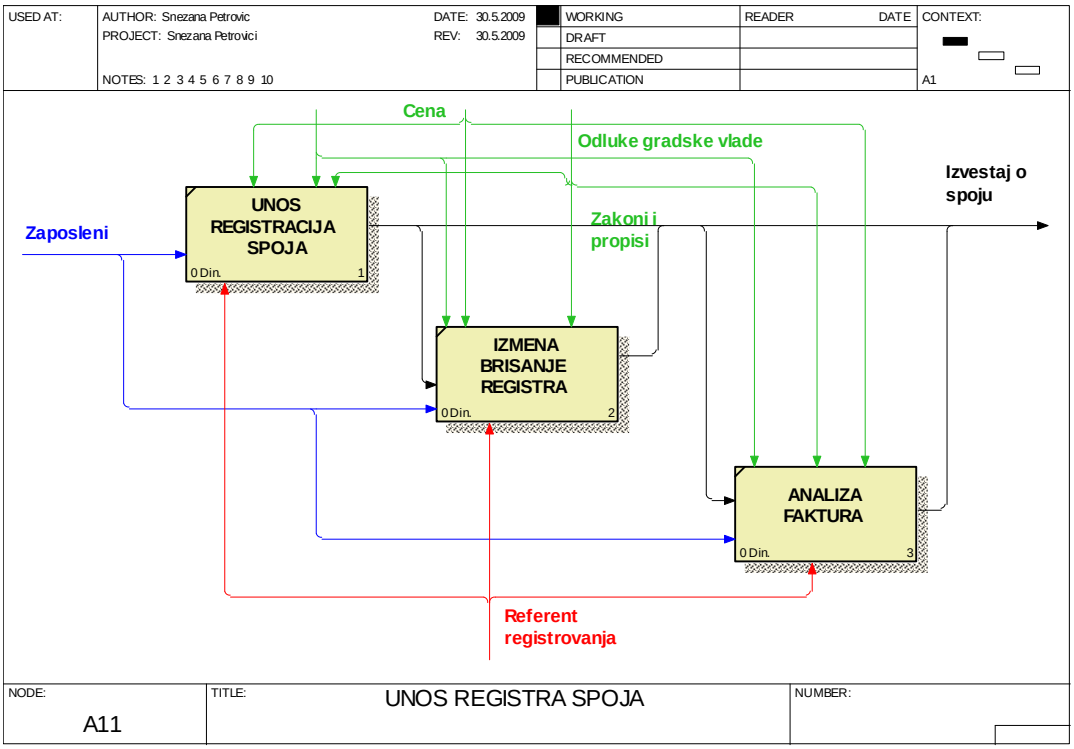
Strelice koje izlaze iz pravougaonika na desnoj strani predstavljaju izlaze (Output). Izlazi su podaci ili objekti proizvedeni od strane aktivnosti i za postavljeni proces RAČUN ZA VODU: faktura, izvestaj knjigovodstvu i izvestaj menadžmentu.

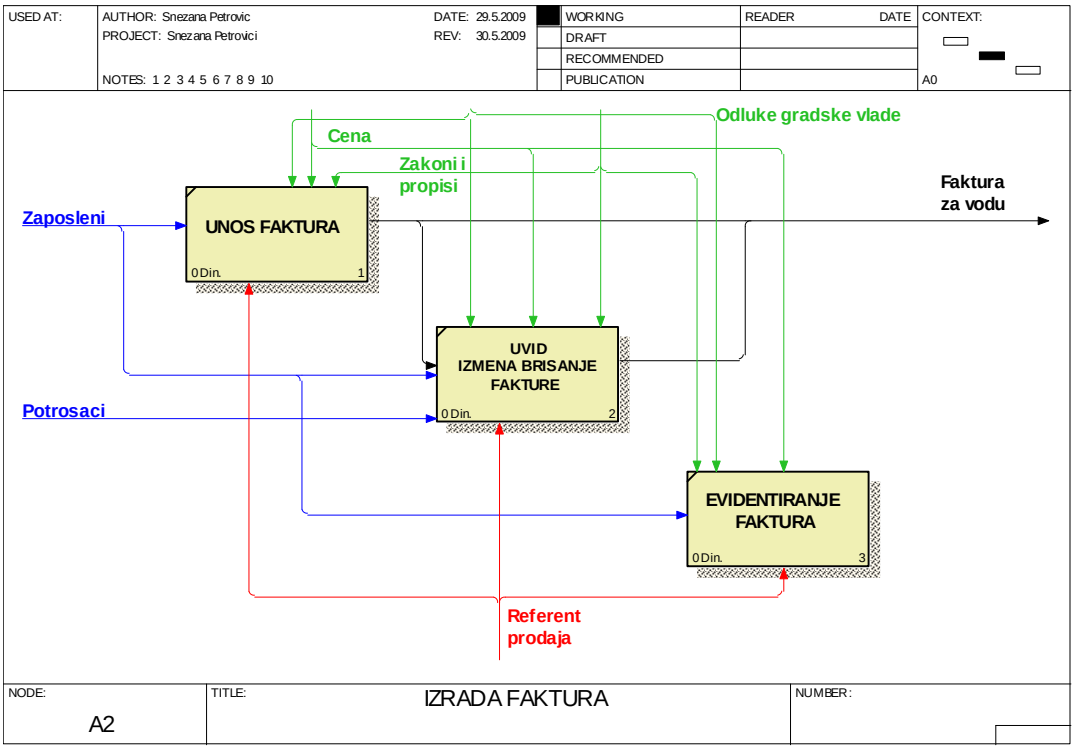
Strelice na donjoj strani pravougaonika predstavljaju mehanizme. Strelice okrenute prema gore identifikuju značenje koje podržava izvršenje aktivnosti i definišu odgovornost izvršilaca posla a to su: referent registrovanja, referenti prodaje i analitičar prodaje.

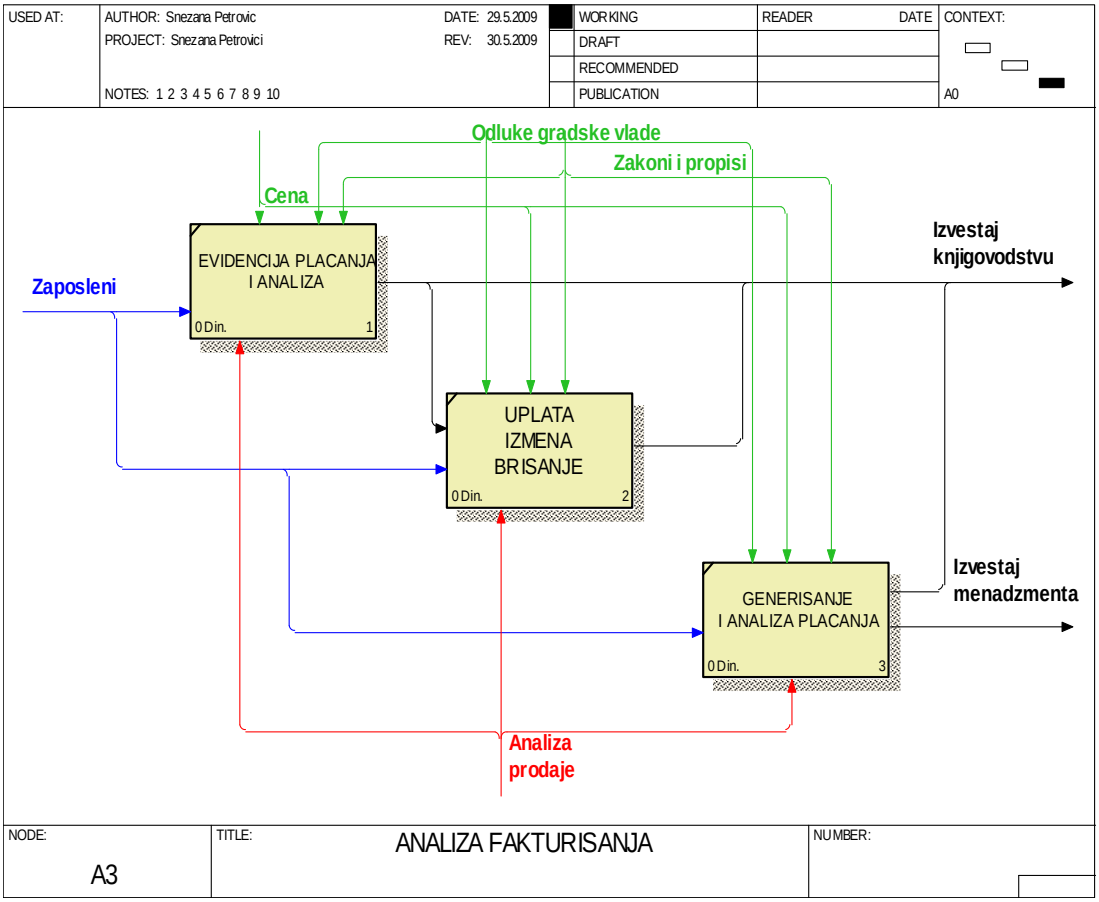
Imajući u vidu ovako postavljeni dijagram konteksta, u sledećem koraku definiše se stablo aktivnosti.


USED AT:
AUTHOR: Snezana Petrovic
PROJECT: Snezana Petrovici
NOTES: 1 2 3 4 5 6 7 8 9 10
DATE: 29.5.2009
REV: 30.5.2009
WORKING
DRAFT
RECOMMENDED
PUBLICATION
READER
DATE
CONTEXT:
TOP
Odluke gradske vlade
Cena
Zakoni i propisi
Zaposleni
Potrosaci
RACUN ZA VODU
FAKTURA
0 Din.
0
Faktura za vodu
Izvestaj knjigovodstvu
Izvestaj menadzmenta
Referent registrovanja
Referent prodaja
Analiza prodaje
NODE:
A-0
TITLE:
RACUN ZA VODU FAKTURA
NUMBER:


USED AT:
AUTHOR: Snezana Petrovic
PROJECT: Snezana Petrovici
NOTES: 1 2 3 4 5 6 7 8 9 10
DATE: 29.5.2009
REV: 30.5.2009
WORKING
DRAFT
RECOMMENDED
PUBLICATION
READER
DATE
CONTEXT:
A-0
Odluke gradske vlade
Cena
Zakoni i propisi
Zaposleni
REGISTROVANJE VODOVODSKOG SPOJA
0 Din.
1
Izvestaj knjigovodstvu
Potrosaci
IZRADA FAKTURA
0 Din.
2
Faktura za vodu
Referent prodaja
ANALIZA FAKTURISANJA
0 Din.
3
Izvestaj menadzmenta
Referent registrovanja
Analiza prodaje
NODE:
A0
TITLE:
RACUN ZA VODU FAKTURA
NUMBER:


USED AT:
AUTHOR: Snezana Petrovic
PROJECT: Snezana Petrovici
NOTES: 1 2 3 4 5 6 7 8 9 10
DATE: 30.5.2009
REV: 30.5.2009
WORKING
DRAFT
RECOMMENDED
PUBLICATION
READER
DATE
CONTEXT:
A1
Cena
Odluke gradske vlade
Izvestaj o spoju
Zaposleni
UNOS REGISTRACIJA SPOJA
0 Din.
1
Zakoni i propisi
IZMENA BRISANJE REGISTRA
0 Din.
2
ANALIZA FAKTURA
0 Din.
3
Referent registrovanja
NODE:
A11
TITLE:
UNOS REGISTRA SPOJA
NUMBER:

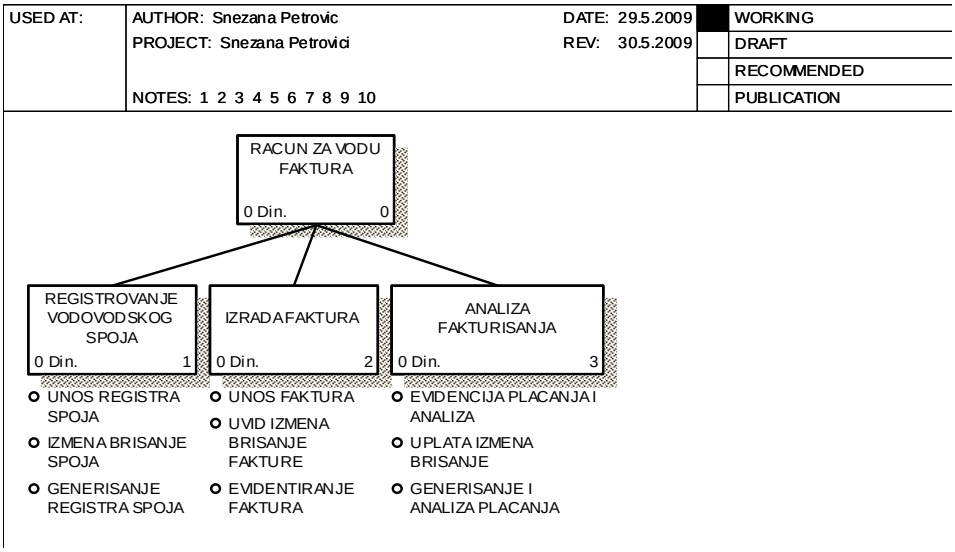

USED AT:
AUTHOR: Snezana Petrovic
PROJECT: Snezana Petrovici
NOTES: 1 2 3 4 5 6 7 8 9 10
DATE: 29.5.2009
REV: 30.5.2009
WORKING
DRAFT
RECOMMENDED
PUBLICATION
READER
DATE
CONTEXT:
A0
Odluke gradske vlade
Cena
Zakoni i propisi
Faktura za vodu
Zaposleni
UNOS FAKTURA
0 Din.
1
UVID IZMENA BRISANJE FAKTURE
0 Din.
2
Potrosaci
EVIDENTIRANJE FAKTURA
0 Din.
3
Referent prodaja
NODE:
A2
TITLE:
IZRADA FAKTURA
NUMBER:


USED AT:
AUTHOR: Snezana Petrovic
PROJECT: Snezana Petrovici
NOTES: 1 2 3 4 5 6 7 8 9 10
DATE: 29.5.2009
REV: 30.5.2009
WORKING
DRAFT
RECOMMENDED
PUBLICATION
READER
DATE
CONTEXT:
A0
Odluke gradske vlade
Zakoni i propisi
Cena
Zaposleni
EVIDENCIJA PLACANJA I ANALIZA
0 Din.
1
Izvestaj knjigovodstvu
UPLATA IZMENA BRISANJE
0 Din.
2
GENERISANJE I ANALIZA PLACANJA
0 Din.
3
Izvestaj menadzmenta
Analiza prodaje
NODE:
A3
TITLE:
ANALIZA FAKTURISANJA
NUMBER:

5.2 Stablo aktivnosti

Definisanjem stabla aktivnosti uspostavljaju se vertikalne (hijerarhijske) veze između aktivnosti. Stablo aktivnosti se definiše primenom metode rešavanja problema odozgo na dole (top-down), kada se složena aktivnost rastavlja na više podređenih aktivnosti a zatim se pristupa rešavanju jednostavnih podređenih aktivnosti.

Drugim rečima, polazna složena aktivnost razvija se u hijerarhiju podređenih aktivnosti, čija je struktura tipa stabla. Koren stabla (to je najviši čvor stabla) sadrži polaznu aktivnost, dok listovi, tj. čvorovi koji nemaju potomke, sadrže aktivnosti čije je rešavanje relativno jednostavno. Rešavanjem svih podređenih aktivnosti iz listova rešena je i polazna složena aktivnost.

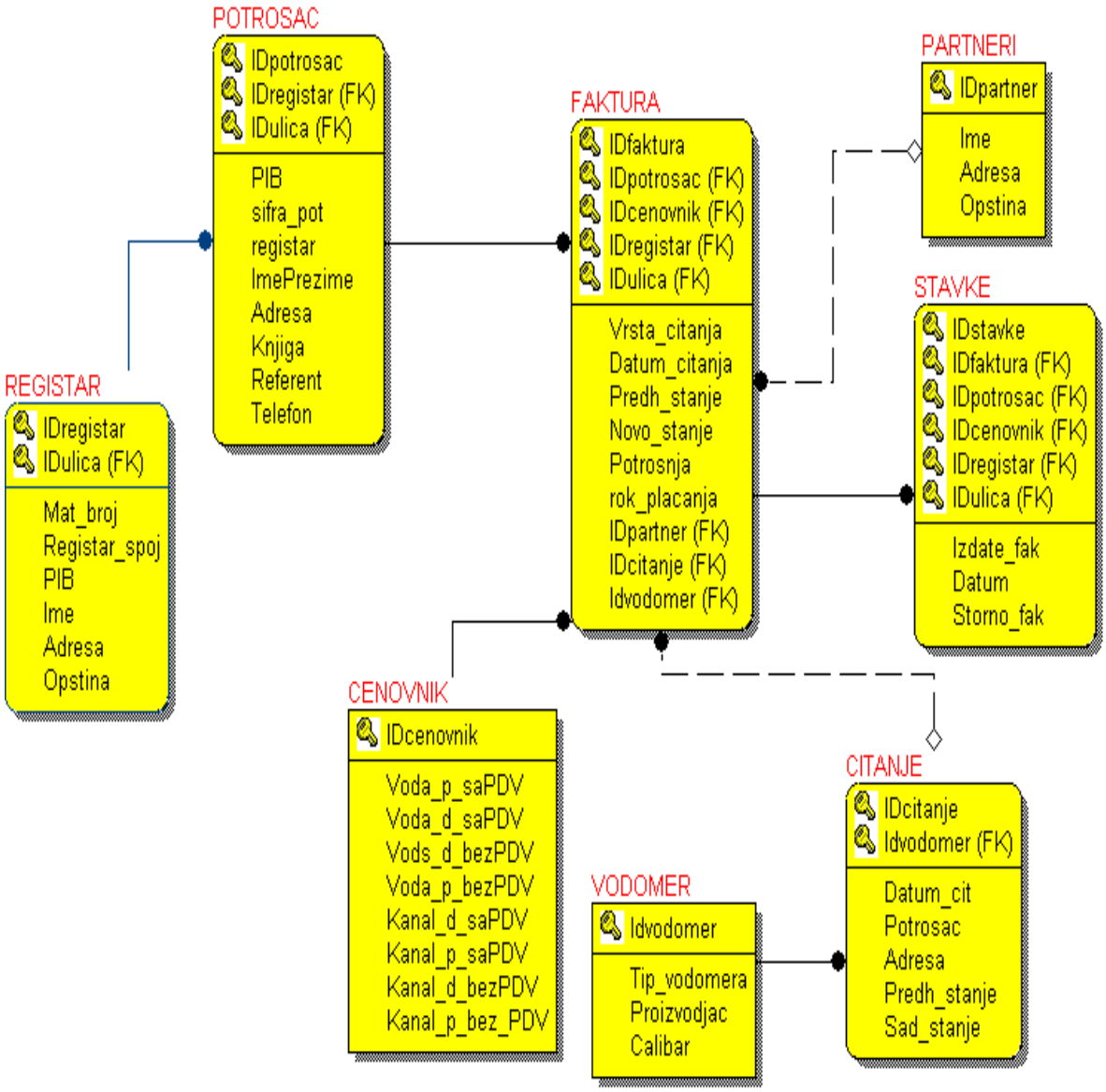


5.3 Modeliranje idef1x metodologijom

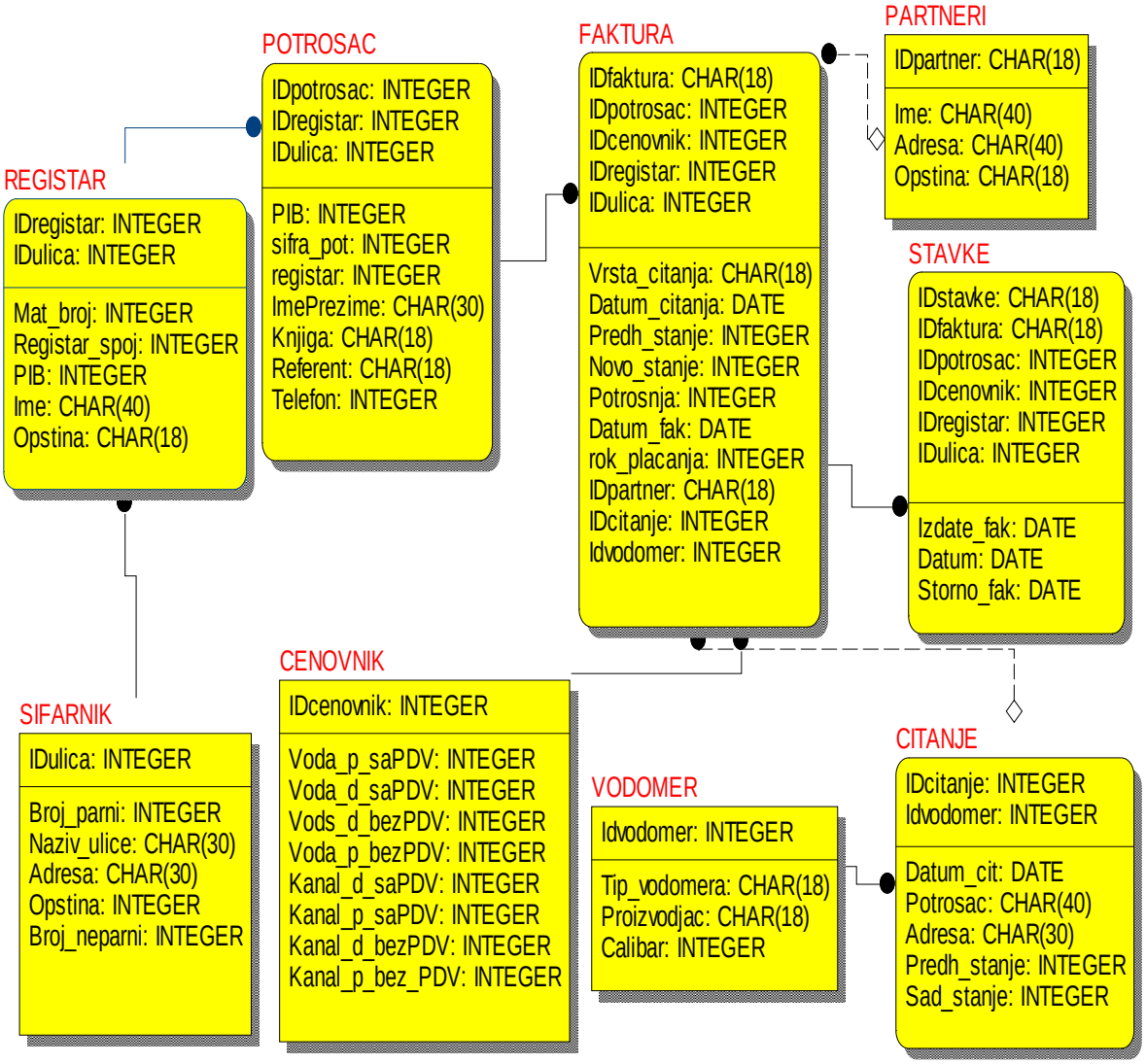
Korišćena je IDEF1X metodologija, upotrebom Case alata ERwin (Entity Relationships for windows) za informaciono modeliranje fakturisanja.

Za funkcionalnu i infromacionu analizu IS-a koristimo odgovarajući CASE alati zasnovani na IDEF0 (**I**ntegation **DEF**inition) standardu realizovanom kroz Bpwin (Bussines Process windows) CASE alat za funkcionalnu analizu IS i IE (Information Engineering) standardi realizovan kroz ERwin (Entity Relationships for windows) CASE alat za Informacionu analizu IS tj. modeliranje podataka. U CASE alatu realizovan je i standard IDEF1X koji je korišćen u osnovnom udzbeniku Modeliranje informacionih sistema za predmet Analiza informacionih sistema.

5.4 Logički model



5.5 Phyalcal model



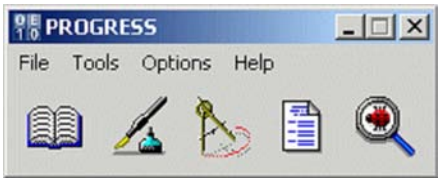
6 4 GL

4GL je proceduralni programski jezik, namenjen da omogući rad svih aspekata primene poslovanja, iz korisničkog interfejsa na pristup bazi podataka i poslovne logike. Za preko 20 godina Progress 4Gl je postao svestran i moćan alat koji ne samo da omogućava da ga koristite na programu aplikacija, već i da se izrade moćne alatke koje koriste za pomoć u stvaranju i podršku aplikacijama. Ovo poglavlje pruža uvod u Progress 4GL-a i obuhvata sledeće delove. Progress je četvrti jezik generacije 4GL, zbog moćnih izjava (deklaracija) i ključnih reči koje su specijalizovane za izgradnju poslovnih aplikacija. Pojedinačne izjave u programiranju 4GL-a mogu se uraditi na desetinama ili stotinama linija koda u standardnom 3GL, kao što je Visual Basic, Java ili C++. Jedna izjava 4GL-a može dati sve podatke iz baze podataka za korisnički interfejs, ili korisnikovu izmenu vratiti u bazu podataka. Drugi izveštaji omogućavaju da program sa velikom preciznošću čak i na nizi nivo odvajanja pojedinačnih bitova od toka podataka. Ova fleksibilnost je ono što daje Progressu veliku moć i razvoj jezika. Većina razvojnih alata koristi Open Edge aplikacije i sami su zapisani u 4GL-u. Na svom početku, u ranim 1980-tim Progressa 4GL programerima je dozvoljeno da izrade karakter interfejsa aplikacije koja je korišćena na širokom spektru hardverskih platformi, uključujući manoge sorte Unixa, DOS-a i neke druge operativne sisteme koji više nisu u upotrebi. Rane Progress aplikacije su, od prvih, potpuno prenosivih između platformi, tako da programer može jednostavno premestiti aplikaciju programa iz jednog tipa mašine ili jednog tipa ekrana terminala na drugi sa poverenjem da će oni raditi ispravno na svakom mestu. Sa povećanjem prisustva Microsoft Windowsa kao platforme za grafičke interfejse, 4GL nastavlja da podržava one interfejse, sa svim svojim različitim vizuelnim kontrolama, kao i događaje kojima upravlja programer konstruktor Danas 4Gl nastavlja da se razvija, sa novim nastavcima da se obezbedi više dinamičnih definicja primene komeponenti, kao i pristup da se otvore tehnologije kao što su HTML i KSML i niz drugi hkonstrukata da podrže otvoreni razvoj aplikacije i primenu životne sredine. I sve vreme, Progress 4GL bazne aplikacije mogu biti izvedene iz jednog izdanja do sledećeg uglavnom bez promena. Progres obezbeđuje stapen kompatibilnosti

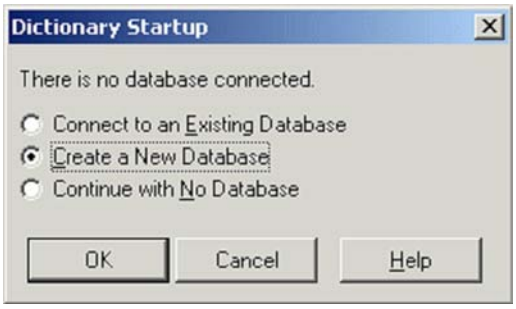
4GL - Fourth (4th)Generation Language

Zajednički izraz koji se koristi na programske jezike orjentisane više ka krajnjem korisniku nego programerskim profesionalcima. 4GL jezici obuhvataju proces kreiranja aplikacija, minimiziraju probleme otklanjanja grešaka, kreiraju ugodnu korisničku sredinu, redukuju troškove održavanja softvera, itd. Oni imaju generatore aplikacija, generatore izveštaja, jezike upita, itd. Neki primeri takvih jezika su: ADF, ADRSII, APL, AS, QBE, QMF, SQL, itd.

Iz menija Start operativnog sistema Windows radnoj površini, izaberite OpenEdge okruženje, koristeći bez obzira na ime koje je dao da ga kada ga instalirate, a pod tim stavku menija, izaberite Desktop opcija. Na primer, izaberite Start → OpenEdge → desktop. Application Development Environment (ADE) desktop pojaviće se prozor:

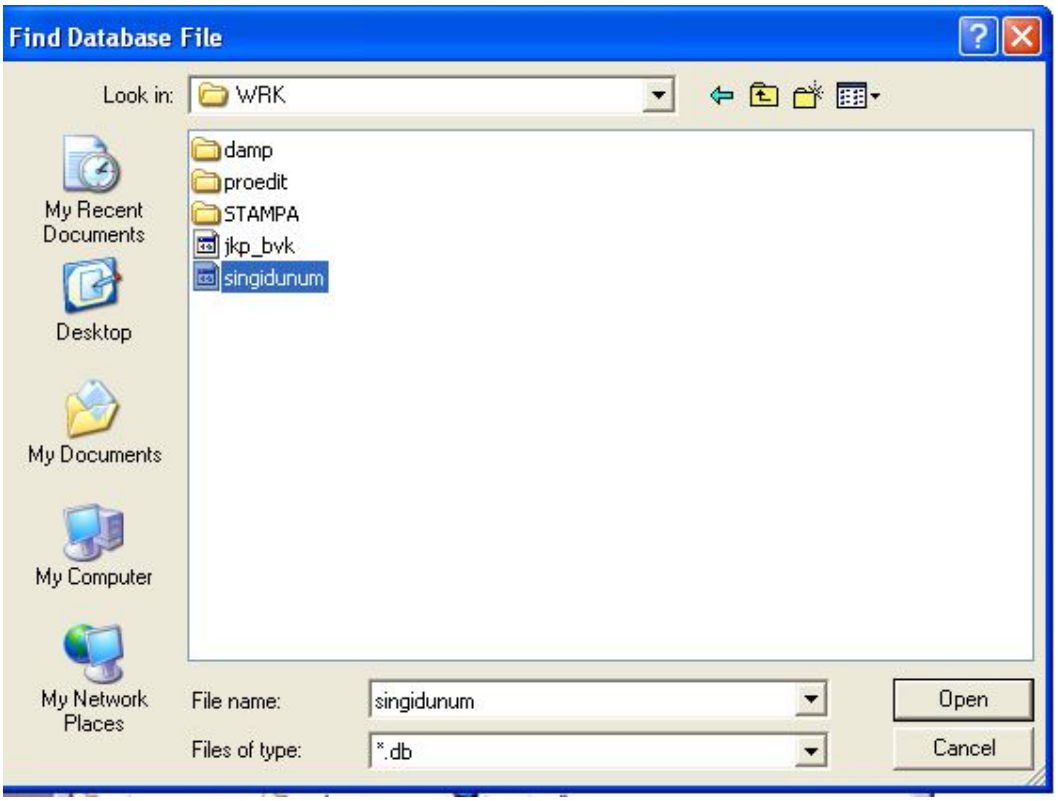


OpenEdge podataka rečnik pokretanje otvoriće se dijalog. Podaci rečnik je mesto gde možete da definišete i upravljate svim tabelama i poljima u aplikaciji baze podataka. Ako ste je započeli, možete da napravite sopstvenu kopiju Singidunum baze podataka, što je standard OpenEdge demo baze podataka. Morate da kopirate bazu podataka, tako da test procedure mogu da izmene da ga bez menjanja verzije ga instalirate u svoj direktorijum.



Kreiranje baze podataka pojavljuje se dijalog i traži od vas za ime svog primerka baze podataka. Tip Singidunum. Po podrazumevanoj vrednosti, OpenEdge stvara bazu podataka pod nazivom Singidunum u radni direktorijum. Ako želite bazu podataka da na neko drugo mesto formirate, možete da izaberete ,datoteke dugme pored fizičke baze podataka. Novo ime popunite polje za pregledanje direktorijuma struktura za pravu lokaciju za bazu podataka. Nakon što ste uneli novo ime baze, izaberite Start sa kopije neke ostale baze podataka opcija, a zatim izaberite datoteke dugme pored popuniti polja:

Pronađite Singidunum baze podataka u direktorijum OpenEdge instalirate, a zatim dvaput kliknite na ulazak ili izaberite Otvori dugme:



6.1 Tabela

Osnovni elemenat svake baze podataka je tabela. Svaka tabela sarži kolone i redove a u njihovom preseku nalazi se polje ili ćelija u koje Progress smešta karakterističan podatak. Red tabele predstavlja jedan slog koji sa svojim poljima daje grupu podataka čiji je karakter određen kolonama tabele. Definisanjem kolone, odnosno karaktera podatka koji će se slagati u koloni, polja u koloni se definišu i svrstavaju u određen tip, koji može biti: tekst, broj, datum/vreme, logičko polje, memo polje (polje u koga se može uneti tekst proizvoljne veličine), polje za unos brojeva u novčanom formatu, posebno polje u kome se

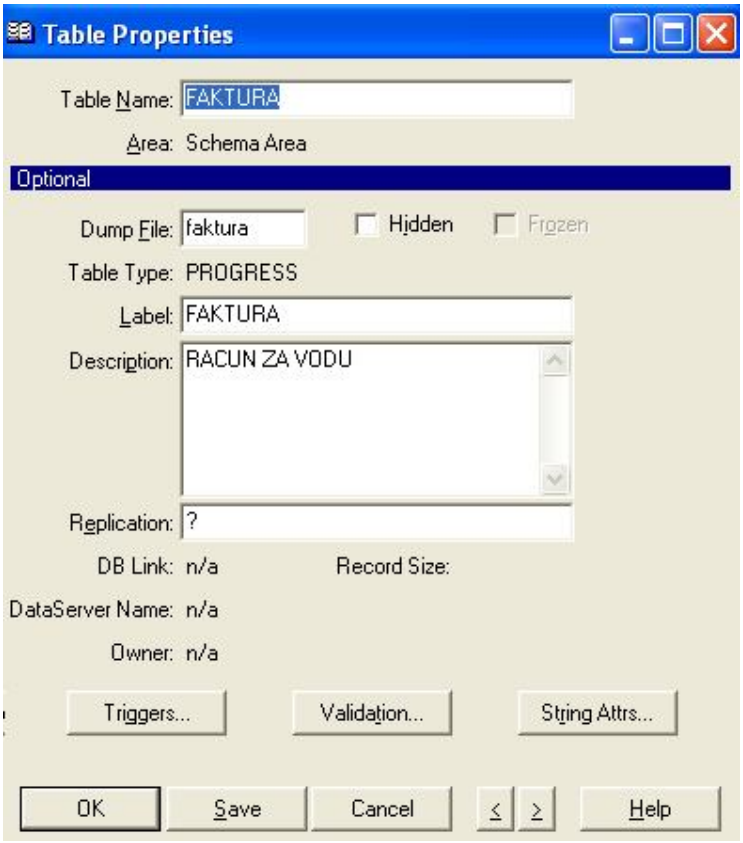
automatski po započetom unosu sloga inkrementira broj (polje AutoNumber), hiperlink polje (pokazivač na adresu na Internetu), OLE objekt polje (polje sa objektom povezanim iz drugog programa) i Look Up polje (polje čiji prikazani sadržaj se nalazi u drugoj tabeli).

Svako polje ima jedinstveno ime i sadrži podatke o redu

1. Izaberite klijenta tabelu iz tabele liste, a zatim izaberite polja dugme

2. Krećite se po listi da biste videli sva polja u tabeli.
3. Scroll doun listi tabela, a zatim izaberite tabelu reda.

Možete videti da je, pored IDcitanje polje, što daje svaki red jedinstveni broj, postoji ID citanje, koje će se koristiti kasnije "Korišćenje Osnovni 4GL Konstrukcije" da se poveže, ili se pridružite, red tabele Faktura zapis sa istim vrednostima.



Postoji mnogo nacina da vidite u rečniku. Ako želite da se prikazuje, samo napred, onda se ne menja ništa! Sve promene koje napravite mogu uticati na korake kasnije.. Ako izvršite promene u tabele ili polja, to može da poništi zapise koji su već u uzorku baze podataka ili zadržite neke od primera procedure rada.

4. Da napustite podaci rečnik, izaberite bazu podataka → Izlaz iz menija rečnik

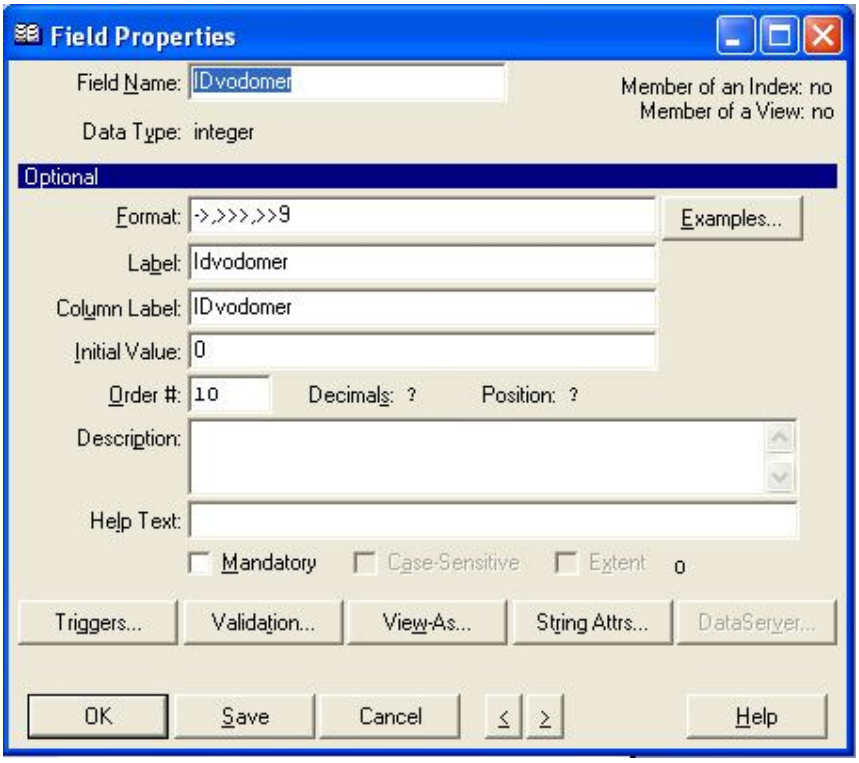
6.2 Tipovi podataka

Progress OpenEdge dolazi s nekim, već unapred definisanim, tipovima podataka (*built-in datatypes*).

Svaka vrednost u nekoj koloni i konstanta , mora imati tip podatka (*datatype*), koji je povezan sa specifičnim načinom spremanja, skupom ispravnih vrednosti, te sa zabranama (*constraints*). Kada kreiramo tablicu moramo odrediti tip podataka koji će se nalaziti u svakoj od njezinih kolona. Ugrađeni tipovi podataka dati su u sledećem primeru. Moram naglasiti da su ovo samo najčešće korišteni, možemo reći – opšti, tipovi podataka.

Tip	Opis	Dužina kolone i podrazumevana vrednost
CHAR (veličina)	Znakovni podaci fiksne dužine	Fiksne veličine za svaki red u tablici (s pratećim praznim poljima); maksimalna veličina je 2000 bajta po redu, podrazumevana veličina je 1 bajt po redu. Potrebno je uzeti u obzir koji se character set koristi, jedno-bajtni (US-ASCII) ili više-bajtni (srednjoeuropski character set), pre no što se odredi veličina.
VARCHAR2 (veličina)	Znakovni podaci promenljive (varijabilne) dužine. Mora biti određena maksimalna veličina	Promenjiva vrednost za svaki red, maksimalno do 4000 bajtova po redu. Također je potrebno uzeti u obzir znakovni skup (character set) pre no što se odredi veličina.
NCHAR (veličina)	Znakovni podaci fiksne dužine, određeni dužinom znakovnog skupa u bajtima, koja zavisi o nacionalnom znakovnom skupu (national character set).	Fiksne veličine za svaki red u tablici (s pratećim praznim poljima). Veličina kolone je broj znakova za fiksnu širinu nacionalnog znakovnog skupa ili broj bajtova za nacionalni znakovni skup promenljive širine. Maksimalna veličina je određena brojem bajtova potrebnih da se spremi jedan znak, s gornjom granicom od 2000 znakova po redu. Podrazumevana vrednost je jedan znak ili jedan bajt, ovisno o znakovnom skupu.
NVARCHAR2 (veličina)	Znakovni podaci promenjive dužine; ovisno o nacionalnom znakovnom skupu mogu biti znakovi ili bajtovi. Mora biti određena maksimalna veličina.	Promenjiva vrednost za svaki red. Veličina kolone je broj znakova za znakovni skup određene širine ili broj bajtova za nacionalni znakovni skup promenjive širine. Maksimalna veličina određena je s brojem bajtova potrebnih da se spremi jedan znak, s gornjom granicom od 4000 bajtova po redu. Podrazumevana vrednost je jedan znak ili jedan bajt, ovisno o zankovnom skupu.
LONG	Znakovni podaci promenjive veličine.	Promenjive vrednosti za svaki red u tablici: mogu poprimiti vrednosti do 231 bajtova, ili 2 GB po redu.
INTEGER	Brojčani podaci (numeric data) promenljive dužine. Maksimalna preciznost p i/ili scale je 38.	Promnljiva vrednost za svaki red. Maksimalni prostor potreban za određenu kolonu je 21 bajt po redu.
DATE	Podaci o datumu i vremenu fiksne veličine, početak računanja od 01.01. 4712 p.n.e do 31.12.9999.	Veličina je tačno određena na 7 bajtova po svakom redu u tablici. Podrazumevani format je string (na primer: DDMON-YY), koji je određen parametrom
RAW (veličina)	Binarni podaci promenljive veličine. Maksimalna vrednost mora biti određena.	Promenljiva vrijednost za svaki red u tablici, do 2000 bajtova po redu.
LONG RAW	Sirovi binarni podaci (raw binary) promnljive veličine.	Promenljiva vrednost za svaki red u tablici, do 231 bajtova, odnosno 2 GB po redu.
BLOB	Binary Large Objects. Binarni podaci velike i promjenljive dužjine.	Promenljiva vrednost za svaki redu u tablici, do 232 bajtova, odnosno 4 GB.
CLOB	Character Large Objects. Jednobajtni znakovni podaci promjenljive veličine.	Promjenljiva vrednost za svaki red u tablici, do 232 bajtova, odnosno 4 GB.
NCLOB	National Character Large Objects. Jednobajtni ili fiksni ili višebajtni podaci promnljive širine izraženi pomoću nacionalnih znakovnih skupova (national character set).	Promenljiva vrednost za svaki red u tablici, do 232 bajtova, odnosno 4 GB.

BFILE	Bianrni podaci spremljeni u "vanjskoj" datoteci (external file)..	Promenljiva vrednost za svaki red u tablici, do 232 bajtova, odnosno 4 GB.
ROWID	Binarni podaci koji predstavljaju fizičku adresu retka u nekoj tablici unutar baze podataka.	Fiksna veličina je 10 bajtova za prošireni ROWID (extended ROWID) ili 6 bajtova za ograničeni ROWID (restricted ROWID) za svaki red u tablici.
UROWID	Binarni podaci koji predstavljaju bilo koji tip adrese retka u tablici: fizički, logički ili vanjski (foreign).	Sprema do 4000 bajtova (ali za logički ROWID, samo 3950 bajtova se može koristiti za primarni ključ). Podrazumevana vrijednost je 4000 bajtova.



6.3 Index

Prilikom procesiranja zahteva za nekim podacima Progress koristi indekse za učinkovito lociranje tih podataka. Indeksi su posebno korisni kada aplikacije često pretražuju tablice zbog određenog skupa podataka. Na primer, pronaci fakuru veću od 5000 dinara.

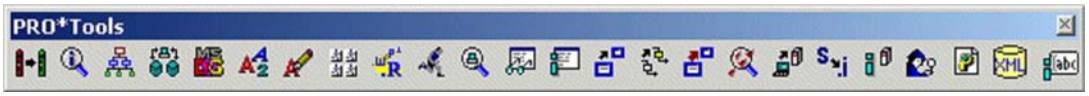
Indeksi se mogu stvoriti u odnosu na jednu ili više kolona u tablici. Jednom stvoren indeks Progress automatski održava i koristi.

Indeksi su potpuno logički i fizički neovisni o podacima. Mogu se obrisati i stvoriti u svakom trenutku, bez uticaja na tablicu ili ostale indekse. Međutim, ako se uklone indeksi, pristup indeksiranim podacima može biti usporen proračun se obavlja samo jednom.

Progress koristi sopstveni direktorijum put listi, pod nazivom Propat, koji se nalazi u okviru svojih inicijalizacia (progress.ini) datoteke. Kada instalirate OpenEdge, dobićete podrazumevani progress.ini datoteku u bin poddirektorijumu ispod instalirate direktorijum. Takođe možete da kreirate lokalnu kopiju ove datoteke ako želite da promenite njen sadržaj.

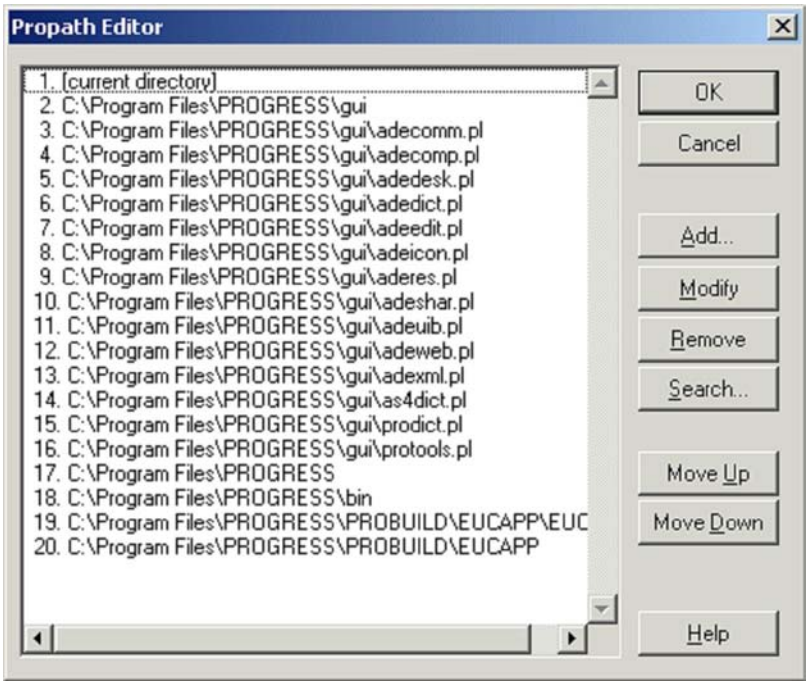
Ako želite da pogledate i održavanje Propath, postoji alat. To je jedna od brojnih korisni alatki možete da pristupite iz postupku urednika.

Da biste pristupili ovim alatkama iz postupku urednika, izaberite Alatke → Pro * Alatke. PRO * Alatke paleta se pojavljuje:



Možete premestiti i preoblikovanju PRO * Alatke paletu.
Da zadrži trajno novi oblik i paleta poziciju:
1. Desnim tasterom miša kliknite na PRO * Alatke paletu izvan područja gde se prikazuje ikonice.
Meni pop-up menija se pojavljuje

Propath Editor prikazuje spisak svih direktorijuma u kome Progress izgleda, kako da pronađete sve svoje izvršne datoteke i druge datoteke za podršku, kao i da pronađete procedure aplikacije. Ovo je podrazumevana Propat dobijate kada instalirate proizvod



6.4 Starting the AppBuilder

Alternativno, možete dovesti do AppBuilder direktno iz operativnog sistema Windows desktop izborom

AppBuilder opcije menija pod OpenEdge pokretanje grupe.

AppBuilder glavni prozor ima svoj meni i traka sa alatkama:, kao što je pokazano i

Pored Paleta prozor, Pro * Alatke prozor dolazi gore kada pokrenete AppBuilder.

U ovom poglavlju ćete izgraditi prozor koji prikazuje neka polja iz tabele korisnika, kao i kao pregledate koji pokazuje Fakturisanje. Dakle, ovo je kao primer, ukoliko izgrađen u prvom poglavlju, osim za dve razlike:

.

- Po izboru grafičke objekte izgradi svoj prozor, dobijate mnogo bolji interfejs.
- Zato što AppBuilder generiše većina podržava 4GL kod vas, vi u stvari morati da napiše još manje nego kod vas nije u prethodnim poglavljima

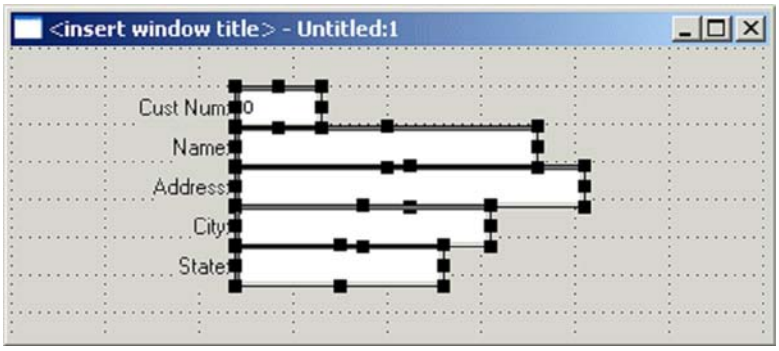
DB-fields - Odaberite polja iz baze podataka tabele da biste dodali u prozor.

Pregledaj - Koristite pretražujete kontrole za pregledanje zapisa baze podataka.

Button – Izaberite dugmad da biste dodali u prozor.

SmartDataObject - Koristite ovu ikonu i ikone ispod njega na paleti za montažu

SmartObjects u SmartWindows kao što je opisano u kasnijim poglavljima



Organizovati polja u više koristan način, prvo kliknite na prozor pozadina opozovite izbor polja, a zatim kliknite na svaku pojedinačnu polje i prevucite ga na više vizuelno razumnu lokaciju:

2. Ako želite da izaberete neki drugi predmet, kao što polje, kliknite na nju.

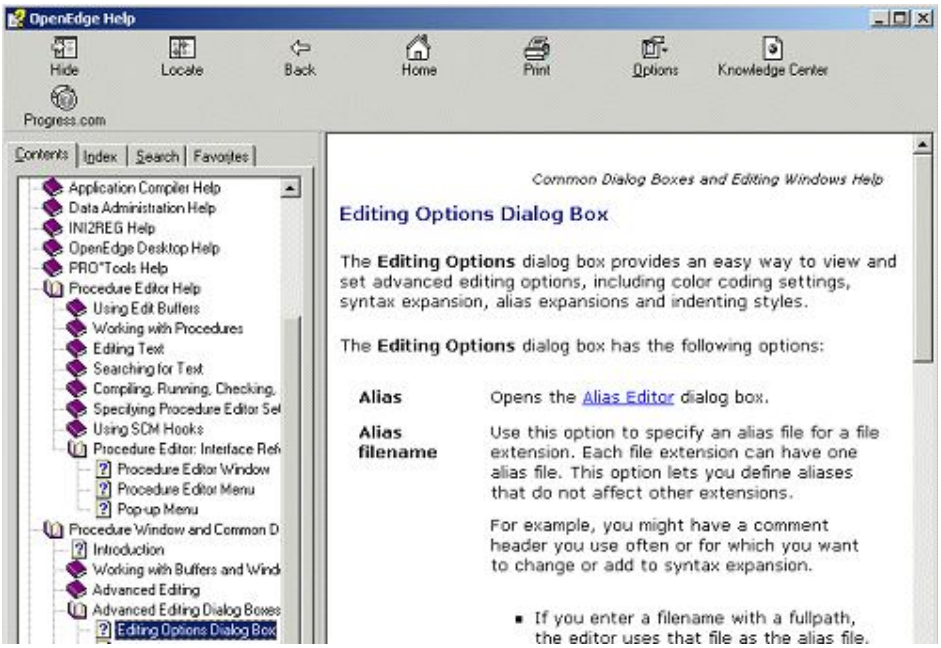
3. Da bi promenili na sebi prozor (na primer, da bi ste izmenili njen naslov),pritisnite taster CTRL-click (to, držite pritisnut taster CTRL i kliknite dole levo dugme miša).

AppBuilder je definisano podrazumevano ime za prozor C-Uin, koje možete videti u objekat polje. Promeni to Faktura. Kao što ćete videti kasnije, objekat je ime progresa 4GL promenljiva koja drži ručka ili pokazivač na prozoru definicija.

4. Izmeni naslov popunite polje u AppBuilder glavnom prozoru, da ga promeni od podrazumevani prozor titulu title> <INSERT prozora na nešto smisla, kao što su Fakturisanje.

6.5 Dobijanje pomoći na mreži

Možete pristupiti OpenEdge online sistem pomoći u svakom trenutku samo pritiskom na taster F1. Pomoć je kontekstno osetljiv, tako da uvek pokušava da dođe do pomoći datoteka koje trenutno radite. Na primer, ako pritisnite taster F1 iz Editovanje Dijalogu Opcije, možete dobiti pomoć od opcija u tom dijalogu:



6.6 Izrada formi, upita i izveštaja

Na osnovu kreiranih tabela potrebno je formirati forme za unos podataka. Naravno da se podaci mogu direktno unositi u tabele, ali potrebno je omogućiti lak i ne smetan unos podataka koji mogu da koriste i priučeni operateri na krajnje jednostavan način. U tu svrhu kreiraju se forme koje sadrže sve elemente jednog sloga tabele ali imaju mnogo prijatniji izgled.

Forma je komponenta baze podataka čija je funkcija interakcija sa korisnikom. Korisnik ne vidi tabele već forme. Forme se koriste za:

- unos podataka
- stvaranje "switchboard"-a tj. forme iz koje se otvaraju druge forme, pokreću upiti i tabele,
- stvaranje "dialog boxes"-a (upitnika) za primanje korisničkog unosa i onda se na osnovi tog unosa nastavljaju akcije (primer je dialog box kod parametarskih upita).

Forme za ovu aplikaciju kreirane su uz pomoć čarobnjaka ali nakon što su kreirane korišćenjem dizain opcije poboljšavani su neki elementi kako bi forme dobile na lepšem izgledu ali i funkcionalnocti (dodavanjem tastera za izlaz iz forme).

Za potrebe ove aplikacije kreirano je deset formi od kojih devet služe za direktno unošenje podataka u tabele, dok je jedna forma kreirana kao forma za unos kriterijuma za pretraživanje odnosno kriterijume za upite.

Unošenjem podataka kroz forme popunjava se baza i tada se stvara mogućnost za pretraživanje podataka po različitim kritetijumima. U tu svrhu formiraju se **upiti** odnosno **Query**. Dakle, upit je komponenta baze podataka kojom se koristimo kada se želi naći, izvući i/ili promeniti određeni podatak ili grupa podataka, iz jedne ili više tabela, pored toga oni se mogu koristiti kao podloga za **izveštaje ili forme**. Baš kod kreiranja upita najviše do izražaja dolaze ključevi koji su definisani pri formiranju tabela. Oni sada omogućuju da se iz tabela mogu izdvajati samo pojedini atributi, i time se umnogome olakšava i ubrzava pretraga podataka.

Pogled (view) je korisnički prilagođena prezentacija podataka iz jedne ili više tablica. Pogled se također može smatrati i "pohranjenim upitnikom". Pogledi ustvari ne zadržavaju niti pohranjuju podatke. Oni umesto toga “izvlače” podatke

Pogledi se često koriste za sledeće stvari:

- Ostvarivanje dodatnog nivoa sigurnosti u tablicama ograničavanjem pristupa samo na unapred određeni skup redova i kolona u tablici. Na primjer, pogled na tablicu može biti stvoren tako da se ne vide poverljivi podaci, kao što su plate radnika.
- Skrivaju kompleksnost podataka. Moguće je, na primer, u jedan pogled, iskombinirati 12 mesečnih tablica s podacima prodaje, da bi se dobili godišnji izveštaji. Jedan pogled također se može koristiti za stvaranje spoja (join), što je u stvari prikaz kolona i redaka iz više tablica koje su na neki način povezane. Pogled skriva činjenicu da su ovi podaci u stvari iz više tablica.
- Pojednostavljuje naredbe za korisnika. Pogled omogućava korisniku da odabere informacije iz više tablica bez potrebe za poznavanjem komplikovanih koreliranih upitnika. Na primer, upitnik može izvršiti zahtevne proračune s informacijama iz tablice. Snimanjem toga upitnika kao pogleda.

Izveštaji se koriste kada se želi neka grupa podataka, najčešće dobijena kroz upite, pripremiti za štampu. U osnovi izveštaj se ne razlikuje mnogo od forme, izuzev što je u izveštaju moguće samo gledati podatke.

Izveštaji se kreiraju na isti način na koji se kreiraju i forme, znači uz pomoć čarobnjaka ili direktno u Design wiev-u gde kreator aplikacije ima veću slobodu pri grafičkom uređivanju tj. slobodnije može stavljati slike, birati izgled i boju fonta,... Pored toga u samom izveštaju moguće je računati sume, proseke, izvoditi totale, itd.

Za potrebe ove aplikacije kreirano je dvanaest izveštaja i to tako da se osnovna forma izveštaja kreirala uz pomoć čarobnjaka a potom detalji korigovani u Design wiev-u.

Using the Section Editor

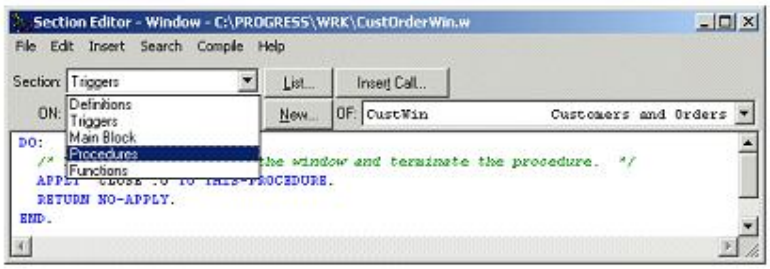
Sekcija Urednik pojavi. Ovo je mesto gde možete pregledati i pisati sve svoje 4GL kada koristite AppBuilder. Ako padajuće liste sekcija, videćete da postoje pet vrste sekcija Sekcija urednik održava za vas:

- Definicije - deo na vrhu vašeg postupka gde možete pisati promenljiva definicije i drugih izveštaja koji su potrebni u ceo postupak.
- Okidači - Blokovi koda koji izvrši kada se desi neki dogadjaj (na primer, kada korisnik izabere dugme). Vi ćete napisati neke aktivira sopstvene samo napred.
- Glavni Blok - deo 4GL koda koji se izvršava čim se postupak pokreće. Vi ćete pogledati na glavnom bloku ovaj uzorak postupka ispod i vidi šta to radi za vas.
- Procedure - unutrašnje procedure, baš kao i ona koju je napisao u poglavlju 3, "Pokretan je Napredak 4GL procedure."
- Funkcije - Korisnički definisane funkcije su kao i interne procedure, ali se vrate vrednost izjavu da ih koristi, kao što je izgrađen-unutra 4GL funkcije koje se koriste.

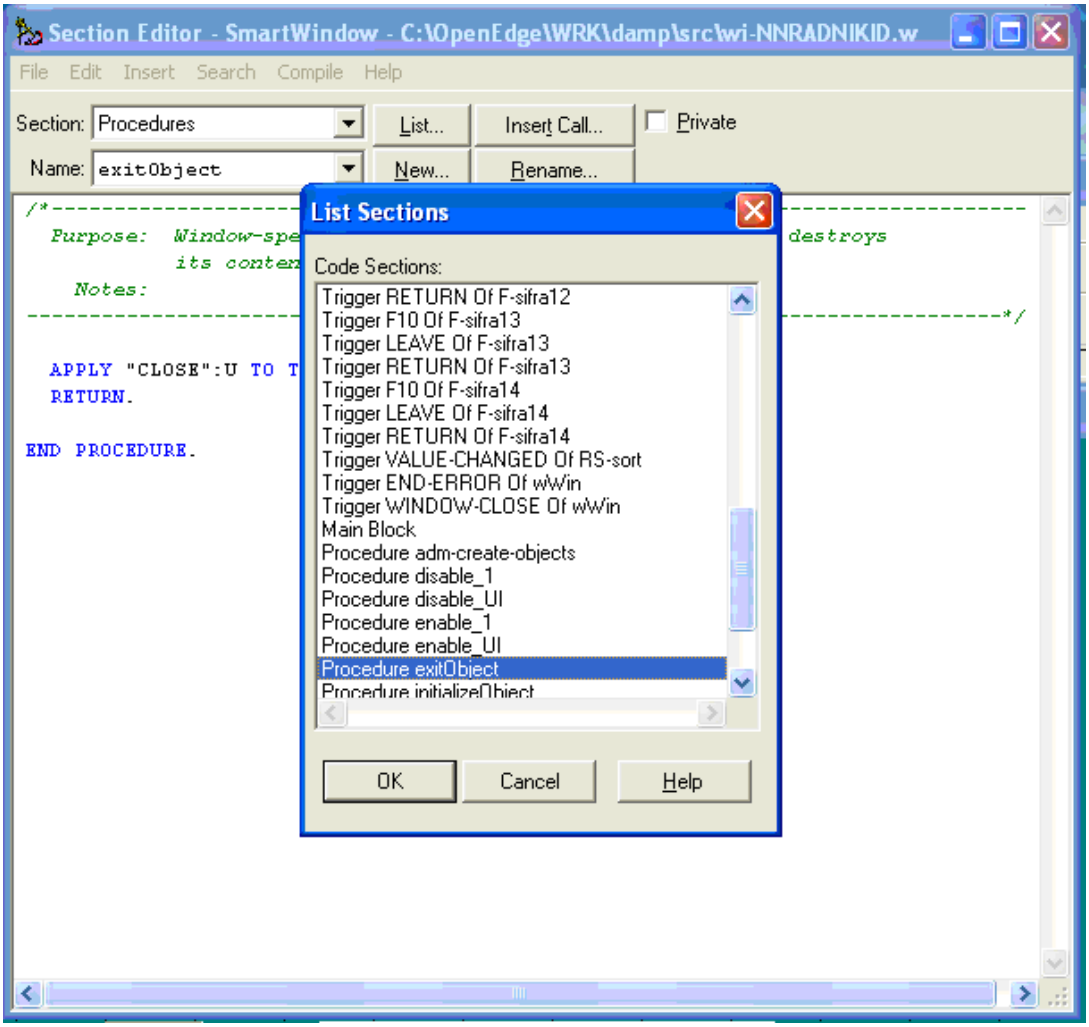
Progress je proceduralni programski jezik. To znači da ste napisali postavke koje možete sačuvati pojedinačno, pod nazivom procedura. Korisnik ima načina za kontrolu aplikacije tako što ćete odabrati dugme ili pravljenje menija selekcije, znaćete aktivirati kod koji postavlja blokove izjave da se kasnije izvršava kada se određena radnja dešava.

Progress 4GL postupak se sastoji od blokova. Sama procedura je glavni blok postupka. Postoji više načina za definisanje ostali blokovi u okviru glavnog postupka blok.

Progress 4GL postupak se sastoji od niza izjava jezika. Svaka izjava nema jednu ili više ključnih reči 4GL, zajedno sa ostalim tokovima kao što su imena polja baze ili promenljive imena. 4GL izjava obično završava periodu. Po konvenciji, izjavu definiše početak Kada definišete promenljive u proceduri, i kada definisati tabele baze podataka i polja, te treba im dati imena koja počinju slovom. Nakon toga, ime može da sadrži slova, numeričku cifre, periodi (), crtice. (-) i donje crte (_). Kada sačuvate 4GL proceduru,daje ime, kao, i to prati ista pravila imenovanja. Imajte na umu da, iako sam 4GL ima mnogo crticu ključnih reči, ne treba da koristite crtice iz baze podataka tabele i imena polja,kao i postupak i imena promenljivih u vašoj aplikaciji. Treba da pratite ove smernice zbog drugih programskih jezika koje ćete možda želeti da kombinuJete sa 4GL procedurom često ne dozvoljavaJu crticu kao znak u postupku ili parametar ime. I SQL pristup bazi podataka jezika, koji takođe možete da koristite za upravljanje podacima u bazi podataka OpenEdge korišćenje drugih alata, ne dozvoljava crtice iz tabele i imena polja. Dobar imenovanja za sve vrste imena 4GL procedura je da se koristi mešoviti slučaj (kao Fak polje) da se razbije na više imena reči ili druge identifikacione delova, korišćenjem velikih slova da identifikue početak svake reči ili Subpart u ime poboljšali čitljivost koda.



*PROGRAM SE PISE U TRIGGERIMA I PROCEDURI***.**



6.7 Trigeri u relacionim bazama podataka

U relacionim bazama podataka, u cilju očuvanja uslova integriteta, definišu se automatski poslovi koji se izvršavaju nakom obavljanja radnji unošenja, brisanja i ažuriranja podataka u relacijama. U relacionom upitnom jeziku SQL takvi postupci se nazivaju trigeri (okidači).

U sastavu savremenih SUBP, kakav je Oracle Designer-a 2000, nalaze se generatori aplikacija (Generate Module Forms), koje omogućavaju brzi razvoj aplikacija za unošenje, pretraživanje, ažuriranje i brisanje podataka. Umesto da se pišu programi, dovoljno je naznačiti željenu aplikaciju upotrebom jednostavnih ekranskih formi. Na ovaj način zadate instrukcije se kombinuju sa mogućnostima SUBP, a kao rezultat se dobija prototip aplikacije.

Ekranska forma koja se dizajnira pomoću generatora aplikacija, se sastoji od blokova i svaki blok odgovara jednoj relaciji BP.Unutar bloka, slog prikazuje jedan zapis (torku) osnovne relacije. U okviru bloka postoje polja, kao prazan prostor na formi. Polja se koriste za prikazivanje, unošenje i uređivanje podataka.Svako polje ima određenu

veličinu, poziciju i tip podatka, koji se u njega unosi. Korisnik određuje složenost dizajnirane forme i ima potpunu slobodu da izabere blokove, postavlja polja i vrši provere učinjenog.

Generator aplikacija sadrže tri nivoa dizajniranja ekranske forme:

- Prvi nivo predstavlja kreiranje blokova i polja. Najjednostavnija forma predstavlja prozore na osnovne relacije.Svako polje se može postaviti pojedinačno ili se može pozvati automatski uslovni generator aplikacija;
- Drugi nivo predstavlja definisanje blokova i polja. Na formi se mogu dodavati osnovne provere ili uputstva za pružanje pomoći. Na primer mogu se definisati veličine polja, uslovne vrednosti ili pomoćne poruke.
- Treći nivo predstavlja definisanje trigera (okidača). Ovo je najnapredniji nivo i mogu se napraviti složene provere i asistencije pisanjem kratkih SQL programa, najčešće u cilju očuvanja uslova integriteta.

6.7.1 Razvrstavanje naredbi u trigerima

Trigeri se mogu definisati kao skup naredbi koje počinju da se izvršavaju u nekom trenutku rada forme generatora aplikacija. Svaki triger može biti sastavljen od jednog ili više koraka, od kojih svaki korak sadrži jednu naredbu.

Naredbe, koje se koriste u trigerima, se mogu razvrstati u tri vrste. Te naredbe se mogu kombinovati u jednom trigeru , ali ne i u jednom koraku. Vrste naredbi su Progress naredbe, Progress naredbe geraratora aplikacija i korisničke izlazne naredbe.

1. **Progress naredbe** rade sa podacima iz relacija BP ili iz ekranske forme. Triger može pretrađivati ili modifikovati bilo koju relaciju za koju korisnik ima pravo pretraživanja ili modifikovanja. Progress sintaksa dozvoljava trigerima da prenose vrednosti između relacija i formi. Na primer, sledeći Progress triger automatski javlja naziv kupca iz relacije KUPAC u polje forme:
   SELECT NAZIV
   INTO: NARUDŽBA.NAZIV
   FROM: KUPAC
   WHERE KUPAC.BRKUPCA=:NARUDŽBA.BRKUPCA
   Navedeni triger izvršava naredbu uvek kada korisnik javlja, unosi ili menja broj kupca u formi NARUČIVANJE.
2. **Progress naredbe generatora aplikacija** simuliraju akcije funkcionalnih tastera i drugih operacija koje korisnik može izvršavati. Na primer, sledeći makrotriger pomiče kursor na blok ARTIKAL i izvršava pretraživanje:
   #EXEMACRO GOBLK ARTIKAL; EXEQRY;
   Makrotriger može biti vezan za funkcionalni taster, tako da korisnik može dobiti slog stavki pritiskom samo na jedno dugme. Ovaj makrotriger se može izvršavati svaki put kada korisnik pregleda blok NARUDŽBA
3. **Korisničke izlazne naredbe** pozivaju korisnikove napisane problemski orjentisanim programskim jezicima kao što su: C, Pascal, COBOL i drugi. Korisničkim izlaznim naredbamase mogu vršiti računske radnje na poljima formi, pristupi poljima u relacijama i drugo.

6.7.2 Načini udruživanja trigera

Trigeri se mogu udruživati za obavljanje različitih funkcija. Udruživanje trigera se može razvrstati u sledećih pet grupa:

1. Unos, pri pokretanju forme ili ulaska kursora u novi blok, slog ili polje
2. Pretraživanje, pre ili nakon dobijanja sloga
3. Promena, nakon promene vrednosti i pre ili posle prihvatanja unosa, ažuriranja ili brisanja u BP.
4. Izlaz, pri napuštanju forme ili kada kursor napusti blok, slog ili polje
5. Pritisak na taster, kada korisnik pritisne funkcionalni taster
6. Trigeri imenovani od korisnika, obični trigeri ili podtrigeri koji se pozivaju iz drugih trigera

Trigeri se mogu definisati na tri nivoa forme, i to: trigeri na nivou polja, kada su udruženi sa određenim poljem, trigeri na nivou bloka, kada su udruženi sa određenim blokom i trigeri na nivou forme, kada su udruženi sa celom formom. Na svakom nivou postoje trigeri koji se mogu primeniti.

Na nivou polja se mogu primeniti sledeći trigeri: pre polja, posle promene, posle polja, trigeri za tastere i trigeri imenovani od korisnika.

Na nivou bloka se mogu primeniti sledeći trigeri: pre bloka, pre pretraživanja bloka, posle bloka, pre sloga, posle pretraživanja sloga, trigeri prihvatanja, pre brisanja, posle brisanja, pre unosa, pre ažuriranja, posle ažuriranja, posle sloga, trigeri za tastere, i trigeri imenovani od korisnika.

Na nivou forme se mogu primeniti sledeći trigeri: pre forme, posle forme, trigeri za tastere i trigeri imenovani od korisnika.

Svaki triger se može definisati na njegovom nivou ili bilo kojem višem nivou. Područje rada trigera je određeno nivoom na kom je definisan.

6.7.3 Optimalni redosled definisanja trigera

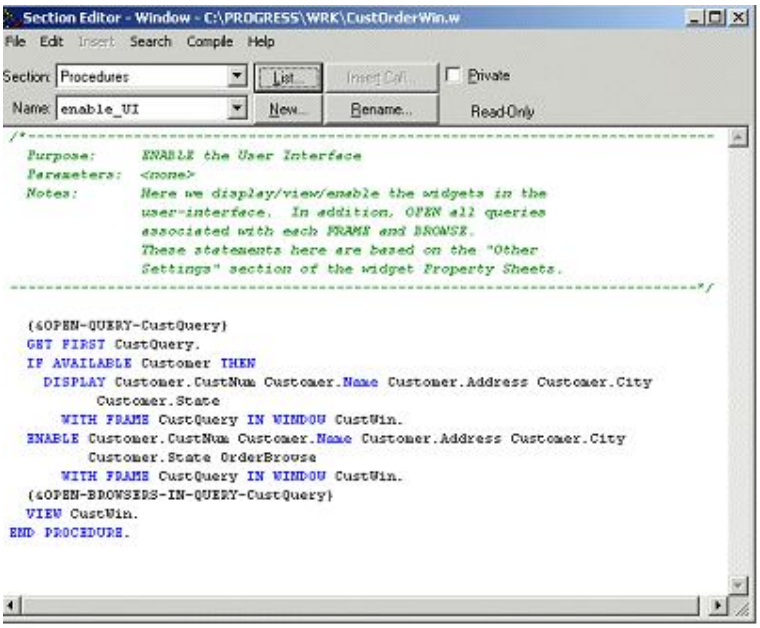
Da bi se trigeri definisali neophodno je navesti njegove korake i vlasništva, uključujući koji je tip trigera, odnosno kada će se izvršavati, koje naredbe će se izvršavati u svakom koraku i šta će se desiti ako korak uspe ili ne uspe.

Da se kreira novi triger potrebno je:

1. Name (Ime): Imenovati triger ili upisati tip trigera koji se želi kreirati. Tip trigera važi na nivou na kome se definiše. U slučaju potrebe dati naredbe TYPES ili KEYS, da se dobiju LIST TYPES ili LIST KEYS koji sadrže listu mogućih trigera na tekućem nivou ili listu mogućih trigera na bilo kom nivou.
2. Izabrati CREATE da se pokaže TRIGGER STEP prozor gde se mogu uneti naredbe koje se trebaju izvršiti.

Moguće je izvršiti modifikaciju postojećeg trigera, brisanje trigera ili promenu njegovog imena. Triger se sastoji od serije koraka koje se obično izvršavaju u nizu. Korak

se sastoji od jedne Progress Naredbe, Progress naredbe generatora aplikacija ili korsničke izlazne aplikacije.



6.8 Procedure

Naredbom CREATE PROCEDURE definiše se *stored* procedura na serveru. Sve procedure kreirane na ovaj način registrovane su u katalogu SYSCAT.ROUTINES.

Korisnički definisana procedura poziva se Progress naredbom CALL.

Mogu se kreirati tri tipa procedura:

Sopljašnje (external). Telo procedure je napisano u nekom programskom jeziku.

Sourced. Definiše se procedura koja se bazira na drugoj proceduri. Telo procedure je deo druge procedure (source). Ovakve procedure mogu biti ugrađene u aplikativni program ili izvršene upotrebom dinamičke SQL naredbe.

Procedura se definiše svojim imenom za kojim može da sledi lista argumenata izmeĐu zagrada.

Argumenti procedure su skalarne vrednosti, koje mogu biti različitih tipova. Argumenti mogu da služe za prenos vrednosti proceduri, prihvatanje povratne vrednosti procedure ili u druge svrhe.

CREATE PROCEDURE (external**)**

Definiše eksternu proceduru na tekućem serveru. Ova naredba može biti ugrađena u aplikativni program ili izvšena dinamički (kao dinamička komanda).

Nivo autorizacije za naredbu CREATE PROCEDURE mora da uključi bar:

CREATE_EXTERNAL_ROUTINE autorizaciju za bazu i bar jedno od sledećih prava:

IMPLICIT_SHEMA pravo za bazu, ako ime sheme procedure ne referiše na postojeću shemu

CREATEIN privilegiju nad shemom, ako ime sheme procedure referiše na postojeću shemu

DBADM nivo autorizacije

Sintaksa:

CREATE [OR REPLACE] PROCEDURE ime_procedure (lista_parametara) lista_opcija

OR REPLACE

Zamena definicije procedure, ako procedura sa istim imenom postoji na serveru. Postojeća definicija se uklanja, pre nego što se nova doda u katalog.

ime_procedure – može biti kvalifikovano (imenom sheme) ili nekvalifikovano. Unutar sheme ne smeju da postoje dve procedure sa istim imenom i istim brojem parametara, čak i kada su različiti njihovi tipovi.

Deklaracija parametra je oblika:

IN | OUT | INOUT ime_parametra tip_parametra [DEFAULT vrednost]

IN – Identifikuje parametar kao ulazni parametar procedure. Bilo koje promene ulaznog parametra unutar procedure nisu dostupne pozivajućoj aplikaciji nakon završetka procedure. IN je podrazumevani mod parametra procedure.

OUT – Identifikuje parametar kao izlazni parametar procedure

INOUT – Identifikuje parametar kao ulazno/izlazni

Ako procedura vrati grešku, OUT parametri ostaju nedefinisani, a INOUT ostaju nepromenjeni.

ime_paramtera se zadaje opciono i mora biti jedinstveno za proceduru

tip_parametra – može biti ugrađeni tip ili nizovski tip (za Java programski jezik).

DEFAULT – parametri sa DEFAULT vrednostima su poslednji u listi, tj. nakon DEFAULT parametara ne može da se navede parametar koji nema DEFAULT vrednost.

DEFAULT ne sme da se koristi za INOUT ili OUT parametre, kao ni za parametre tipa ARRAY, ROW ili CURSOR.

Za podrazumevane vrednosti se najčešće koriste *null* konstante, globale promenljive ili izrazi (unutar malih zagrada). Ako se koristi izraz, mora da odgovara tipu parametra i ne sme da modifikuje Progress podatke i vrši spoljašnje akcije.

7 Izrada fakture

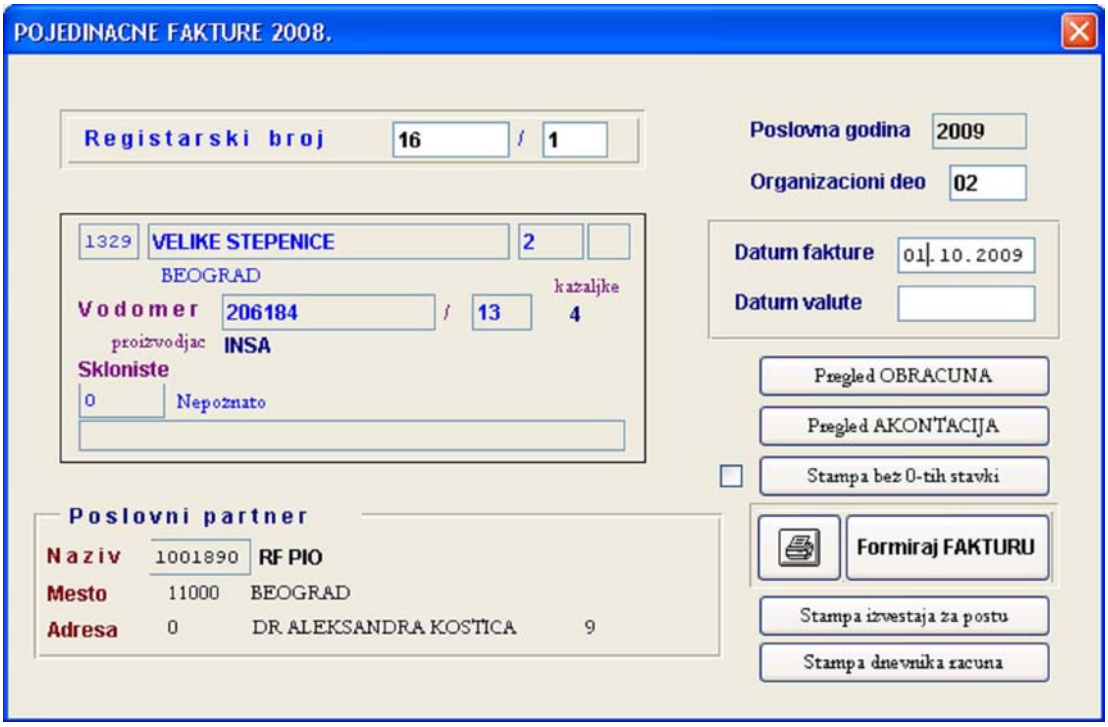
7.1 Fakturisanje

Nakon što su formirane obrade podataka po čitačkim knjigama ili granama, prelazi se na formiranje samih faktura za date potrošače.

Postoje tri načina formiranja faktura za BVK potrošače i posebno fakturisanje za SON potrošače koji se fakturišu preko Infostana.

BVK potrošačima se mogu formirati fakture na tri načina:

- Pojedinačne fakture, gde se formira faktura za konkretnog potrošača na konkretnom registru

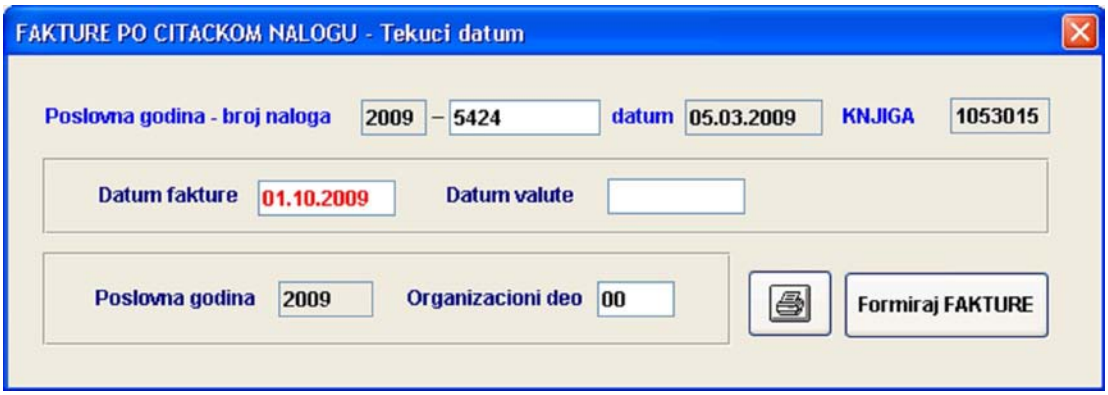


Unese se registar a program sam dodeljuje tekudi datum za datum fakture prilikom pritiska na taster ENTER na polju dok se datum valute računa kao tekudi plus 10 dana. Izborom na dugme Formiraj fakturu – kreirana je faktura za dati registar.

- Fakturisanje po čitačkoj knjizi, gde se formiraju fakture za sve potrošače iz date čitačke knjige

U ovom slučaju unosi se broj čitačke knjige a datumi se formiraju na prethodno opisan način i klikom na dugme Formiraj fakture – kreiraju se fakture za sve potrošače iz date čitačke knjige bez obzira po kojim su nalozima rađene obrade.

- Fakturisanje po čitačkom nalogu, gde se formiraju fakture za sve potrošače koji su se našli u istom čitačkom nalogu.



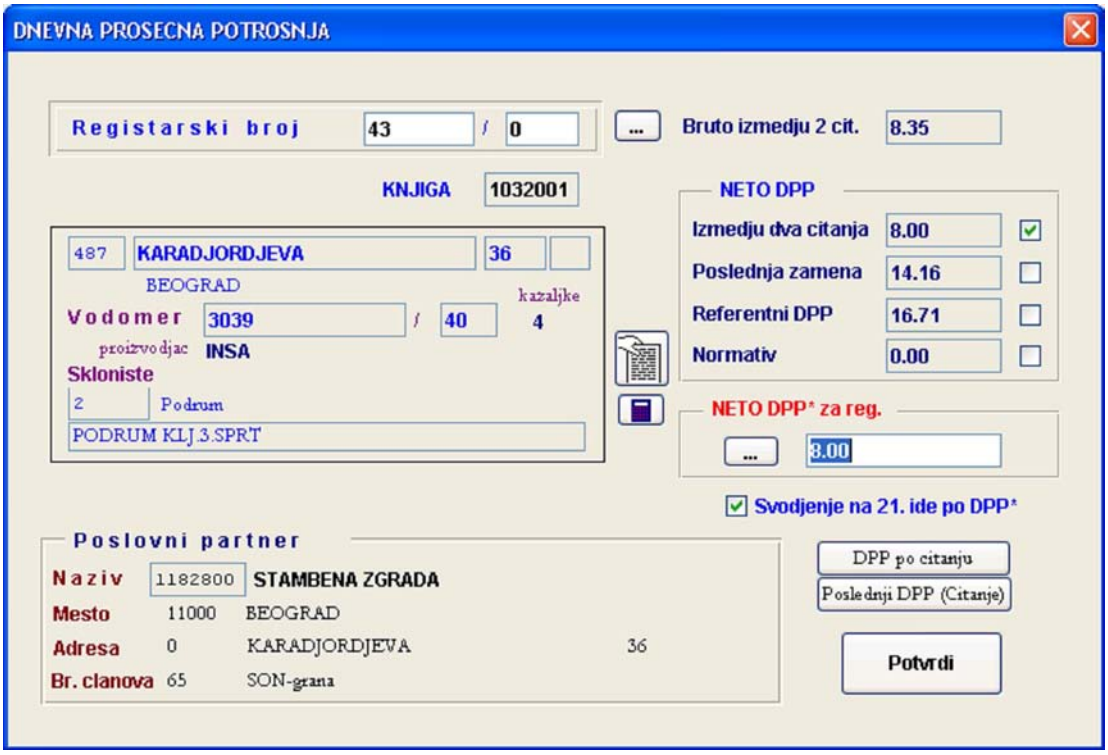
Princip je isti kao kod prethodnog slučaja, samo što je ovde potrebno uneti broj čitačkog naloga po kome je rađena obrada. Napominjemo da de se fakture formirati za sve registre koji se nalaze u datom naloga ali de obuhvatiti i eventualne obrade po drugim nalozima za prethodni period na datim registrima a za koje nije do tada bila formirana faktura.

Princip formiranja faktura za svakog potrošača je isti: **u fakturu ulaze sve obrađene stavke od datuma prijave potrošača do poslednjeg unetog stanja a koje do tog trenutka nisu dobile fakturu!** Naravno, fakture se formiraju samo za one potrošače za koje je u masci potrošača definisano polje Faktura – DA.

Referenti sami formiraju fakture za svoje potrošače po jednom od ova tri načina, ali se štampa faktura radi za sve fakture formirane toga dana od strane operatera Računskog centra. Prilikom formiranja štampe automatski se svakoj fakturi dodeljuje jedinstveni R-broj za datu godinu i dati mesec koji se odnosi na poštansku ekspediciju.

Za SON potrošače obrade potrošnje rade se na isti način kao i za BVK potrošače, ali postoji razlika što se obrada potrošnje od poslednjeg datuma čitanja do datuma fakturisanja ne radi po poslednjem dpp-u iz obrade, ved po aktontativnom dpp-u, odnosno, dpp-u koji je odredio referent po nekom od postojedih kriterijuma (po poslednjem čitanju,

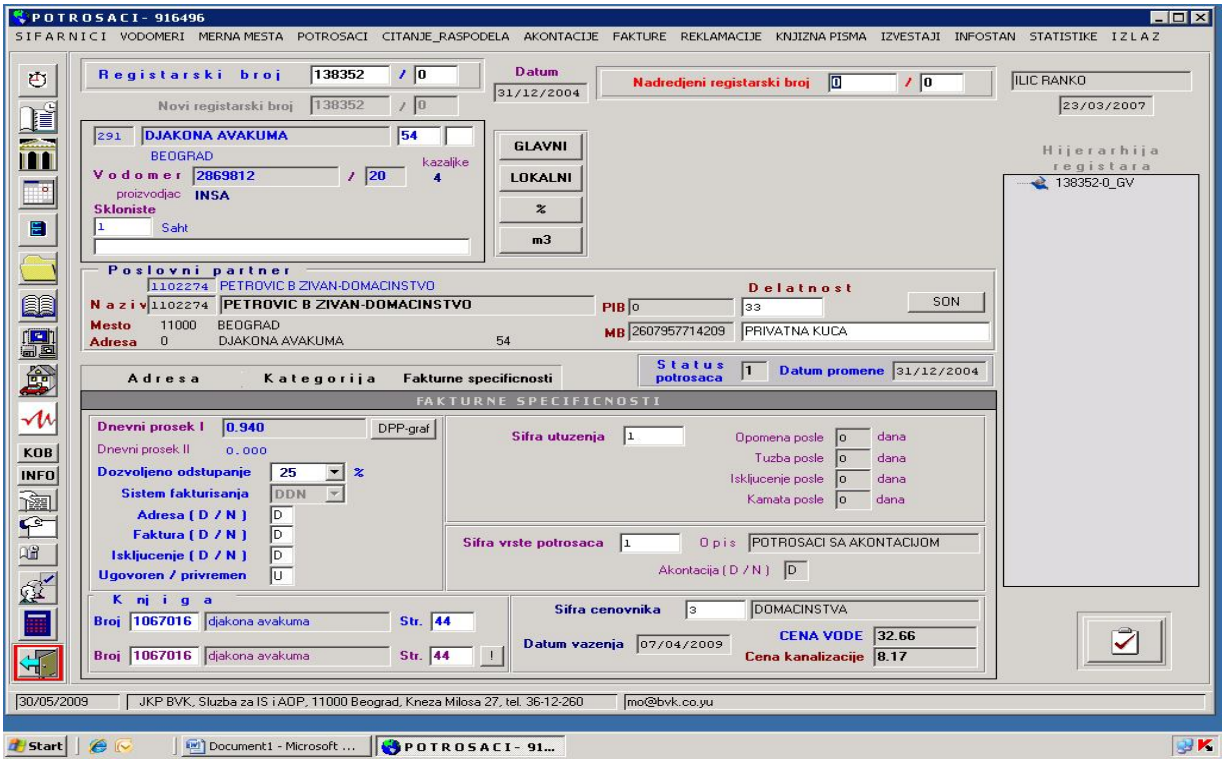
proseku u dužem periodu, na osnovu važedih normativa, ...). Korekcija akontativnog dpp-a se može raditi i kroz masku za obradu unetih stanja, ali i kroz posebnu masku iz menija AKONTACIJE -> Ažuriranje dpp-a za registar.



Nakon što se unese registar, program prikaže dpp-ove za istog po različitim kriteriumima (neto dpp između poslednja dva čitanja, dpp po poslednjoj zameni, referentni dpp, dpp po važedem normativu). Na istoj masci referent može pregledati i odgovarajude kartice potrošnje. Referent može odabrati neki od ponuđenih dpp-ova tako što klikne na polje pored datog dpp-a, ili može sam odrediti neki drugi dpp na osnovu istorije potrošnje ili na osnovu infomracija od samog potrošača.

Vrednost upisana u polje Neto dpp za registar upisuje se kao važeda pritiskom na dugme Potvrdi.

Inače, istorija promena akontativnog dpp-a za dati registar može se videti klikom na dugme sa tri tačkice koje se nalazi sa leve strane polja Neto dpp za reg.



02.10.2008 25.11.2008 450.40 0.00

Faktura broj: 332914 R 31.12.2008 10.01.2009 Iznos: 309.65

25.11.2008 31.12.2008 309.65 0.00

Faktura broj: 25534 R 24.02.2009 06.03.2009 Iznos: 309.65

31.12.2008 24.02.2009 309.65 0.00

Faktura broj: 88907 R 21.04.2009 01.05.2009 Iznos: 1,387.28

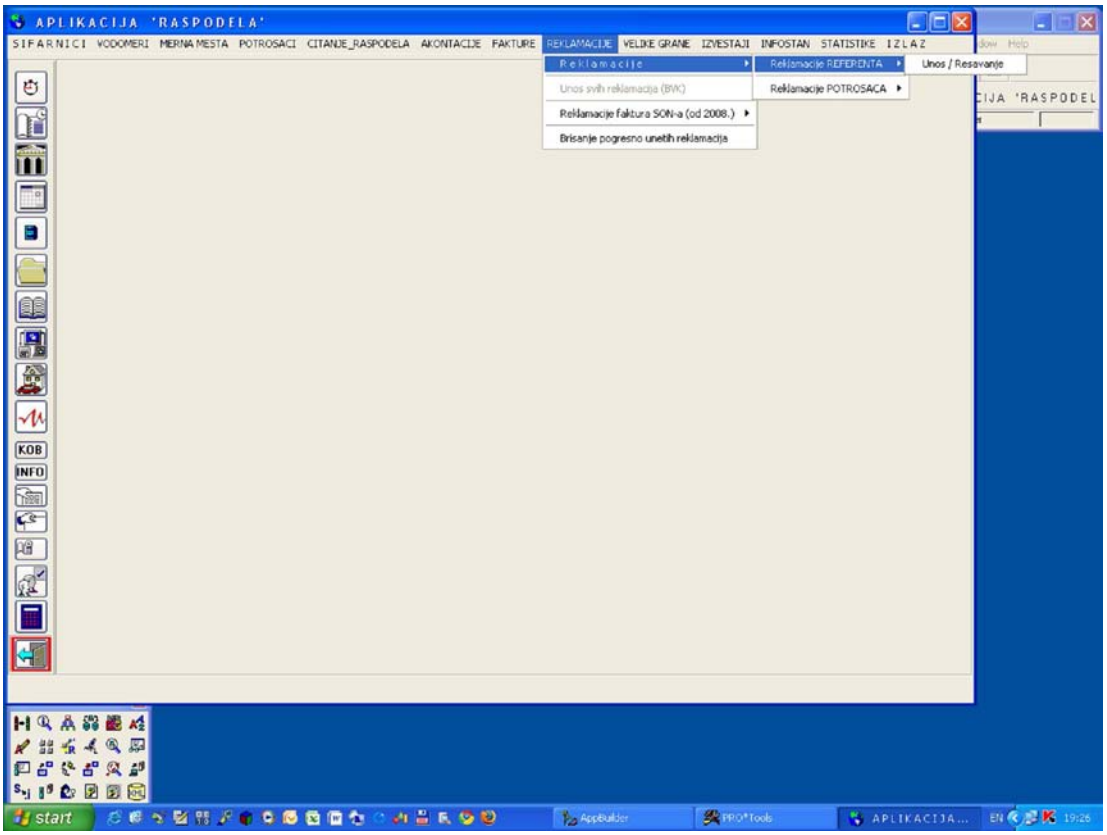
24.02.2009 07.04.2009 1,126.00 0.00
07.04.2009 16.04.2009 261.28 0.00
****

7.2 Storniranje obračuna i faktura - reklamacije

Osnovna novina je da više ne postoji storno fakture (ili raspodele) nezavisno od reklamacije – bilo da je reklamaciju uložio potrošač bilo da je inicirao referent. U tom smislu u meniju REKLAMACIJE nalaze se podmeniji Reklamacije REFERENTA i Reklamacije POTROŠAČA. Reklamacije referenta su one koje referent samostalno inicira bez intervencije potrošača i ne zahtevaju odgovor, ved služe da bi referent ušao u proces storniranja raspodele ili fakture na datom registru.

Reklamacije potrošača su reklamacije koje je inicirao sam potrošač podnošenjem formalne reklamacije. U budude, sve reklamacije potrošača, nezavisno da li se radi o SON-u ili fakturisanju BVK, morade da se obuhvataju kroz masku za unos svih reklamacija.

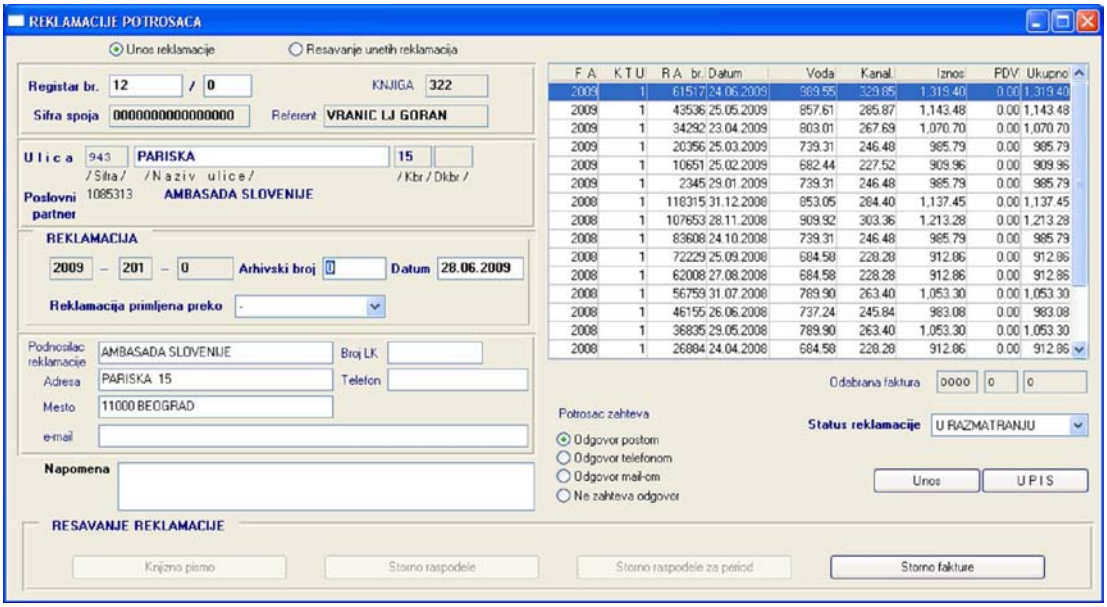
Privremeno, reklamacije za SON potrošače unose se kroz „staru“ masku koja se nalazi u podmeniju Reklamacije faktura SON-a (od 2008.), dok se reklamacije za potrošače kojima fakturiše BVK rade kroz nove maske.



Reklamacije potrošača – unos

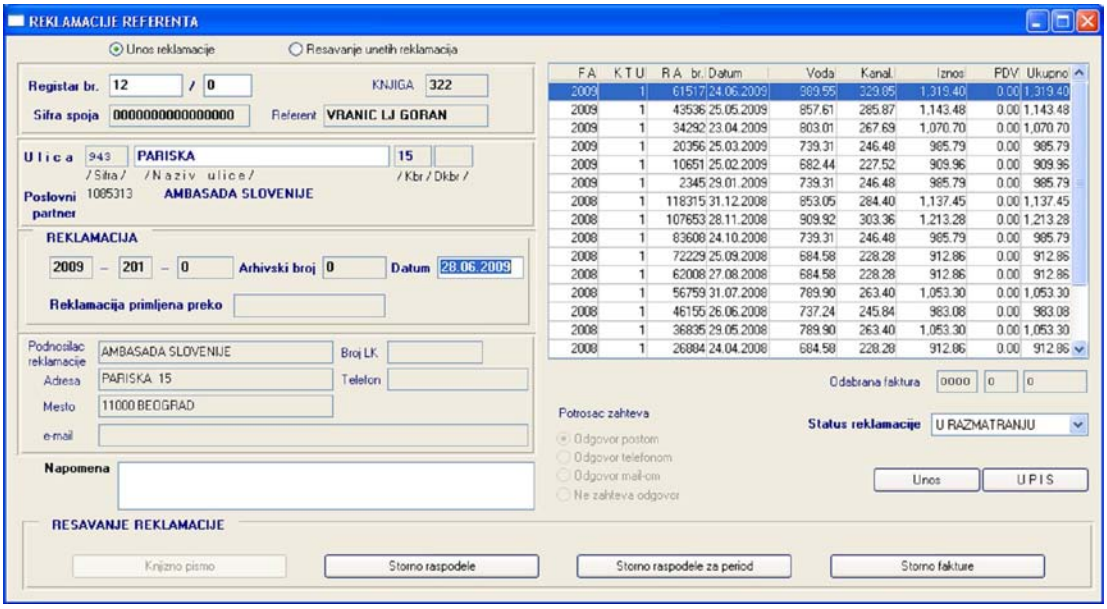
U masku reklamacija potrošača obavezno se unose slededi podaci: registarski broj potrošača, arhivski broj reklamacije, preko koga je primeljena reklamacija (BVK, Infostan – za SON potrošače, eventualno telefonski), podaci o podnosiocu reklamacije, kao i broj fakture na koju se potrošač žalio.

Izbor fakture vrši se jednostavno tako što se u browser-u (tabelarnom prikazu) u desnom delu maske odabere faktura jednostavnim klikom miša na njen broj. Odabrana faktura pojavljuje se u istoimenom polju.



Reklamacije referenta - unos

Reklamacija referenta predstavlja pojednostavljenu verziju reklamacije potrošača. Naime, referent unosi samo registarski broj spoja i ukoliko je u pitanju reklamacija fakture – potrebno je da odabere fakturu koja se stornira.



Na kraju se unos završava izborom opcije UPIS. U zavisnosti da li se radi o storniranju fakture, storniranju jedne raspodele ili stornu raspodela u nekom periodu, bira odgovarajude dugme. Ako se radi o stornu fakture, potrebno je status reklamacije prebaciti na OSNOVANA i izvršiti upis. U slučaju storna raspodele status ostaje U RAZMATRANJU i bira se odgovarajude dugme. Storno raspodele za period odnosi se samo na SOLO vodomere, dok se storno raspodele primenjuje na sve vrste potrošača koji nemaju fakturu po datoj raspodeli.

Napominjemo da su sama storna po svojoj funkcionalnosti ista kao i do sada samo što se sada parametri (registar, broj fakture) prenose automatski iz reklamacije.

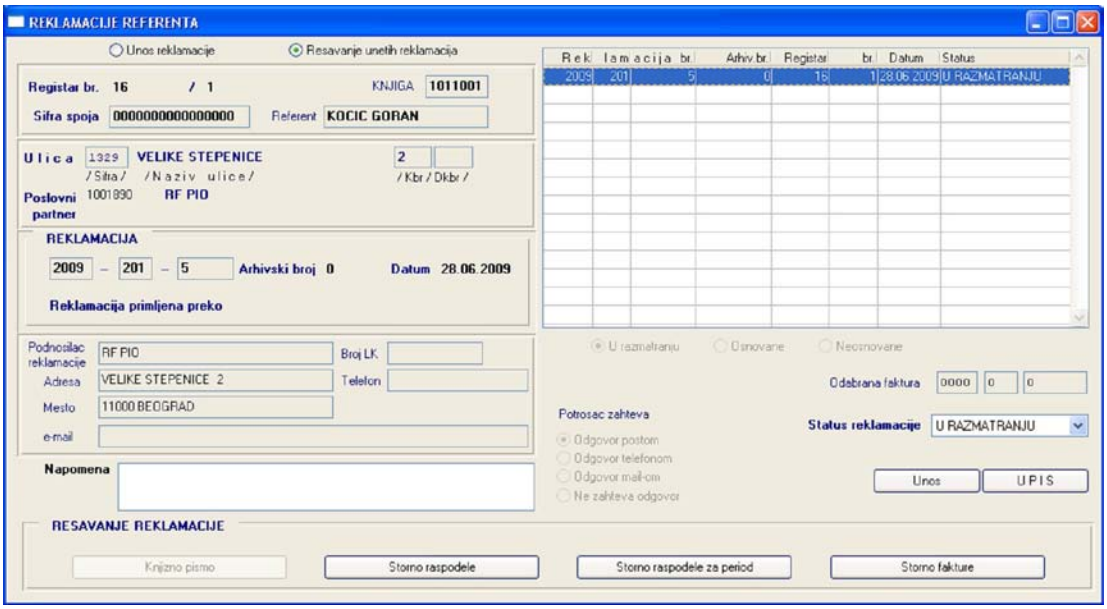
7.3 Rešavanje reklamacija

Kao što je već napomenuto, reklamacije se mogu rešavati odmah nakon što se unesu (upišu) u sistem, ali je mogude uneti više raklamcija (važi i za reklamacije potrošača i za reklamacije referenta) na različitim registrima a zatim se odabere opcija Rešavanje unetih reklamacija. U ovom slučaju u desnom delu maske pojavide se sve unete reklamacije za datog referenta koje su u statusu U RAZMATRANJU. U zavisnosti da li su unete kao reklamacije referenta ili potrošača – pojavide se u odgovarajudoj masci.

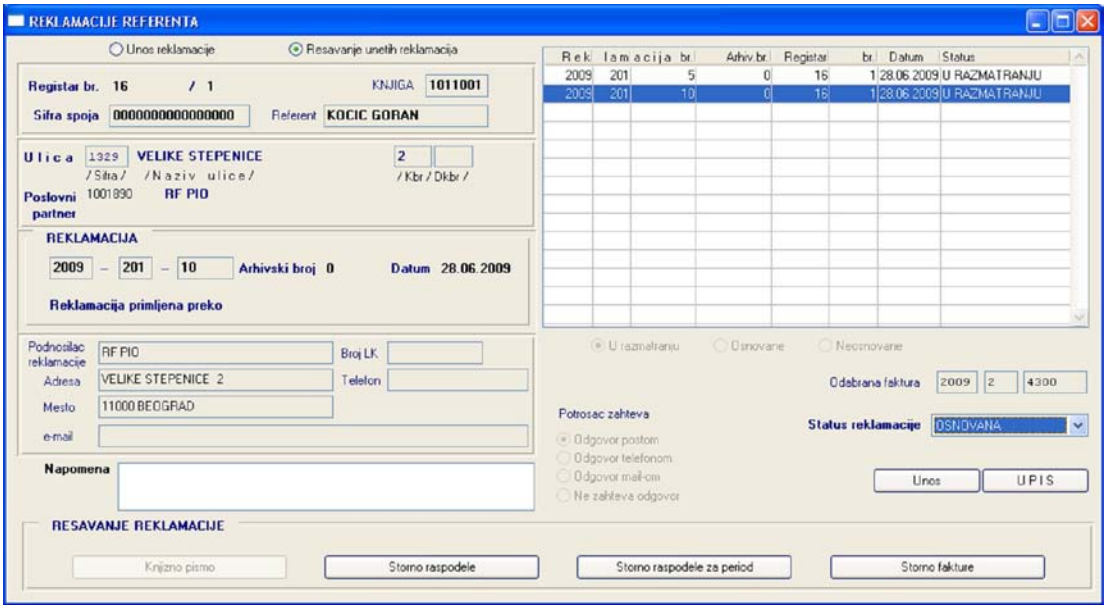
Na kraju se unos završava izborom opcije UPIS. U zavisnosti da li se radi o storniranju fakture, storniranju jedne raspodele ili stornu raspodela u nekom periodu, bira odgovarajude dugme. Ako se radi o stornu fakture, potrebno je status reklamacije prebaciti na OSNOVANA i izvršiti upis. U slučaju storna raspodele status ostaje U RAZMATRANJU i bira se odgovarajude dugme. Storno raspodele za period odnosi se samo na SOLO vodomere, dok se storno raspodele primenjuje na sve vrste potrošača koji nemaju fakturu po datoj raspodeli.

Napominjemo da su sama storna po svojoj funkcionalnosti ista kao i do sada samo što se sada parametri (registar, broj fakture) prenose automatski iz reklamacije.

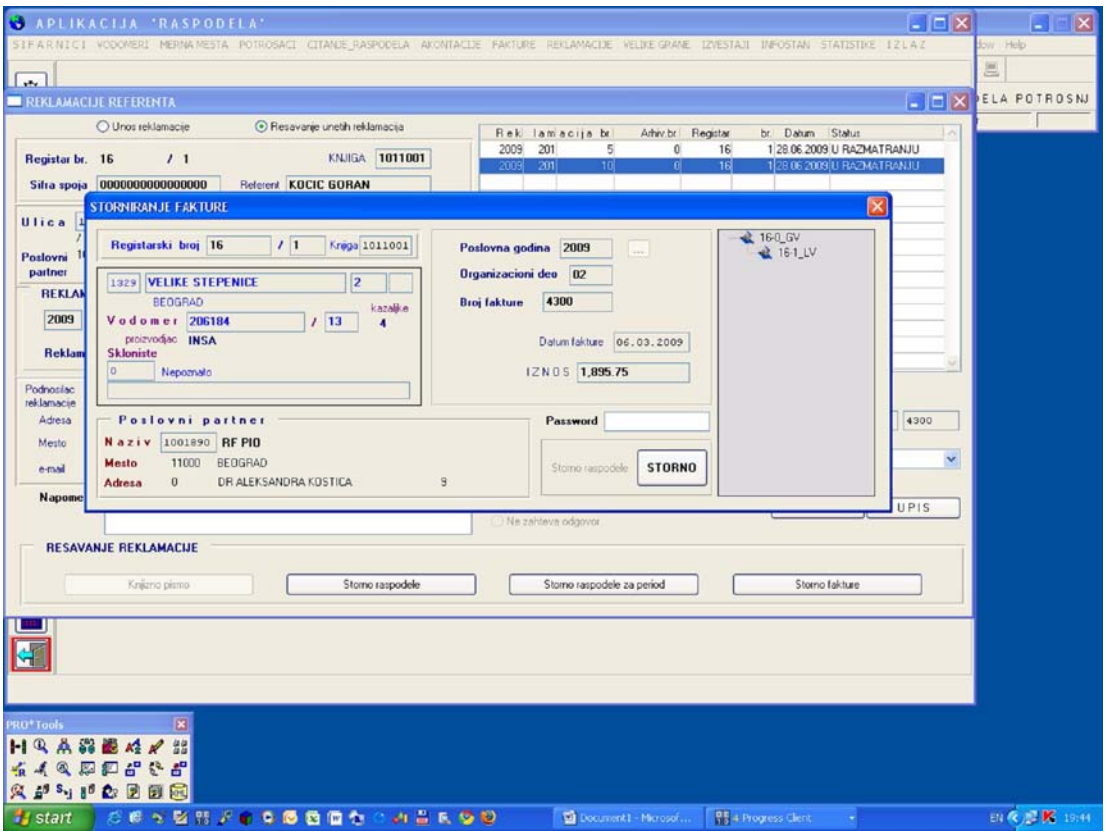
.



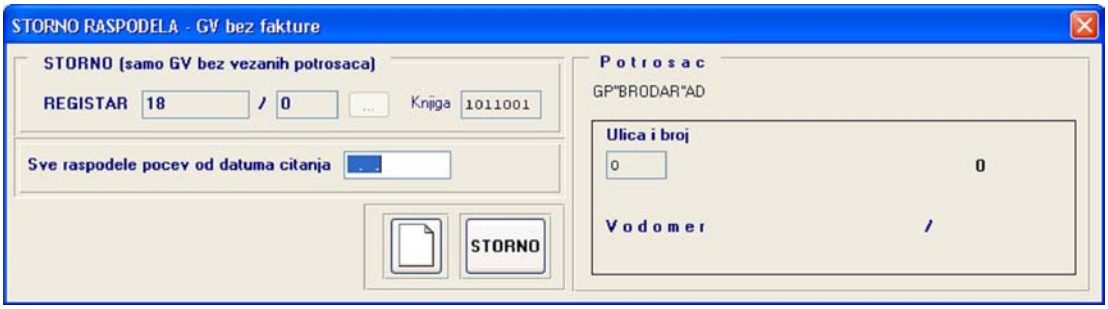
Izbor reklamacije koja se rešava svodi se na jednostavni klik mišem na broj željene reklamacije.



U slučaju izbora opcije storna fakture, sa maske reklamacija prenose se svi potrebni parametri i pojavide se nova maska koja sadrži registar i broj fakture koja se stornira. Referent klikom na dugme STORNO stornira datu fakturu i po završetku operacije maska se automatski zatvara i vrada ga na masku reklamacija, pri čemu se reklamacija po kojoj je storno rađen takođe gubi iz liste.

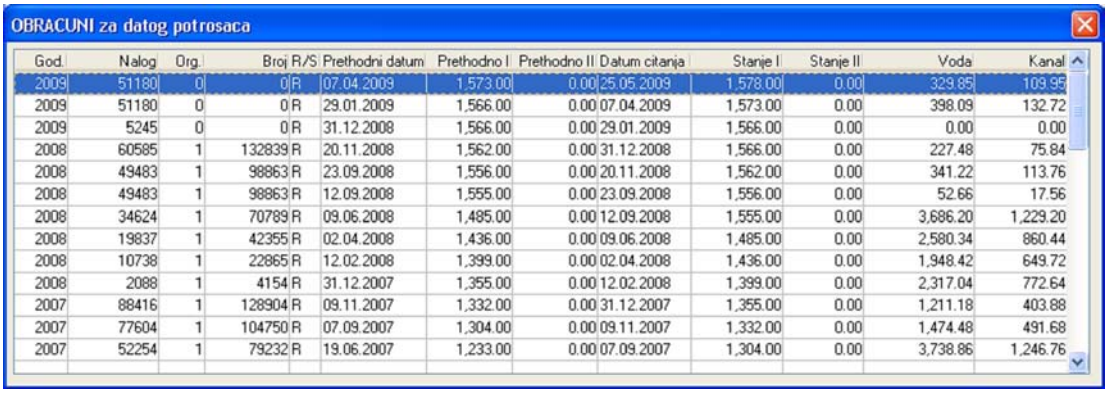


U slučaju storna raspodele za period (solo vodomeri), na masci koja de se pojaviti referent treba da unese samo datum od koga se storniraju sve raspodele do poslednje (naravno samo raspodele za koje ne postoji faktura).



Nakon toga odabere se dugme STORNO i po završenoj operaciji maska se automatski zatvara i vrada u masku reklamacija u čijoj listi više nema reklamacije po kojoj je izvršen ovaj storno.

Kod izbora storniranja raspodele, prvo se pojavi maska sa svim raspodelama za dati registar i referent bira raspodelu po željenom nalogu. Naravno, nije mogude stornirati raspodele za koje postoji faktura.



OBRACUNI za datog potrosaca

God.	Nalog	Org.	Broj	R/S	Prethodni datum	Prethodno I	Prethodno II	Datum citanja	Stanje I	Stanje II	Voda	Kanal
2009	51180	0	0	R	07.04.2009	1,573.00	0.00	25.05.2009	1,578.00	0.00	329.85	109.95
2009	51180	0	0	R	29.01.2009	1,566.00	0.00	07.04.2009	1,573.00	0.00	398.09	132.72
2009	5245	0	0	R	31.12.2008	1,566.00	0.00	29.01.2009	1,566.00	0.00	0.00	0.00
2008	60585	1	132839	R	20.11.2008	1,562.00	0.00	31.12.2008	1,566.00	0.00	227.48	75.84
2008	49483	1	98863	R	23.09.2008	1,556.00	0.00	20.11.2008	1,562.00	0.00	341.22	113.76
2008	49483	1	98863	R	12.09.2008	1,555.00	0.00	23.09.2008	1,556.00	0.00	52.66	17.56
2008	34624	1	70789	R	09.06.2008	1,485.00	0.00	12.09.2008	1,555.00	0.00	3,686.20	1,229.20
2008	19837	1	42355	R	02.04.2008	1,436.00	0.00	09.06.2008	1,485.00	0.00	2,580.34	860.44
2008	10738	1	22865	R	12.02.2008	1,399.00	0.00	02.04.2008	1,436.00	0.00	1,948.42	649.72
2008	2088	1	4154	R	31.12.2007	1,355.00	0.00	12.02.2008	1,399.00	0.00	2,317.04	772.64
2007	88416	1	128904	R	09.11.2007	1,332.00	0.00	31.12.2007	1,355.00	0.00	1,211.18	403.88
2007	77604	1	104750	R	07.09.2007	1,304.00	0.00	09.11.2007	1,332.00	0.00	1,474.48	491.68
2007	52254	1	79232	R	19.06.2007	1,233.00	0.00	07.09.2007	1,304.00	0.00	3,738.86	1,246.76

Po odabranoj raspodeli, potrebno je kliknuti na dugme STORNO nakon čega se po izvršenju operacije storniranja raspodele maska automatski zatvara i vrada u masku reklamacija (na čijoj listi više nije reklamacija po kojoj je izvršen ovaj storno).

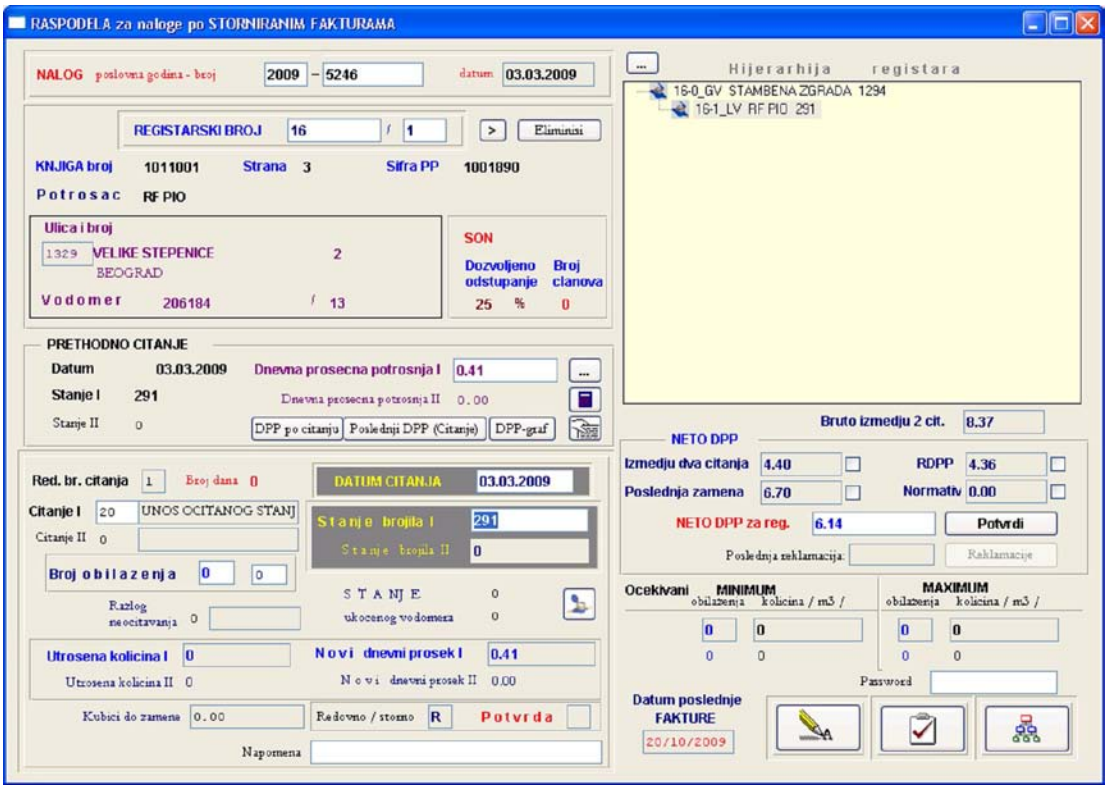
Treba napomenuti da se u kartici stavki (raspodela) sada pojavljuju dva sloga jedan sa oznakom S (storno) a drugi sa oznakom IS (označava da data stavka ima pripadajudi storno).

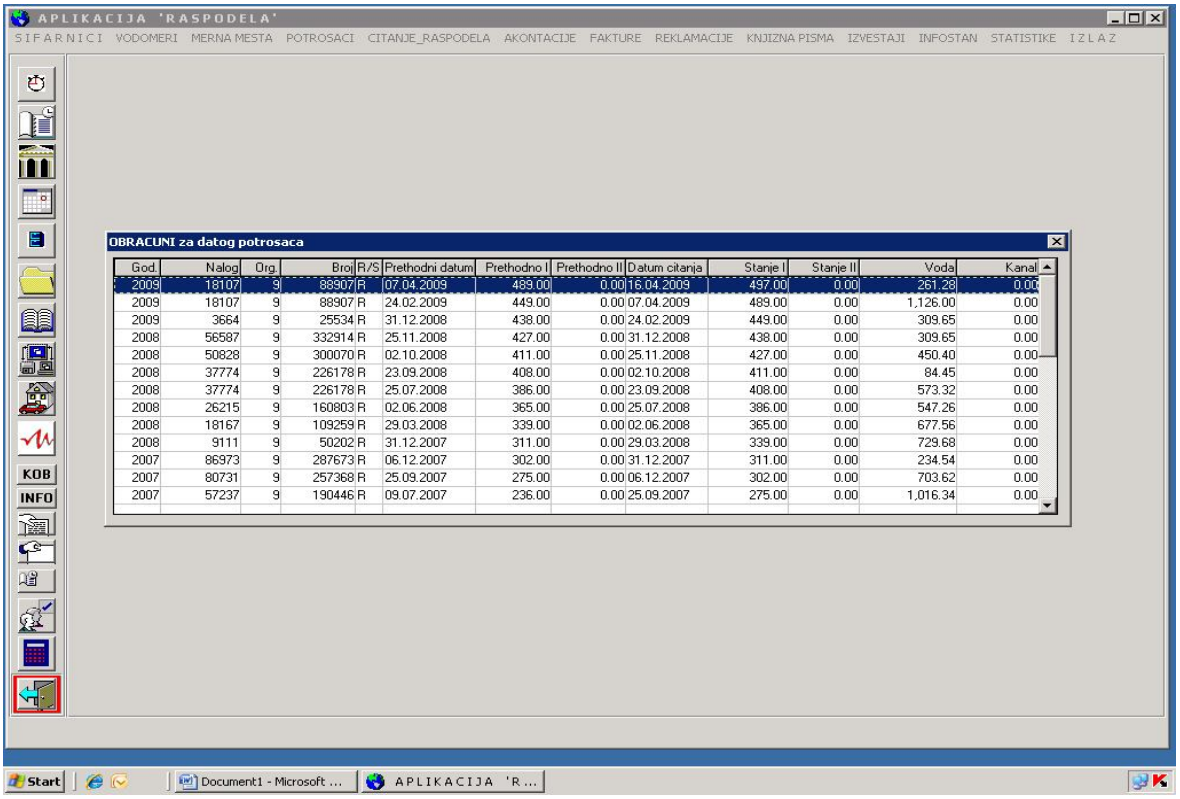
Takođe, dodat je i novi meni “Velike grane” kako referenti koji rade na ovoj vrsti potrošača ne bi morali da izlaze i ulaze u različite osnovne menije.

Novi obračun po storniranim raspodelama (obračunima i fakturama)

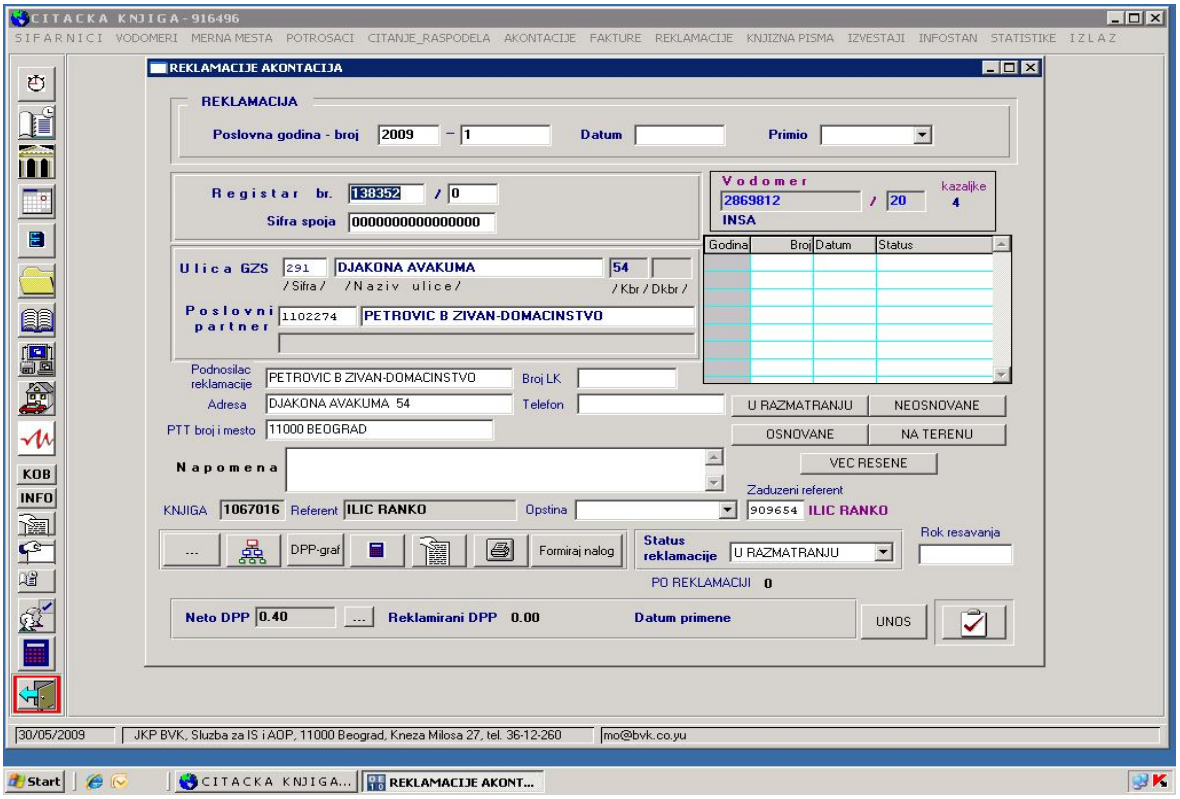
Ukoliko se radi o solo vodomerima, novi obračun za stornirani period radi se kao i redovni obračun – formiranjem novih čitačkih naloga za dati period, unosom ispravnih stanja i obradom popunjenih čitačkih naloga.

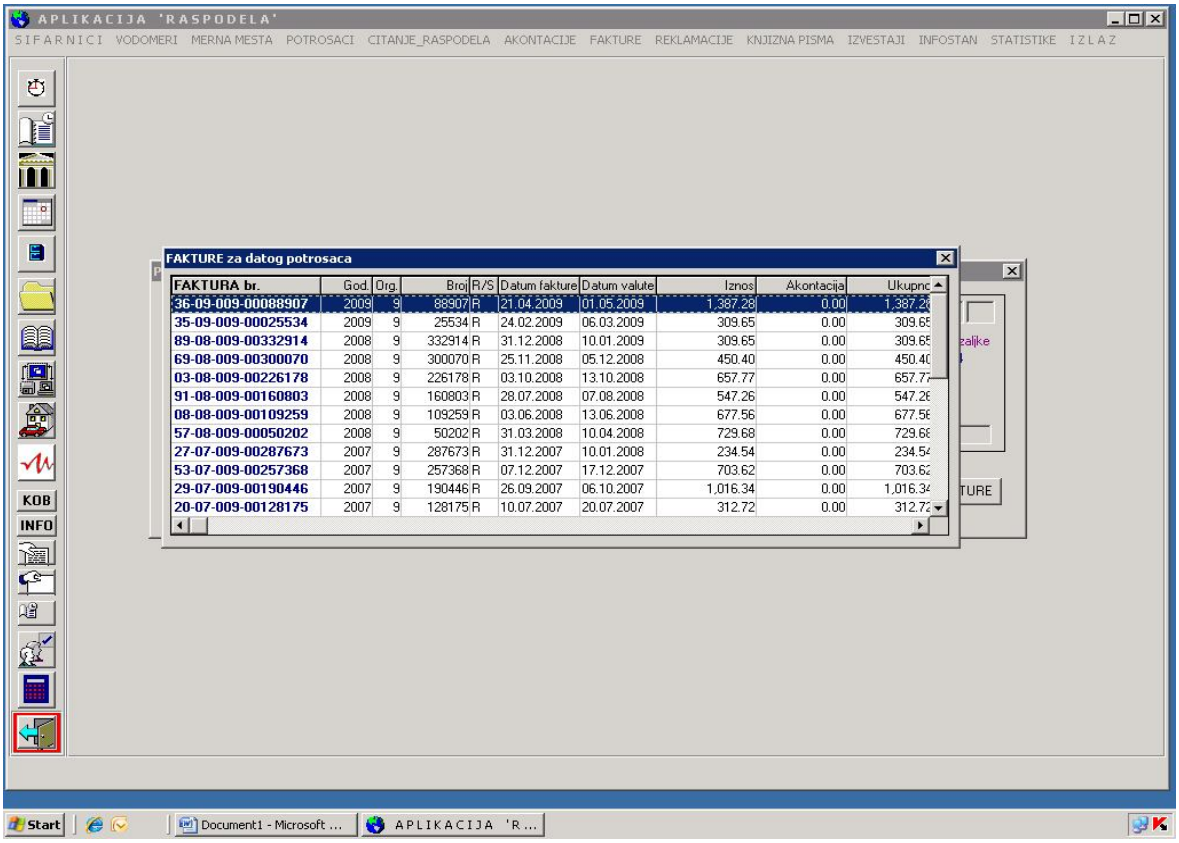
Ukoliko se radi o stornu raspodele ili fakture potrošača koji se nalaze u nekoj grani (postoje vezani potrošači), novi obračuni se rade po ved postojedim (storniranim) čitačkim nalozima i to kroz programsku formu koja se nalazi u meniju FAKTURE -> STORNO -> Obrada naloga za storniranu fakturu. U polje broja naloga unese se broj storniranog naloga i izbom tastera ENTER otvoride se registar za koji treba uraditi ispravku obračuna za dati period. Potrebno je uneti novo stanje ili, ako se radi po proseku, korigovati DPP i izvršiti upis (dugme "najki"). Nakon unosa svih storniranih registara iz datog naloga, pritiskom tastera F12 na polju broja naloga odabrati registar vrha grane i pritisnuti taster za novu raspodelu (obračun). Po završetku obračuna program de izbaciti obaveštenje o uspešnom okončanju operacije.



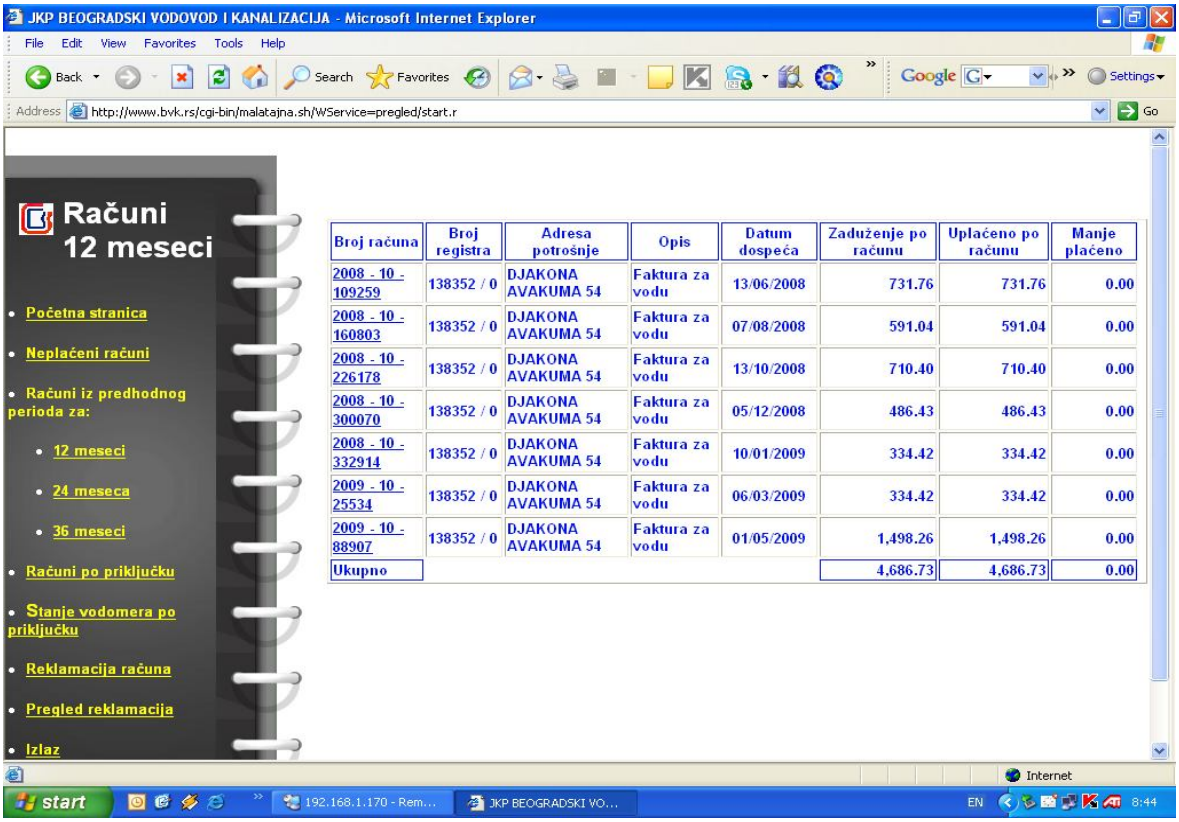


God.	Nalog	Org.	Broj	R/S	Prethodni datum	Prethodno I	Prethodno II	Datum citanja	Stanje I	Stanje II	Voda	Kanal
2009	18107	9	88907	R	07.04.2009	489.00	0.00	16.04.2009	497.00	0.00	261.28	0.00
2009	18107	9	88907	R	24.02.2009	449.00	0.00	07.04.2009	489.00	0.00	1,126.00	0.00
2009	3664	9	25534	R	31.12.2008	438.00	0.00	24.02.2009	449.00	0.00	309.65	0.00
2008	56587	9	332914	R	25.11.2008	427.00	0.00	31.12.2008	438.00	0.00	309.65	0.00
2008	50828	9	300070	R	02.10.2008	411.00	0.00	25.11.2008	427.00	0.00	450.40	0.00
2008	37774	9	226178	R	23.09.2008	408.00	0.00	02.10.2008	411.00	0.00	84.45	0.00
2008	37774	9	226178	R	25.07.2008	386.00	0.00	23.09.2008	408.00	0.00	573.32	0.00
2008	26215	9	160803	R	02.06.2008	365.00	0.00	25.07.2008	386.00	0.00	547.26	0.00
2008	18167	9	109259	R	29.03.2008	339.00	0.00	02.06.2008	365.00	0.00	677.56	0.00
2008	9111	9	50202	R	31.12.2007	311.00	0.00	29.03.2008	339.00	0.00	729.68	0.00
2007	86973	9	287673	R	06.12.2007	302.00	0.00	31.12.2007	311.00	0.00	234.54	0.00
2007	80731	9	257368	R	25.09.2007	275.00	0.00	06.12.2007	302.00	0.00	703.62	0.00
2007	57237	9	190446	R	09.07.2007	236.00	0.00	25.09.2007	275.00	0.00	1,016.34	0.00





FAKTURA br.	God	Org.	Broj	R/S	Datum fakture	Datum valute	Iznos	Akontacija	Ukupno
36-09-009-00088907	2009	9	88907	R	21.04.2009	01.05.2009	1,387.28	0.00	1,387.28
35-09-009-00025534	2009	9	25534	R	24.02.2009	06.03.2009	309.65	0.00	309.65
89-08-009-00332914	2008	9	332914	R	31.12.2008	10.01.2009	309.65	0.00	309.65
69-08-009-00300070	2008	9	300070	R	25.11.2008	05.12.2008	450.40	0.00	450.40
03-08-009-00226178	2008	9	226178	R	03.10.2008	13.10.2008	657.77	0.00	657.77
91-08-009-00160803	2008	9	160803	R	28.07.2008	07.08.2008	547.26	0.00	547.26
08-08-009-00109259	2008	9	109259	R	03.06.2008	13.06.2008	677.56	0.00	677.56
57-08-009-00050202	2008	9	50202	R	31.03.2008	10.04.2008	729.68	0.00	729.68
27-07-009-00287673	2007	9	287673	R	31.12.2007	10.01.2008	234.54	0.00	234.54
53-07-009-00257368	2007	9	257368	R	07.12.2007	17.12.2007	703.62	0.00	703.62
29-07-009-00190446	2007	9	190446	R	26.09.2007	06.10.2007	1,016.34	0.00	1,016.34
20-07-009-00128175	2007	9	128175	R	10.07.2007	20.07.2007	312.72	0.00	312.72



Broj računa	Broj registra	Adresa potrošnje	Opis	Datum dospeća	Zaduženje po računu	Uplaćeno po računu	Manje plaćeno
2008 - 10 - 109259	138352 / 0	DJAKONA AVAKUMA 54	Faktura za vodu	13/06/2008	731.76	731.76	0.00
2008 - 10 - 160803	138352 / 0	DJAKONA AVAKUMA 54	Faktura za vodu	07/08/2008	591.04	591.04	0.00
2008 - 10 - 226178	138352 / 0	DJAKONA AVAKUMA 54	Faktura za vodu	13/10/2008	710.40	710.40	0.00
2008 - 10 - 300070	138352 / 0	DJAKONA AVAKUMA 54	Faktura za vodu	05/12/2008	486.43	486.43	0.00
2008 - 10 - 332914	138352 / 0	DJAKONA AVAKUMA 54	Faktura za vodu	10/01/2009	334.42	334.42	0.00
2009 - 10 - 25534	138352 / 0	DJAKONA AVAKUMA 54	Faktura za vodu	06/03/2009	334.42	334.42	0.00
2009 - 10 - 88907	138352 / 0	DJAKONA AVAKUMA 54	Faktura za vodu	01/05/2009	1,498.26	1,498.26	0.00
Ukupno					4,686.73	4,686.73	0.00

8 Izgled računa za vodu potrošača:

JKP BVK
PIB 100346317

NALOG ZA PLAĆANJE

Datum plaćanja:

Potpis nalogodavca:

Nalogodavac	Primalac	
Osnova: FAKTURA ZA VODU	Ime: JKP BEOGRADSKI VODOVOD I KANALIZACIJA	
Iznos: ********1,498.26	Tekući račun: 160-6789-79	Datum dospeća: 01.05.2009 Šifra:
Tekući račun:	Mod. 97 Poziv na broj: 36-09-009-00088907	
Poziv na broj: 36-09-009-00088907		
Ime: PETROVIC B ZIVAN-DOMACINSTVO		
Adresa: DJAKONA AVAKUMA 54	Adresa: KNEZA MILOSA 27	
Mesto: 11000 BEOGRAD	Mesto: 11000 BEOGRAD	
	Upis banaka:	

Referenca: Broj računa: Iznos: Šifra trasata: Tekst:

0900900088907h 0000678979c 00000149826f 16000000h 11c

Molimo Vas, da na ovom mestu ne pišete i ne stavljate pečat

Pozicija predviđena za potvrdu plaćanja

PRIZNANICA
JKP BEOGRADSKI VODOVOD I KANALIZACIJA
11000 BEOGRAD, Kneza Miloša 27

Osnova: FAKTURA ZA VODU
Iznos: ********1,498.26
Mod. 97 Poziv na broj: 36-09-009-00088907
Ime: PETROVIC B ZIVAN-DOMACINSTVO
Adresa: DJAKONA AVAKUMA 54
Mesto: 11000 BEOGRAD

PETROVIC B ZIVAN-DOMACINSTVO
DJAKONA AVAKUMA 54
PIB potrosaca:
Adresa potrosnje: DJAKONA AVAKUMA 54
Registar: 138352 / 0

RACUN br.: 36-09-009-00088907

JKP BEOGRADSKI VODOVOD I KANALIZACIJA
PIB: 100346317
Datum prometa: 21.04.2009
Rok placanja: 01.05.2009
Sifra potrosaca: 1102274
PIN: 7513

Vrsta citanja	Prethodno datum	Prethodno stanje	Novo datum	Novo stanje	Potrosnja / m3 /	Odbitak / m3 /	Za naplatu / m3 /	VODA cena	VODA iznos bez PDV	VODA PDV	KANALIZACIJA cena	KANALIZACIJA iznos bez PDV	KANALIZACIJA PDV	UKUPNO bez PDV
21	24.02.09	449	07.04.09	489	40.00	0.00	40.00	28.15	1,126.00	90.08	7.04	0.00	0.00	1,126.00
20	07.04.09	489	16.04.09	497	8.00	0.00	8.00	32.66	261.28	20.90	8.17	0.00	0.00	261.28

Dnevna prosecna potrosnja na vodomeru: 0.94 m3

Obracun je izvrsen na osnovu Clana 4. i Clana 5. Odluke o nacinu placanja komunalnih usluga.
Prodavac je obveznik PDV-a, broj prijave 130960460.

OSNOVICA ZA OPOREZIVANJE ******1,387.28
PDV 8 % ********110.98

Mesto i datum izdavanja racuna: Beograd, 21.04.2009

Racun je punovazan bez potpisa i pecata

ZA NAPLATU (sa PDV) ********1,498.26

Postovani, na Internet prezentaciji JKP BVK (www.bvk.rs) mozete pogledati Vasa zaduzenja i uplate po racunima. Za logovanje koristite Vasu sifru potrosaca 1102274 i PIN 7513 .

PRESAVITI I ODCEPITI

JKP BEOGRADSKI VODOVOD I KANALIZACIJA
KNEZA MILOŠA 27, 11000 BEOGRAD
PIB: 100346317
KONTAKT CENTAR: 3 606 606
WWW.BVK.RS

P.P. 120

Postarina placena kod poste 11010 Beograd 48

11010 Beograd 48
R No 28499

RACUN br.: 36-09-009-00088907

PETROVIC B ZIVAN-DOMACINSTVO

DJAKONA AVAKUMA 54
11040 VOZDOVAC
VOZDOVAC

9 Zaključak

Cilj ovog diplomskog rada je prikazati bazu podataka Progress i pokazati način kako se ona koristi. U tu svrhu, kao osnovni izvor podataka korišćeni su podaci JKP BVK.

Namera je bila da se prikaže informatička podrška i efikasno poslovanje svih delova Sektora (od registracije potrošača, pogona zamene vodomera, evidencije o vodomerima, spojevima, potrošačima, rasturu vode do fakturisanja utroška vode) povezanih u jedinstvenu bazu podataka kao i informaciono povezivanje Sektora sa ostalim delovima JKP BVK, što je značajno doprinelo efikasnijem poslovanju celokupnog sistema.

Zaključak je da korištenje baze podataka Progress, uvelike pojednostavljuje rukovanje podacima.

Uvođenje novih informatičkih tehnologija u JKP BVK od izuzetne je važnosti za kvalitetno i efikasno izvršavanje zadataka. Na žalost, ne postoji standard po kojem bi moglo da se radi fakturisanje. Postoje pojedinačni modeli koji se moraju posebno pratiti, što otežava uvođenje novih informatičkih tehnologija.

Literatura

[1] Veinović, M, Šimić, G, **Uvod u baze podataka**, Beograd, 2007.

[2] Đorđević, B, Pleskonjić, D, Maček, N, **Operativni sistem Unix,** Viša elektrotehnička škola Beograd, 2004.

[3] Njeguš, A., **Poslovni informacioni sistemi**, drugo izdanje, Univerzitet Singidunum, Beograd, 2008.

[4] Lazarević B., Marjanović Z., Ančić N., Babarogić S., **Baze podataka**, FON, Beograd 2003

[5] Welling L. Thomson L. **Priručnik za My SQL,** Mikro knjiga , Beograd 2005

[6] Korišćeni podaci JKP BVK

[7] http://www.bvk.rs/

[8] http://**progress**.com/**progress**.../**openedge_10_a**/

[9] http://web.progress.com/en/business-need/responsive-process-management.html

[10] http://en.wikipedia.org/wiki/Progress