Podatkovna baza za zvezne industrijske procese

Iztok Savnik*, Matjaž Šubelj**, Giovanni Godena**

* Laboratorij za informatiko
Fakulteta za računalništvo in informatiko
Univerza v Ljubljani

** Odsek za računalniško avtomatizacijo in regulacije
Institut Jožef Stefan, Ljubljana

Povzetek. Podatkovna baza za shranjevanje podatkov v aplikacijah za vodenje zveznih industrijskih procesov vsebuje dve vrsti podatkov: statične in dinamične podatke. Statični podatki definirajo strukturo in lastnosti industrijskega procesa. Z dinamičnimi podatki se spremljajo lastnosti objektov ter izvajanje postopkov med delovanjem zveznega procesa. V prispevku je predstavljen prototip podatkovne baze procesnega dela Centralne čistilne naprave v Domžalah. Opisana je struktura statičnega dela podatkovne baze ter rešitve za predstavitev dinamičnega dela podatkovne baze, ki predstavlja jedro procesne podatkovne baze.

1. Uvod

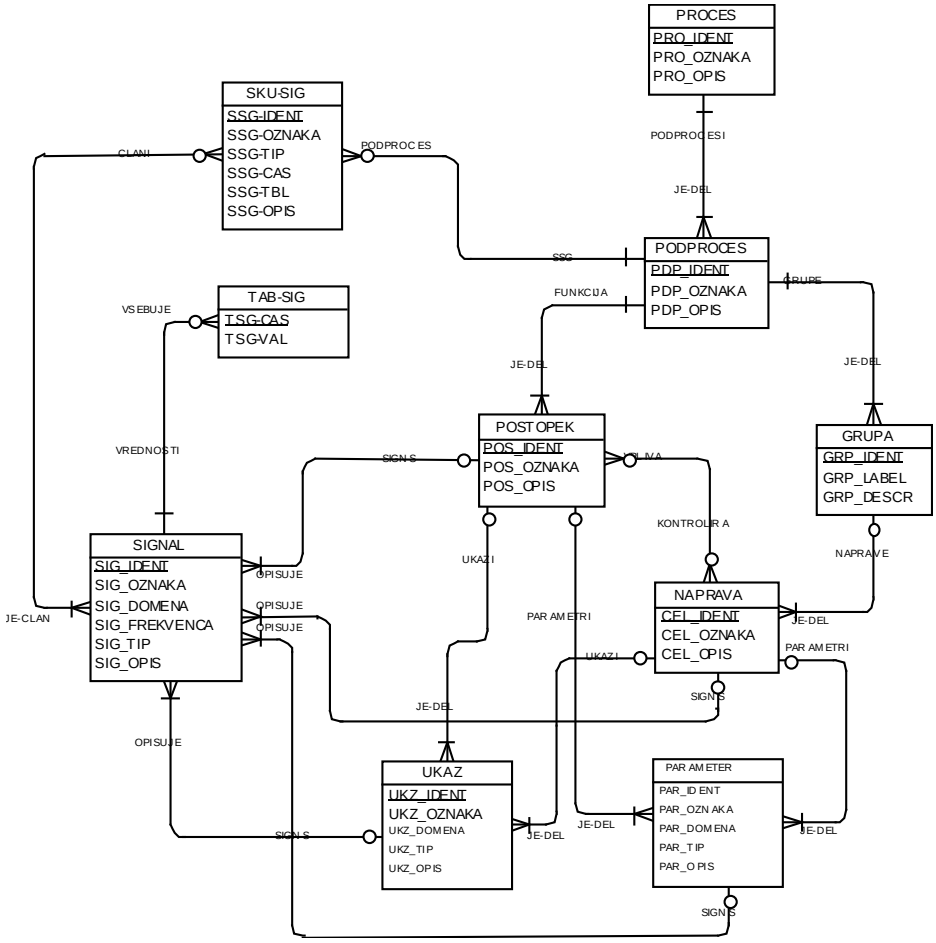
Podatke, ki se shranjujejo v sistemu za vodenje zveznih procesov lahko razdelimo na dva tipa podatkov: statične in dinamične podatke. S statičnimi podatki je predstavljena struktura sistema, to so opisi in definicije naprav, skupin naprav, podprocesov, meritev, ukazov ter drugih entitet, ki sodelujejo v procesu. Struktura statičnega dela procesne podatkovne baze je fiksna; definiramo jo z relacijskim podatkovnim modelom. Z dinamičnimi podatki beležimo lastnosti snovi in objektov, ki so predmet obdelave procesa, sledimo stanjem naprav ter beležimo izvajanje postopkov procesa. Struktura dinamičnega dela podatkovne baze je odvisna od vsebine statičnega dela podatkovne baze.

V tem dokumentu je predstavljen prototip podatkovne baze procesnega dela Centralne čistilne naprave v Domžalah. Konceptualni načrt podatkovne baze je opisan s podatkovnim modelom Entiteta-Razmerje (okr. E-R). Relacijska podatkovna baza je definirana z avtomatskim prevodom konceptualnega modela v SQL. Prototip je realiziran z relacijskim sistemom za delo s podatkovnimi bazami *Oracle*. V okviru izdelave prototipa podatkovne baze za shranjevanje podatkov v procesnem sistemu smo ocenili alternativne rešitve za predstavitev dinamičnega dela podatkovne baze, ki predstavlja jedro sistema za shranjevanje procesnih podatkov. Čeprav sta tako konceptualni načrt podatkovne baze kot tudi implementacija podatkovne baze prirejena specifičnim zahtevam okolja konkretne čistilne naprave, je jedro podatkovne baze definirano splošno in je namenjeno za shranjevanje podatkov v aplikacijah za vodenje zveznih procesov. Zasnova procesne podatkovne baze je nastala na osnovi izkušenj s shranjevanjem podatkov v vrsti podobnih aplikacij za vodenje zveznih procesov.

V nadaljevanju je najprej predstavljen konceptualni načrt podatkovne baze, ki je bil izdelan z uporabo podatkovnega modela E-R. V grobem so predstavljene osnovne entitete podatkovne baze. Predstavitvi konceptualnega načrta sledi v Sekciji 3 opis implementacije procesne pdoatkovne baze. Implementacija je predstavljena iz treh vidikov. Opisana je realizacija konceptualnega modela opisanega v Sekciji 2, predstavljeni so nekateri moduli sistema ter uporabniški vmesnik do podatkovn baze. V zadnji sekciji podamo zaključek.

2. Konceptualni načrt procesne podatkovne baze

Konceptualni načrt procesne podatkovne baze je definiran z uporabo podatkovnega modela Entiteta-Razmerje. Nekaterih ključnih lastnosti strukture dinamičnega dela podatkovne baze ni mogoče predstaviti z gradniki podatovnega modela E-R; v takšnih primerih so lastnosti entitet predstavljene s tekstovnim opisom. V nadaljevanju bo najprej predstavljen entitetni diagram procesne podatkovne baze. V sledečih razdelkih bodo opisane osnovne entitete statičnega dela podatkovne ter entitete s katerimi je definirana struktura dinamičnega dela podatkovne baze.



Slika 1: Konceptualni načrt podatkovne baze

2.1 Entitetni diagram

Na Sliki 1 je predstavljen konceptualni načrt procesne podatkovne baze opisan s podatkovnim modelom E-R. Statični del podatkovne baze predstavljajo entitete: PROCES, PODPROCES, GRUPA, POSTOPEK, NAPRAVA, UKAZ, PARAMETER in SIGNALIZACIJA. Dinamični del podatkovne baze opisujejo entitete SKU-SIG, ki definira skupine signalizacij in entitete TAB-SIG katera definira tabele vrednosti signalizacij

Statični del podatkovne baze

Hierarhija entitet, ki se nahaja na desni strani načrta predstavlja osnovno strukturno hierarhijo procesnega sistema. Entitete v tej hierarhiji so v večini primerov povezane z razmerjem tipa 1-N, kar pove, da je nadrejeni objekt običajno sestavljen iz množice podobjektov. Na primer, vsaka grupa je sestavljena iz ene ali več naprav. Izjema je razmerje med entitetama POSTOPEK in NAPRAVA; vsak postopek lahko vpliva na več naprav in tudi vsako napravo lahko kontrolira eden ali več postopkov.

Postopki so v podatkovni bazi predstavljeni z instancami entitete POSTOPEK. V obstoječem prototipu procesne podatkovne baze niso predstavljeni celotni postopki ampak samo vmesniki postopkov, ki so definirani z množico parametrov postopka (instance entitete PARAMETER) ter z naborom ukazov postopku (instance entitete UKAZ). Shranjevanje celotnih postopkov bi omogočalo centralen nadzor nad postopki procesa ter avtomatsko nalaganje postopkov v krmilne enote procesa.

Struktura dinamičnega dela podatkovne baze je definirana z entitatama SIGNALIZACIJA in SKU-SIG. Signalizacije predstavljajo vrednosti, s katerimi opisujemo stanje naprav, postopkov, ukazov in parameterov, ter raznovrstne meritve. Ločimo med diskretno in zvezno signalizacijo. Vrednost diskretne signalizacije se običajno spreminja relativno počasi (npr. vsake 1/2 ure). Diskretna signalizacija se uporablja za spremljanje sprememb stanja procesov in naprav. Vrednost zvezne signalizacije se spreminja v rangu sekund (npr. vsakih 20s). Običajno zvezne signalizacije predstavljajo neko meritev; na primer koncentracijo kisika v odplaki. Signalizacije so zbrane v skupine, ki opisujejo stanje nekega dela podprocesa. Posamezne skupine so definirane kot instance entitete SKU-SIG.

Dinamični del podatkovne baze

Osnovna entiteta dinamičnega dela procesne podatkovne baze je entiteta TAB-SIG, ki shranjuje vrednosti signalizacij. Z entitetnim diagramom ni mogoče definirati dejanski pomen entitete TAB-SIG. Vsaki skupini signalizacij, ki je instanca entitete SKU-SIG, ustreza ena tabela opisana z entiteto TAB-SIG. V tej tabeli se zbirajo vrednosti tistih signalizacij, ki so člani dane skupine signalizacij. Instance entitete TAB-SIG vsebujejo tri atribute: identifikar signalizacije, ki je definiran preko razmerja med entitetama TAB-SIG in SIGNALIZACIJA; vrednost signalizacije; ter čas zapisovanja vrednosti signalizacije.

Opisan načrt dinamičnega dela podatkovne baze predstavlja eno izmed rešitev za shranjevanje vrednosti signalizacij po skupinah signalizacij. V naslednji sekciji so

predstavljene različne rešitve, ki so ocenjene glede na uporabljen pomnilniški prostor, kompleksnost rešitve ter primernost tabel vrednosti signalizacij za nadaljnjo obdelavo. Prej opisana rešitev je implementirana v obtoječem prototipu.

3. Implementacija procesne podatkovne baze

Procesna podatkovna baza je implementirana z relacijskim sistemom za delo s podatkovnimi bazami. V nadaljevanju je najprej opisana implementacija statičnega in dinamičnega dela podatkovne baze. Konceptualni načrt statičnega dela podatkovne baze je preveden v relacije po običajnih pravilih za prevod E-R diagramov v relacije. Dinamični del podatkovne baze je predstavljen v obliki študije alternativnih rešitev za implementacijo tabel za shranjevanje vrednosti signalizacij.

V drugem razdelku te sekcije so predstavljeni osnovni programski moduli prototipa implementirani s programskim jezikom PL/SQL. Opisani moduli omogočajo enostavno delo s tabelami vrednosti signalizacij ter izvajanje simulacije procesnega sistema, ki služi za preverjanje delovanja sistema. V zadnjem razdelku te sekcije je predstavljen uporabniški vmesnik do procesne podatkovne sistema.

3.1 Podatkovne strukture procesne podatkovne baze

Statični del podatkovne baze je sestavljen iz tabel, ki jih dobimo s pretvorbo iz entitet in razmerij konceptualnega načrta podatkovne baze. Ta del podatkovne baze vsebuje tabele s katerimi predstavimo statično strukturo procesa. Pri definiciji statičnega dela podatkovne baze z vnosom podatkov o procesih, podprocesih, itn. definiramo strukturo dinamičnega dela podatkovne baze. Ključne entitete s katerimi je definirana vsebina dinamičnega dela podatkovne baze sta entiteti SIGNALIZACIJE in entiteta SKU-SIG s katero predstavimo različne skupine signalizacij.

Poglejmo si najprej osnovni gradnik dinamičnega dela podatkovne baze, to je *vrednost signalizacije*. Nižji nivoji procesnega sistema odčitavajo vrednosti signalizacij, ki lahko opisujejo stanje naprav, postopkov, ukazov in parameterov ali vrednost neke meritve. Opis vsake signalizacije, ki jo dobimo iz nižjih nivojev procesnega sistema mora biti shranjen v statičnem delu podatkovne baze. Opis vsake signalizacije vsebuje vse podatke, ki so potrebni za pravilno interpretacijo signalizacije ter referenco na objekt (npr. napravo) katere stanje dana signalizacija predstavlja. V primeru, da podatkovna baza mora hraniti vrednosti od nekaj sto do tisoč signalizacij ali več, količina podatkov naraste na nekaj miljard zlogov podatkov na mesec.

Signalizacije lahko pogosto povežemo v skupine na osnovi nekega kriterija. Kriterij za definicijo skupine signalizacij je lahko pripadnost k nekem fizičnem ali logičnem objektu, kot so na primer fizični prostor, naprava, podproces, postopek in podobno. Nadalje, skupine signalizacij so lahko izbrane tudi tako, da vsebujejo samo tiste signalizacije, ki so uporabljene za določen pregled delovanja procesa.

Cilji pri definiciji podatkovne strukture za shranjevanje vrednosti signalizacij je: dobra izraba pomnilniškega prostora; čim manjša kompleksnost programov, ki vstavljajo in berejo podatke; ter možnost hitre pretvorbe v tabele, ki so primerne za

nadaljnjo obdelavo. Pri snovanju strukture procesne podatkovne baze smo ocenili naslednji dve podatkovni strukturi za predstavitev vrednosti skupin signalizacij.

Prvič, vrednosti signalizacij lahko hranimo v tabelah, katerih vrstica predstavlja vrednost ene same signalizacije ob določenem času. Takšne tabele bomo imenovali dogodkovne tabele. Stolpci tabel morajo vsebovati identifikacijo, vrednost in čas branja vrednosti signalizacije. Ta rešitev je uporabljena v obstoječem prototipu.

Čas	Sig-Ident	Sig-Val

Drugič, vrednosti signalizacij so lahko shranjene v tabelah, katerih struktura je odvisna od množice signalizacij, ki jih tabela shranjuje. Prvi stolpec tabele vedno predstavlja čas branja vrednosti signalizacije. Preostali stolpci tabele hranijo vrenosti signalizacij. Takšne tabele bomo v nadaljevanju imenovali ravninske tabele.

Čas	Sig1-Val	Sig2-Val	Sig3-Val	…

Dogodkovne tabele

Prva rešitev za shranjevanje vrednosti signalizacij temelji na uporabi tabel, ki imajo fiksno tristolpčno strukturo. Poglejmo si najprej lastnosti dogodkovne tabele brez da bi definirali kako se bodo vrednosti različnih signalizacij porazdelile po večih tabelah. Pozitivne lastnosti uporabe dogodkovnih tabel so naslednje. Prvič, struktura tabel je uniformna in enostavna; vse tabele s katerimi se shranjujejo vrednosti signalizacij imajo enako tristolpčno strukturo. Drugič, tabele lahko shranjujejo vrednosti določene množice signalizacij. Ob spremembi sestave množice signalizacij ni potrebna sprememba strukture tabele. Tretjič, tabele, ki predstavljajo vrednosti signalizacij več različnih množic signalizacij, lahko enostavno združujemo. Nove tabele imajo spet enako strukturo kot originalne tabele.

Med osnovne negativne lastnosti te rešitve štejemo potrebo po pretvarjanju tabele v drugo obliko pred nadaljnjo obdelavo. Čas pretvarjanja tabele, ki vsebuje nekaj 10.000 vrednosti signalizacij je nekaj sekund na SQL strežniku sistema Oracle, ki teče pod operacijskim sistemom Windows NT.

Poglejmo še na kašne načine lahko organiziramo vrednosti različnih signalizacij po večih tabelah. Prvič, najenostavnejša rešitev je uporaba ene same tabele, ki vsebuje vrednosti vseh signalizacij. Dobra stran rešitve je enostavnost, saj je potrebno skrbeti za eno samo tabelo; rešitev je torej primerna za shranjevanje dinamičnih podatkov v manjših procesnih sistemih. Slabe strani rešitve so naslednje. Najprej, velikost tabele bo zelo hitro naraščala. Drugič, vrednosti signalizacij, ki pripadajo določeni skupini signalizacij je potrebno izbrati iz tabele programsko.

Drugi način realizacije entitete TAB-SIG, ki je trenutno uporabljen v prototipu procesne podatkovne baze, je definicija večih tabel; vsaka tabela shranjuje vrednosti signalizacij, ki pripadajo natančno določeni skupini signalizacij (instanca entitete SKU-SIG). Ločitev vrednosti signalizacij po skupinah signalizacij nam na nek način omogoča kontrolo nad velikostjo tabel ter logično razdeli vrednosti signalizacij na osnovi strukture celotnega procesa. Slabi strani rešitve sta večja kompleksnost programov za delo s tabelami ter možno podvajanje vrednosti signalizacij v podatkovni bazi.

Tretja rešitev, ki smo jo obravnavali pri izboru podatkovnih struktur za predstavitev dinamičnega dela procesne podatkovne baze je definicija dveh nivojev grupiranja vrednosti signalizacij. Fizični nivo grupiranja vsebuje skupine signalizacij, ki so določene glede na čas branja signalizacij ter fizično mesto, kjer se generirajo vrednosti signalizacij. Logični nivo grupiranja signalizacij je določen glede na logično strukturo procesa. Slaba stran te rešitve je predvsem povečana kompleksnost strukture podatkovne baze, kot tudi programov za obdelavo vrednosti signalizacij. Ločitve med fizično in logično organizacijo shranjevanja vrednosti signalizacij bi bila smiselna le pri kompleksnih zveznih procesih kjer bi z uvedbo fizičnega nivoja grupiranja signalizacij poenostavili zbiranje vrednosti signalizacij.

Ravninske tabele

Alternativni način shranjevanja vrednosti signalizacij je uporaba ravninskih tabel. Enako kot v primeru dogodkovnih tabel si najprej oglejmo lastnosti ravninskih tabel brez da bi definirali kako bodo signalizacije porazdeljene po večih tabelah. Pozitivni aspekti te rešitve so naslednji. Prvič, vrednosti signalizacij se nahajajo v takšni obliki, da jih lahko uporabljamo za nadaljnjo analizo brez predhodne pretvorbe podatkov v kakšno drugo obliko. Drugič, rešitev je primerna za shranjevanje množice vrednosti signalizacij, ki se odčitujejo ekvidistančno.

Poglejmo si še negativne aspekte rešitve. Vrstica ravninske tabele vsebuje vrednosti množice signalizacij ob določenem času. Praviloma bo tabela vsebovala veliko redundantnih podatkov, kar lahko vpliva tudi na hitro naraščanje velikosti tabel. Naprej, v primerjavi z dogodkovno tabelo je struktura ravninskih tabel bolj kompleksna. Struktura tabel je določena ob kreiranju tabel. Spreminjanje sestave skupine signalizacij, ki se shranjujejo v tabelah zahteva spremembo strukture tabele. Na primer, ob dodajanju nove signalizacije v skupino je potrebno tabelo razširiti z novim stolpcem. Še več, zaradi spremembe strukture tabele bo najverjetneje potrebno spreminjati tudi programe, ki tabelo uporabljajo.

Z ravninskimi tabelami lahko delamo na dva načina. Prvič, predpostavimo, da bo struktura tabel ostala fiksna in programe, ki delajo s tabelami vežemo na to strukturo. V tem primeru sprememba strukture tabele zahteva popravljanje obstoječih programov za delo s tabelami. Drugič, za delo z ravninskimi tabelami lahko uporabljamo dinamični SQL, ki omogoča pisanje SQL stavkov brez da bi v času pisanja programov poznali natančno strukturo tabel. Programi napisani z uporabo dinamičnega SQL so praviloma kompleksni in pogosto je implementacija dinamičnega SQL vezana na sistem za delo s podatkovnimi bazami.

Podobno kot pri uporabi dogodkovnih tabel lahko tudi zdaj razdelimo signalizacije po tabelah na več načinov. Najprej, vse signalizacije so lahko predstavljene v eni sami tabeli. Pozitivna lastnost te rešitve je enostavnost saj imamo vrednost vseh signalizacij v eni sami tabeli. Ta rešitev se lahko uporablja v primeru, da procesni sistem vsebuje majhno število različnih signalizacij. Med negativne lastnosti te rešitve uvrščamo predvsem redundantnost podatkov v tabeli ter s tem povezano hitro naraščanje velikosti tabele.

Drugič, vsako skupino signalizacij lahko predstavimo z eno samo tabelo. Pozitivne in negativne lastnosti te rešitve so zajete pri opisu splošnih lastnosti predstavitve vrednosti signalizacij z ravninskimi tabelami. Med pozitivne aspekte rešitve sodi predvsem to, da tabele za nadaljnjo obdelavo ni potrebno pretvarjati v drugo obliko. Bistvena negativna lastnost rešitve je togost: ob spremembi vsebine skupin signalizacij je potrebno spreminjati strukturo tabel kot tudi programe, ki delajo s temi tabelami.

In še zadnja možnost; signalizacije lahko razvrstimo v skupine po dveh kriteriji. Z logičnim nivojem grupiranja signalizacij definiramo skupine signalizacij, ki so določene glede na uporabo vrednosti signalizacij za nadaljnje obdelave. Fizični nivo grupiranja signalizacij je prilagojen fizičnim lastnostim signalizacij, to sta predvsem mesto in čas zbiranja signalizacij. Ta rešitev bi zaradi dodatnega nivoja grupiranja vrednosti signalizacij povzročila večjo kompleksnost programov. Enako kot v primeru uporabe dogodkovnih tabel, bi bila ta rešitev smiselna le za predstavitev podatkov v kompleksnih procesnih sistemih, kjer so deli procesa fizično ločeni zaradi velikosti celotnega procesa.

3.2 Osnovni programski moduli sistema

Med osnovne programske module sistema uvrščamo: modul SSGTAB za delo s tabelami vrednosti signalizacij, modul DYNTAB za delo z ravninskimi tabelami signalizacij ter modul SIMUL za simulacijo polnenja podatkovne baze med izvajanjem procesa. Vsi moduli so izdelani v programskem jeziku PL\SQL, ki vsebuje običajne gradnike višjih programskih jezikov ter nudi enostaven dostop do podatkovnih baz z uporabo SQL.

Modul SSGTAB

Vrednosti signalizacij se shranjujejo v tabelah, katerih statična struktura je določena z entiteto TAB-SIG. Ena tabela vrednosti signalizacij ustreza eni skupini signalizacij, ki določa signalizacije katerih vrednosti se shranjujejo v tej tabeli. Ime tabele signalizacij posamezne skupine je shranjeno skupaj z opisom skupine - instanco entitete SKU-SIG. Število aktivnih tabel vrednosti signalizacij je torej odvisno od števila definiranih skupin signalizacij.

Ker so tabele vrednosti signalizacij odvisne od definicije skupin signalizacij ni mogoče vnaprej definirati statičnega vmesnika za pregledovanje in popravljanje tabel. Modul SSGTAB vsebuje množico podprogramov, ki omogočajo enostavno izvedbo vmesnika za delo s tabelami vrednosti signalizacij. Vmesnik je definiran na osnovi referenčne tabele, ki služi kot ogrodje za implementacijo zaslona z uporabo orodja

Oracle Forms. Uporabo referenčne tabele kot ogrodje za realizacijo vmesnika do tabel vrednosti signalizacij omogoča enotna struktura dogodkovnih tabel.

Modul DYNTAB

Modul DYNTAB vsebuje podprograme za pretvarjanje tabel vrednosti signalizacij v obliko, ki je primerna za nadaljnjo obdelovo s statističnimi programi. Vsaka tabela vrednosti signalizacij shranjuje vrednosti tistih signalizacij, ki pripadajo določeni skupini signalizacij. Struktura vseh tabel vrednosti signalizacij je uniformna; definirana je s tremi stolpci, ki hranijo identifikacijo, vrednost in čas generiranja vrednosti signalizacije.

V splošnem podprogrami modula DYNTAB pretvorijo tristolpčno tabelo vrednosti signalizacij v ravninsko tabelo, ki vsebuje vrednosti različnih signalizacij v različnih stolpcih tabele. Prvi stolpec nastale tabele hrani čas, vsi nadaljnji stolpci pa hranijo vrednosti posameznih signalizacije.

Ravninska tabela je lahko organizirana po naslednjih kriterijih. Najprej, ravninska tabela je lahko organizirana dogodkovno; v tabeli urejeni po času predstavlja vsaka vrstica tabele odčitavanje vrednosti vsaj ene signalizacije (stolpca) oz. odraža vsaj eno vrstico originalne tabele vrednosti signalizacij. Drugič, ravninska tabela je lahko organizirana ekvidistančno; vrstice tabele vsebujejo vrednosti signalizacij v urejene po enakomernih časovnih razmikih. Tretjič, ravninske tabele lahko hranijo vrednosti signalizacij, ki pripadajo bodisi eni sami skupini signalizacij ali več različnim skupinam signalizacij.

Modul SIMUL

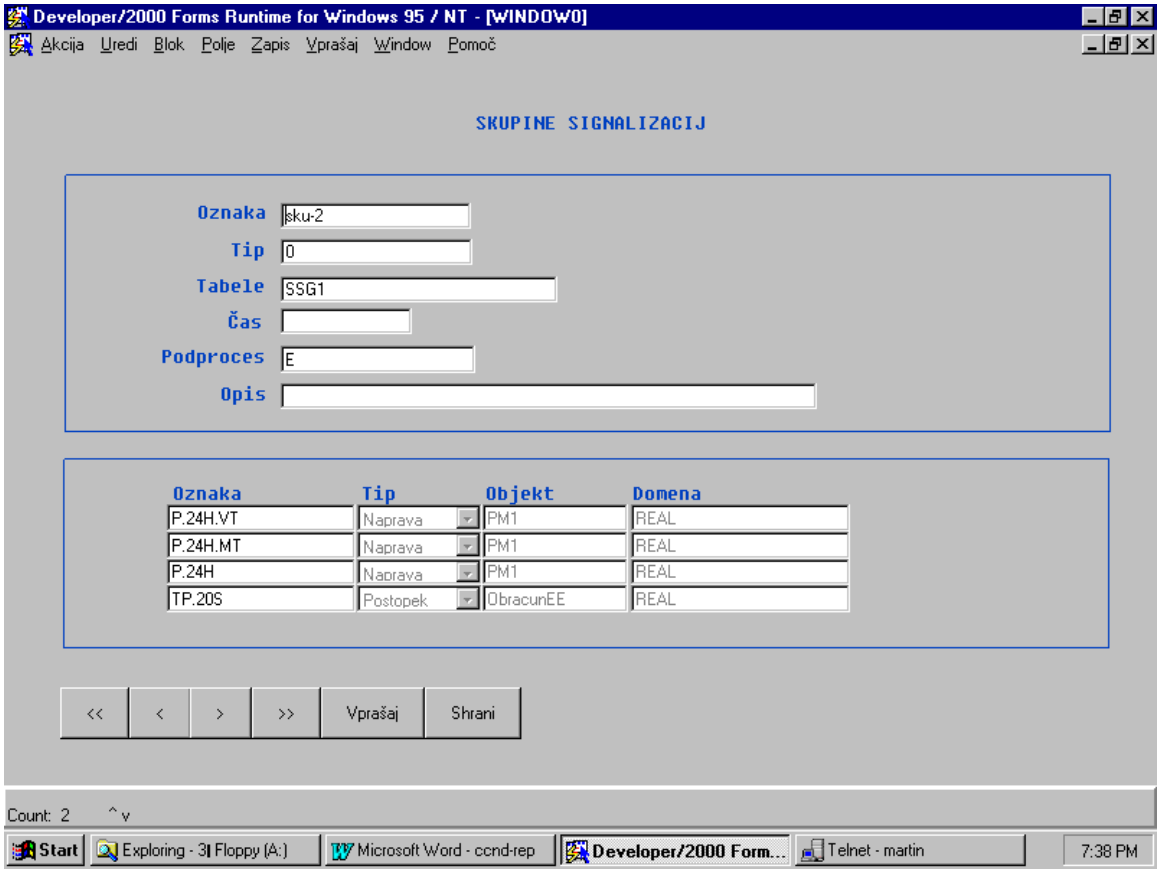
Modul SIMUL vsebuje podprograme za simulacijo polnenja dinamičnega dela podatkovne baze z vrednostmi signalizacij, ki jih generira procesni del čistilne naprave. Modul generira vrednosti signalizacij v predhodno določenih časovnih razmikih, ki so za posamezno signalizacijo podani z vrednostjo atributa SIG_FREQ tabele SIGNALIZACIJE. Vrednost atributa SIG_FREQ definira čas (v sekundah) v katerem se v povprečju generira vrednost signalizacije.

Modil simulira polenenje podatkovne baze v naprej določeno število sekund procesnega časa. Vsako sekundo program določi signalizacije katerih vrednosti je ob tem času potrebno generirati. Vrednosti izbranih signalizacij, ki so določene z uporabo generatorja naključnih števil, se uvrstijo v podatkovno bazo. V primeru, da v bazi hranimo 10 signalizacij v treh skupinah, katerih vrednosti se v povprečju generirajo vsakih 5 sekund, simulacijski program, ki simulira 10000 realnih sekund procesnega časa teče na strežniku podatkovne baze približno 150 sekund.

3.4 Uporabniški vmesnik

Uporabniški vmesnik do procesne podatkovne baze je realiziran z orodjem *Oracle Forms*. To orodje vsebuje podsistem za grafično načrtovanje uporabniškega vmesnika, obsežno knjižnico podprogramov za izdelavo uporabniških vmesnikov do podatkovne baze, ter prevajalnik programskega jezika PL/SQL. V PL/SQL je

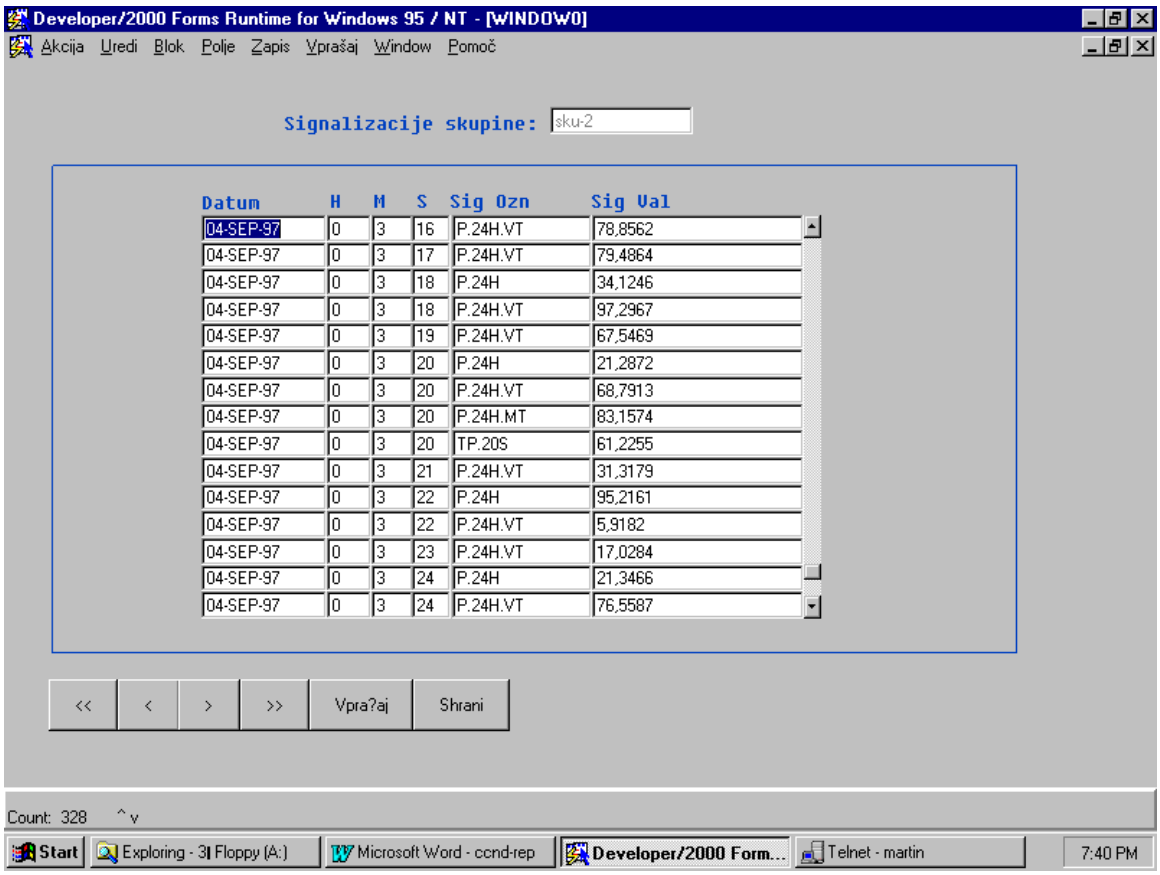
realizirana koda za implementacijo kompleksnih integritetnih omejitev relacij ter izvedbo kompleksnih zaslonov za vnos podatkov.



Slika 2: Ažuriranje skupin signalizacij

Uporabniški vmesnik do procesne podatkovne baze je organiziran z menuji, ki omogočajo proženje zaslonov za delo s tabelami podatkovne baze. Zasloni so organizirani v skupine; med osnovne skupne zaslonov uvrščamo: zaslone za definicijo in ažuriranje statičnega dela podatkovne baze; zaslone za pregledovanje in ažuriranje dinamičnega dela podatkovne baze; zaslone s katerimi prožimo programe za pretvorbo dinamičnih tabel podatkovne baze v tabele, ki se lahko obdelujejo s statističnimi in drugimi programi; ter zaslone s katerimi prožimo simulacijo izvajanja procesnega sistema.

Poleg zaslonskega vmesnika za delo s procesno podatkovno baze je omogočen tudi dostop do podatkov preko standardnega vmesnika ODBC. S pomočjo tega vmesnika je omogočena izvedba dela celotnega procesnega sistema z drugimi programski paketi. Statistične obdelave dinamičnih podatkov lahko tako realiziramo s pomočjo programskega orodja *Excel* ter podobnimi programskimi orodji.



Slika 3: Ažuriranje tabele vrednosti signalizacij

4. Zaključek

V članku smo predstavili arhitekturo sistema za delo s podatkovnimi bazami, ki hranijo podatke za vodenje in spremljanje zveznih industrijskih procesov. Podatkovna baza je sestavljena iz dveh delov: statičnega dela, kjer so shranjeni podatki o sestavi in lastnosti procesa, ter dinamičnega dela, ki shranjuje podatke, ki nastajajo med izvajanjem zveznega procesa. V prispevku smo predstavili študijo s katero so ocenjene alternativne rešitve za definicijo dinamičnega dela podatkovne baze, ki predstavlja jedro procesne podatkovne baze. Predstavljena študija lahko služi kot pomoč pri izboru podatkovnih struktur za predstavitev dinamičnih podatkov v sorodnih aplikacijah.

Literatura

[1] T. Mohorič. *Načrtovanje relacijskih podatkovnih baz.* Fakulteta za računalništvo in informatiko, Univerza v Ljubljani, 1997.

[2] PL/SQL User's Guide and Reference, Version 2, *Oracle* Documentation, 1992

[3] I. Savnik, M. Šubelj, G. Godena, M. Ostroveršnik. *Prototip podatkovne baze procesnega dela Čistilne naprave v Domžalah.* Tehnično poročilo CSD-TR-97-9, Odsek za računalniške sisteme, Institut Jožef Stefan, 1997.