

UNIVERZA V NOVI GORICI
POSLOVNO-TEHNIŠKA FAKULTETA

DIPLOMSKA NALOGA

**RAZVOJ PROGRAMSKE PODPORE ZA SPREMLJANJE DELOVNIH
NALOG V PROIZVODNEM PODJETJU**

Andrej Nuncija

Mentor: doc. dr. Bogdan Filipič

Nova Gorica, 2006

ZAHVALA

Za posredovano znanje in koristne informacije pri nastajanju diplomskega dela se zahvaljujem predavateljem na Poslovno-tehniški fakulteti Univerze v Novi Gorici in sodelavcem v podjetju Plama-pur d.d. Posebej pa se zahvaljujem mentorju doc. dr. Bogdanu Filipiču za spodbudo, nasvete in pregled celotnega dela.

IZVLEČEK

V diplomski nalogi je predstavljeno delo pri zasnovi in implementaciji programske podpore za spremljanje delovnih nalog v podjetju Plama-pur d.d. Opisan je razvoj novega programa Naloge, ki je obsegal prilagoditev obstoječe podatkovne zbirke zmogljivejšemu orodju za upravljanje podatkovnih zbirk Microsoft SQL Server in izdelavo uporabniškega vmesnika z razvojnim okoljem Borland Delphi. Tako je program pridobil povezavo s skupnimi podatkovnimi zbirkami podjetja, večuporabniško delovanje in prijaznejši uporabniški vmesnik. Predvidena je tudi možnost spletnega dostopa do podatkov o delovnih nalogah s pomočjo tehnologije Microsoft ASP.NET.

ABSTRACT

This diploma thesis presents the design and implementation of software support for monitoring work assignments in Plama-pur d.d. company. It describes the development of a new program Naloge which consisted of adjusting the existing database to a more efficient database management tool Microsoft SQL Server, and implementing the user interface with Borland Delphi programming environment. As a result, the developed program acquired a connection with common databases of the company, multi-user operation capability and a friendly user interface. We foresee an additional feature of web access to work assignment data through the Microsoft ASP.NET technology.

KLJUČNE BESEDE

programska oprema, relacijska podatkovna zbirka, uporabniški vmesnik, informacijski sistem

KEY WORDS

software, relational database, user interface, information system

KAZALO

1. Uvod.....	1
1.1. Problematika delovnih nalog	1
1.2. Uporabljen programski orodja	3
1.3. Razvoj novega programa Naloga	3
1.4. Rezultati.....	4
2. Analiza stanja.....	5
2.1. Podjetje Plama-pur d.d.	5
2.1.1. Poslovna odličnost, kakovost, varstvo okolja.....	5
2.1.2. Organiziranost podjetja in proizvodno-prodajni program	6
2.2. Informacijski sistem podjetja.....	7
2.3. Obstoječi program Naloga.....	8
2.3.1. Uporabniški vmesnik	8
2.3.2. Podatkovna zbirka.....	12
3. Uporabljen metodologija in programski orodja	13
3.1. Relacijske zbirke podatkov.....	13
3.2. Sistem za upravljanje podatkovnih zbirk Microsoft SQL Server.....	14
3.2.1. Upravljanje strežnika	14
3.2.2. Arhitektura podatkovnih zbirk.....	15
3.3. Razvojno programsko okolje Borland Delphi.....	16
4. Razvoj novega programa Naloga.....	21

4.1. Podatkovna zbirka	21
4.1.1. Uvoz podatkov	21
4.1.2. Razvoj podatkovne zbirke	22
4.1.3. Uporabniki in pravice dostopa do podatkovne zbirke	24
4.1.4. Arhiviranje podatkov	25
4.2. Uporabniški vmesnik.....	25
4.2.1. Glavni meni.....	26
4.2.2. Obrazci šifrantov.....	27
4.2.3. Obrazec Seznam in vnos nalog.....	30
4.2.4. Obrazec za knjiženje delovnih nalog.....	32
4.2.5. Poročilo o delovnih nalogah	33
5. Umestitev novega programa Naloge v informacijski sistem podjetja	35
5.1. Testiranje programa.....	35
5.2. Uporabniška dokumentacija	35
5.3. Uporaba programa in izkušnje uporabnikov	36
5.4. Interakcija z informacijskim sistemom podjetja.....	38
5.5. Učinki programa	38
6. Zaključek	40
6.1. Opravljeno delo	40
6.2. Izkušnje z razvojnimi orodji	40
6.3. Uporabnost v drugih podjetjih.....	41

6.4. Nadaljnje delo.....	41
7. Literatura.....	43

KAZALO SLIK

Slika 1: Seznam interno izdelanih programov podjetja Plama-pur d.d.	8
Slika 2: Ekranska slika glavnega menija programa Naloge	9
Slika 3: Ekranska slika vnosa podatkov programa Naloge.....	9
Slika 4: Ekranska slika pregleda nalog v programu Naloge.....	10
Slika 5: Ekranska slika šifrantov v programu Naloge	11
Slika 6: Ekranska slika šifranta odgovornih oseb v programu Naloge.....	11
Slika 7: Entitetno-relacijski model podatkovne zbirke Naloge	12
Slika 8: Ekranska slika Enterprise Managerja	14
Slika 9: Ekranska slika poizvedbe v SQL Server Query Manager	15
Slika 10: Ekranska slika razvojnega okolja Borland Delphi	17
Slika 11: Ekranska slika razvojnega okolja s pogledom na izvorno kodo.....	17
Slika 12: Ekranska slika razvojnega okolja s pogledom na vizualne gradnike	18
Slika 13: Ekranska slika orodja Object Inspector.....	19
Slika 14: Funkcija Data Transformation Services v orodju Enterprise Manager	22
Slika 15: Entitetno-relacijski model podatkovne zbirke novega programa Naloge ..	23
Slika 16: Logotip interno izdelanih programov podjetja Plama-pur d.d.	26
Slika 17: Obrazec za prijavo uporabnika v novi program Naloge.....	26
Slika 18: Ekranska slika glavnega menija novega programa Naloge.....	27
Slika 19: Ekranska slika obrazca za pregled virov	27
Slika 20: Ekranska slika obrazca za ažuriranje virov	28

Slika 21: Ekranska slika obrazca za pregled tematskih področij.....	29
Slika 22: Ekranska slika obrazca za pregled odgovornih oseb.....	30
Slika 23: Ekranska slika obrazca Seznam in vnos nalog	31
Slika 24: Ekranska slika obrazca za popravljanje delovnih nalog.....	32
Slika 25: Ekranska slika obrazca za dodajanje delovnih nalog	33
Slika 26: Ekranska slika poročila o delovnih naloga.....	34
Slika 27: Ekranska slika obrazca seznam in vnos z iskanjem po odgovorni osebi ...	37
Slika 28: Ekranska slika obrazca seznam in vnos z iskanjem po vsebini naziva	37

1. UVOD

Diplomska naloga obsega zasnovano in implementacijo programske podpore za spremljanje delovnih nalog v podjetju Plama-pur d.d. Podjetje se nahaja v Podgradu v bližini Ilirske Bistrice. Sedanji računalniški program Naloga, ki teče na enem računalniku in temelji na orodju za delo s podatkovnimi bazami Microsoft Access, bomo zasnovali na novo in razširili njegove zmožnosti. Podatke o delovnih nalogah bomo prenesli v zmogljivejši sistem za upravljanje baz podatkov Microsoft SQL Server, določili pravice uporabnikov za vnos in pregledovanje stanja delovnih nalog, za razvoj programske podpore uporabili okolje Delphi ter predvideli možnost spletnega dostopa do podatkov o nalogah s pomočjo tehnologije Microsoft ASP.NET. Prav tako bomo proučili možnost uporabe razvite programske opreme v drugih organizacijah.

Glavni cilj diplomske naloge je zasnovati in razviti programsko podporo za spremljanje delovnih nalog v podjetju Plama-pur d.d. ter jo preizkusiti do te mere, da bo uporabna za delo v podjetju. S prehodom na sistem za upravljanje s podatki Microsoft SQL Server bomo pridobili večuporabniško delovanje ter zanesljivost sistema. V prihodnosti pa bo mogoče pregledovati podatke o delovnih nalogah tudi preko spletnega vmesnika s pomočjo tehnologije Microsoft ASP.NET.

1.1. Problematika delovnih nalog

Delovne naloge v podjetju Plama-pur d.d. so navadno sklepi poslovnega kolegija, tima kakovosti, tima za reklamacije ali uprave družbe. Ti organi si za določeno problematiko oziroma problem določijo končno ali želeno stanje in odgovorno osebo. Odgovorni osebi določijo ponavadi tudi sodelujoče osebe, ki ji pri izvajanju delovne naloge pomagajo. Na izbiro odgovorne osebe vpliva predvsem področje, s katerega določen problem izvira ali pa je na njem potrebno nekaj ukreniti. Podobno so izbrane tudi sodelujoče osebe.

V podjetju Plama-pur d.d. imajo tudi osebo, ki je zadolžena za nadzor in spremljanje delovnih nalog. Ta je odgovorna tudi za delegiranje glede poteka izvedbe in nadzora časovnih rokov posameznih nalog. Le-ta odgovorni osebi preda delovno nalogo s časovnim načrtom izvedbe. Odgovorna oseba problem preuči in poišče najboljši

način za rešitev problema. Nato poskuša rešitev izpeljati, pri tem pa ji pomagajo tudi sodelujoče osebe. Ko je naloga zaključena, poda poročilo z opisom rešitve, ki ga pregleda oseba, odgovorna za nadzor in spremljanje nalog. Če je naloga zaključena, poročilo posreduje organu, ki je delovno nalogo podal.

V primeru, da je naloga zaključena, lahko organ, ki je delovno nalogo podal oziroma prosil za rešitev problema, na podlagi rezultatov, ki so jih podali odgovorna oseba in drugi sodelujoči, ustvari še druge naloge ali pa le poda predlog za določen organizacijski predpis ali delovno navodilo. Podobno se lahko odloči tudi v primeru, ko delovna naloga ne doseže zastavljenega cilja in bi bile za rešitev potrebne dolgoročne aktivnosti. Določenih nalog pa tudi ni mogoče pripeljati do cilja. Večinoma so to naloge s tehničnega področja, na primer izdelava novega tipa izdelka z določenimi lastnostmi.

Tako v sedanjem kot v novem programu za spremljanje delovnih nalog je tok dokumentov zelo podoben. Eden izmed organov družbe zastavi novo nalogo oziroma zaprosi za rešitev neke problematike ter določi tudi odgovorno in sodelujoče osebe. Vse to oseba, odgovorna za nadzor in spremljanje nalog, vpiše v program in posredno tudi v podatkovno zbirko. Izpiše se tudi poročilo, ki ga prejme odgovorna oseba. Ko odgovorna oseba pride do rešitve naloge, poročilo z opisom rešitve preda osebi za nadzor in spremljanje nalog, ki podatke vpiše v program. Poročilo z vsemi podatki se nato vrne organu, ki je delovno nalogo podal.

Zaradi velikega števila nalog in obsega podatkov o delovnih nalogah so se v podjetju že pred nekaj časa odločili, da je za spremljanje nalog potrebna programska oprema. Z njeno pomočjo bi izdelali evidenco nalog in tako povečali preglednost na tem področju. Zato so kupili računalniški program Naloge, namenjen vodenju evidence o nalogah. Z njim so vpisovali podatke o posamezni nalogi: številko naloge, naziv, cilj, ime odgovorne in sodelujočih oseb, prioriteto itd. Program Naloge temelji na orodju za podatkovne zbirke Microsoft Access. To orodje pa ne podpira večuporabniškega delovanja, kar se je v praksi izkazalo za veliko pomanjkljivost. Ker je imela dostop do podatkov o delovnih nalogah le oseba, ki je odgovorna za nadzor in spremljanje nalog, drugi udeleženci niso imeli vpogleda v evidenco nalog. Z novim programom bodo imeli vpogled v evidenco tudi drugi udeleženci. Prav tako v obstoječi program

ni bilo mogoče vključiti globalnih šifrantov podjetja, kot je šifrant delavcev. Tudi ta možnost bo vključena v novi program. S tem se bomo izognili zamudnemu vpisovanju v šifrante programa Naloge, podatki pa se bodo osveževali glede na globalne šifrante podjetja.

1.2. Uporabljeni programska orodja

Pri razvoju programske podpore za spremljanje delovnih nalog smo uporabljali sistem za upravljanje podatkovnih baz Microsoft SQL Server in orodje za programiranje Delphi.

Microsoft SQL Server je zmogljiv strežnik za zbirke podatkov, ki z integriranimi orodji zagotavlja prvovrstno upravljanje podatkov. Mehanizem v SQL Serverju varno in zanesljivo shranjuje tako relacijske kot strukturirane podatke in tako omogoča upravljanje podatkovnih aplikacij. Jezik Microsoft SQL je prav zaradi svoje vloge upravljanja s podatki ključen del sistema Microsoft SQL Server. Vsebuje funkcije analize, poročanja, integracije in obveščanja.

Delphi je visoko zmogljivo vizualno okolje za programiranje v okolju Windows. V podjetju so se za izbor tega orodja odločili, ko so ocenili razmerje med njegovo zmogljivostjo in enostavnostjo. Izbrali so ga predvsem zaradi zagotovitve skladnosti z obstoječo programsko opremo in informacijskim sistemom v podjetju. Za pisanje programske kode v Delphiju se uporablja jezik predmetni pascal. Enostaven program s tem orodjem lahko izdelamo skoraj brez vsakih programerskih izkušenj, z malo več znanja pa hitro izdelamo tudi profesionalno aplikacijo.

1.3. Razvoj novega programa Naloge

Pri razvoju novega programa Naloge se bomo posvetili dvema sestavinama, ki sta ključni za njegovo delovanje, to sta to podatkovni del in programski del. V zvezi s podatkovnim delom bo podrobneje opisana podatkovna zbirka in njen razvoj. Prav tako se bomo posvetili selitvi podatkov iz prvotnega programa Naloge v strežnik Microsoft SQL Server ter arhiviranju podatkov. Posvetili se bomo tudi uporabnikom in njihovim pravicam dostopa do podatkovne zbirke.

V okviru programskega dela bomo podrobneje opisali menije, posamezne obrazce in njihovo delovanje ter poročilo o delovnih nalogah.

1.4. Rezultati

Novi program Naloge bo omogočal vnos in pregledovanje podatkov o delovnih nalogah. Ker bo deloval na osnovi strežnika Microsoft SQL Server, bo omogočal večuporabniško delovanje in uporabo skupnih šifrantov podjetja. Prav tako pa bo ta strežnik osnova za uporabo spletne tehnologije Microsoft ASP.NET. Program bomo izpopolnili do te mere, da bo primeren za redno uporabo v podjetju.

2. ANALIZA STANJA

Zaradi problematike delovnih nalog in razvoja novega programa je potrebno boljše poznavanje okolja, v katerem bo program deloval, zato bomo v tem poglavju najprej predstavili podjetje Plama-pur d.d., nato pa opisali njihov informacijski sistem ter obstoječi program Naloge za spremljanje delovnih nalog.

2.1. Podjetje Plama-pur d.d.

Plama-pur d.d. je proizvajalec in predelovalec mehkih poliuretanskih pen. Vgrajene v avtomobilskih sedežih, oblazinjenem pohištvu, v čevljih in oblačilih ali kot čistilne gobice nam te pene omogočajo udobje, toplino in zaščito. Osnovna izdelka podjetja Plama-pur d.d. sta polieterska in poliestrska pena z različnimi fizikalno-kemijskimi lastnostmi. Izdelujejo ju v obliki blokov, folij, plošč in oblikovanih izdelkov. Plama-pur d.d., tovarna mehke poliuretanske pene, ima 50 letno tradicijo. Izdelke iz mehke poliuretanske pene prodajajo v 14 držav. Uspešno sodelujejo s številnimi partnerji v svetu in z njimi razvijamo dolgoročne odnose (Plama-pur d.d., 2006).

Podjetje je organizirano kot samostojna in enovita delniška družba. V tovarni, ki stoji v Podgradu, približno 20 km od slovensko-italijanske meje, je zaposlenih okoli 250 delavcev (podatek velja za prvo polovico leta 2005), ki delajo na na eni lokaciji.

Na trgu široke potrošnje je podjetje Plama-pur d.d. znana po čistilnih gobicah, med odjemalci industrijske porabe pa po široki paleti mehkih poliuretanskih pen. Svoje zmogljivosti usmerjajo v razvoj in izdelavo različnih tipov mehke poliuretanske pene po kontinuiranem postopku, v zahteven razrez in obdelavo pene za najrazličnejše namene. Kupcem želijo omogočiti udobje, ki ga s svojo mehko in fizikalno kemijskimi lastnostmi nudijo mehke poliuretanske pene. Cilj podjetja je postati pomemben dobavitelj mehke poliuretanske pene z različnimi fizikalno kemijskimi lastnostmi sestavljalcem avtomobilskih delov na evropskem trgu.

2.1.1. Poslovna odličnost, kakovost, varstvo okolja

V podjetju Plama-pur d.d. so sistemsko delo in pravilnost svoje vzpenjajoče se poti na področju zagotavljanja kakovosti svojih izdelkov in storitev potrdili v letu 1997 s

pridobitvijo certifikata skladnosti standardu ISO 9001 ter leta 2000 s pridobitvijo certifikata skladnosti avtomobilskemu standardu zagotavljanja kakovosti VDA 6.1. Vzporedno so naredili tudi nekaj samoocen po modelu Priznanja Republike Slovenije za poslovno odličnost in zadostili zahtevam standarda ISO TS 16949. Vedno pa poudarjajo svoj osnovni cilj, to je popolnoma zadostiti zahtevam kupcev .

Plama-pur d.d. je podjetje, ki je vpeto v krog pašnikov, travnikov, njiv in gozdov. Zato je interes podjetja in vseh zaposlenih v podjetju ohraniti ta biotop takšen, kot je bil že tisočletja. To pa zahteva tudi večja vlaganja na področja, ki prinašajo rezultat predvsem v okolju. Uporabo CFC, ki so bili sestavni del proizvodnje pene v preteklosti, so ukinili že leta 1995, postopoma pa ukinjajo tudi nekatere druge sestavine, ki so manj prijazne do okolja.

2.1.2. Organiziranost podjetja in proizvodno-prodajni program

Podjetje je organizirano kot samostojna in enovita delniška družba. Poslovni sistem podjetja Plama-pur d.d. je organiziran v poslovne procese, ki predstavljajo tudi jedro izdelčnih skupin:

- razvoj in izdelava mehke poliuretanske pene po kontinuiranem postopku,
- razrez in obdelava mehke poliuretanske pene v obliki,
- razrez in obdelava kratkih blokov,
- izdelava čistilnih gobic,
- izdelava folij,
- izdelava oblikovanih izdelkov.

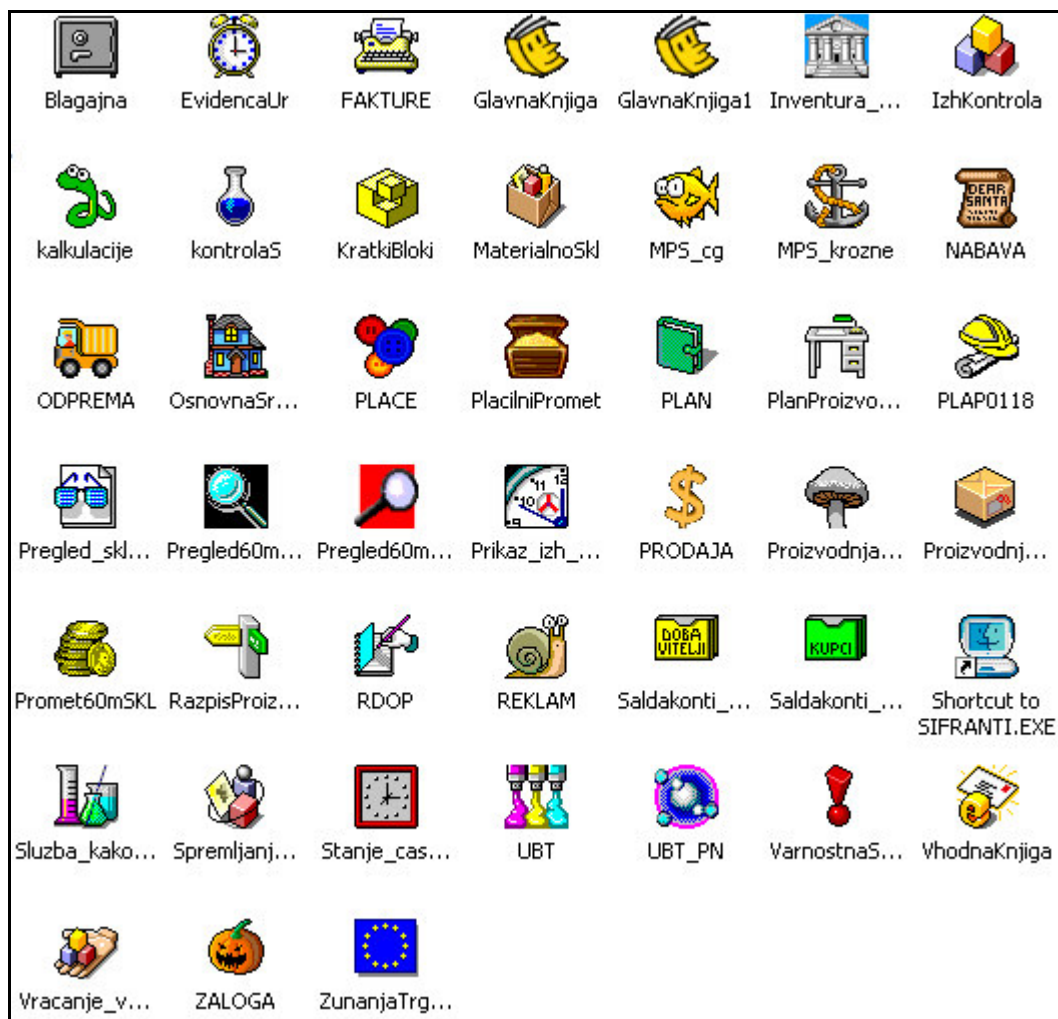
Proizvodno-prodajni program podjetja Plama-pur d.d. predstavljata dve temeljni skupini mehkih poliuretanskih pen: polieterske pene Puhin (75%) in polieterske pene Ipen (25%). Številke v oklepajih pomenijo delež prodaje posamezne skupine v skupni prodaji podjetja. Glede na zahteve kupcev lahko družba danes ponudi 100 tipov pen, ki se razlikujejo po fizikalno-kemijskih lastnostih. Pene ponujajo v različnih oblikah in dimenzijah. Osnovne izdelčne skupine glede na vrednostno

strukturo so: folije v rolah (34%), bloki (21%), oblikovani izdelki (23%), čistilne gobice (18%) in reciklirana pena, kot so kosmiči in kosmit (4%).

2.2. Informacijski sistem podjetja

Informacijski sistem podjetja Plama-pur d.d. sloni na štirih med seboj povezanih strežniških sistemih, na katerih teče Microsoft SQL Server. Na vseh štirih strežniških sistemih teče ista verzija 7.0. Tu so zbrane vse zbirke podatkov, ki jih za delovanje potrebujejo programi informacijskega sistema. Zaradi velikega števila delujočih programov imajo tudi veliko različnih podatkovnih zbirk. Nekatere so namenjene posameznim programom, določene pa so skupne za celotno podjetje. Primer take zbirke je šifrant delavcev. Vpogled vanj je mogoč iz različnih programov. Različni podatki pa pritekajo tudi iz drugih sistemov. Eden izmed njih je sistem za sledenje blokov v proizvodnji in skladiščih. Ker je sistem je nastal s pomočjo zunanjih izvajalcev, si drugi programi podatke z zbirko podatkov za sledenje blokov le izmenjujejo. Izmenjava podatkov poteka avtomatsko in se opravi vsaki dve minuti. Ker so tako zbrani vsi podatki na strežniškem sistemu podjetja, imajo lahko različni programi vpogled le v zbirko že izmenjanih podatkov. V nasprotnem primeru bi moral vsak program na posamezni postaji sproti preverjati tudi stanje v sistemu za sledenje blokov. Podobna izmenjava poteka tudi s sistemom SPRO – Spremljanje proizvodnje. Novo pridobljene podatke o proizvodnji, predvsem podatke o delovnih nalogih, sistem SPRO neposredno vpisuje v podatkovne zbirke podjetja.

Kljub temu, da so sistem za slednje blokov v proizvodnji in skladiščih ter sistem za spremljanje proizvodnje razvili zunanji sodelavci, pa velik del programske opreme v podjetju razvijejo sami (slika 1). Tako na primer z interno izdelanimi programi pokrivajo področja prodaje, nabave, računovodstva, planiranja in nadzora proizvodnje, tehnologije dela, skladiščenja in odpreme. Velika večina teh programov je izdelana z orodjem Borland Delphi in sloni na podatkovnih zbirkah Microsoft SQL Server. Nekaj jih še vedno deluje v okolju DOS, vendar se trudijo vse programe prenesti v okolje Windows.



Slika 1: Seznam interno izdelanih programov podjetja Plama-pur d.d..

2.3. Obstoječi program Naloge

Obstoječi program Naloge deluje na osnovi orodja za delo s podatkovnimi zbirkami Microsoft Access. Z istim orodjem je izdelan tudi uporabniški vmesnik obstoječega programa. Ta omogoča vpisovanje, popravljanje in brisanje podatkov v podatkovni zbirki.

2.3.1. Uporabniški vmesnik

Obstoječi program, kot je razvidno iz ekranske slike glavnega menija (slika 2), omogoča različne funkcije, kot so vnos nalog, pregled oziroma popravljanje, tiskanje poročil in vpogled v šifrante.



Slika 2: Ekranska slika glavnega menija programa Naloga

Če na glavnem meniju izberemo funkcijo *vnos nalog*, se odpre obrazec za vnos nalog (slika 3). Vanj moramo vnesti *vir*, *tematsko področje*, *organizacijsko enoto*, *datum*, *naziv*, *opis* in *cilj* ter izbrati *odgovorno* in *sodelujoče osebe*. Potrebno je tudi izbrati vsakega izmed faktorjev in tako izbrati prioriteto. Ko je naloga zaključena, v istem obrazcu vnesemo še opis rešitve ter datum dokončanja posamezne delovne naloge.

Faktor		Plan.		Dej.	
Nosilec:	Bojan Gorjanc	100	100		
Sodelujoči:	0	0	0		

Slika 3: Ekranska slika vnosa podatkov programa Naloga

Če v glavnem meniju izberemo *pregled nalog*, se prikažejo podatki o delovnih nalogah (slika 4): *šifra, naziv, datum vpisa, rok, naziv, vir, cilj, organizacijska enota, odgovorna oseba*. Med posameznimi vnosi se lahko premikamo s smernimi tipkami.

The screenshot shows a software interface for viewing tasks. At the top, there is a menu bar with options: Datoteka, Urejanje, Orodja, Zapisi, Pomoč. Below the menu bar are navigation buttons (back, forward, home) and a date range filter: Datum od: 01.01.2005 do: 06.09.2005. There are also icons for search, print, and refresh.

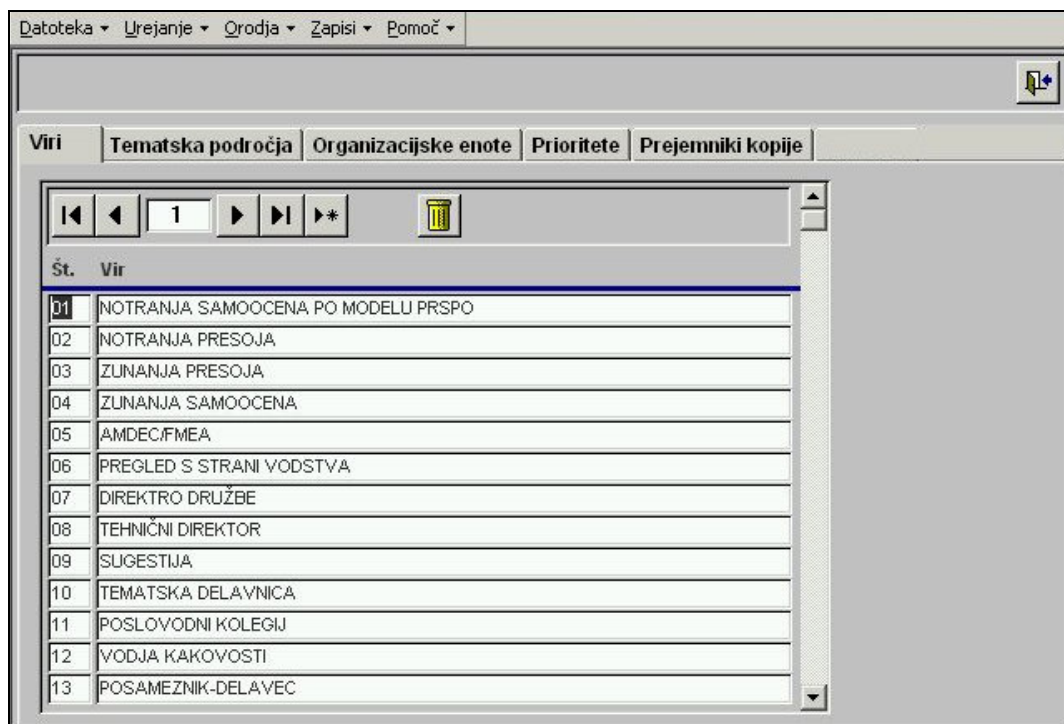
Below the date filter, there are several filter dropdown menus: **Dokončanost** (set to <VSE>), **Vir** (set to <VSI>), **Tematsko področje** (set to <VSA>), **O.E.** (set to <VSE>), **Prior.** (set to <VSE>), and **Prim. odgovornost** (set to <VSE>).

The main area displays a table of tasks with the following columns: Šifra, Naziv, Datum, Vir, Cilj, Tem. področje, Org. enota, Prior., and Odg. oseba. Three task entries are visible:

Šifra	Naziv	Datum	Vir	Cilj	Tem. področje	Org. enota	Prior.	Odg. oseba
15378	Rekl.I050015-z	04.03.2005	KOREKTI	Opozoriti delavc	Izdelava	PE PP	2	Alojz Križman
	Rok za izvedbo:	14.03.2005	Dokončana:	Ver. 0:	Ne	Dat. ver. 0:		Slavica Drobnjak
	Datum rešitve:		<input type="radio"/> Da <input checked="" type="radio"/> Ne	Ver. 1:	Ne	Dat. ver. 1:		
15379	Reklam.I050014	04.03.2005	KOREKTI	Ulivati ustrezno š	Izdelava	PE OP	2	Slavica Drobr
	Rok za izvedbo:	14.03.2005	Dokončana:	Ver. 0:	Ne	Dat. ver. 0:		
	Datum rešitve:		<input type="radio"/> Da <input checked="" type="radio"/> Ne	Ver. 1:	Ne	Dat. ver. 1:		
15340	Rekl.D040136-r	11.01.2005	KOREKTI	Izdelava kvalitetr	Izdelava	PE OP	2	Sandi Jagodn
	Rok za izvedbo:	20.01.2005	Dokončana:	Ver. 0:	Ne	Dat. ver. 0:		Alojz Križman,
	Datum rešitve:		<input type="radio"/> Da <input checked="" type="radio"/> Ne	Ver. 1:	Ne	Dat. ver. 1:		Miro Babuder

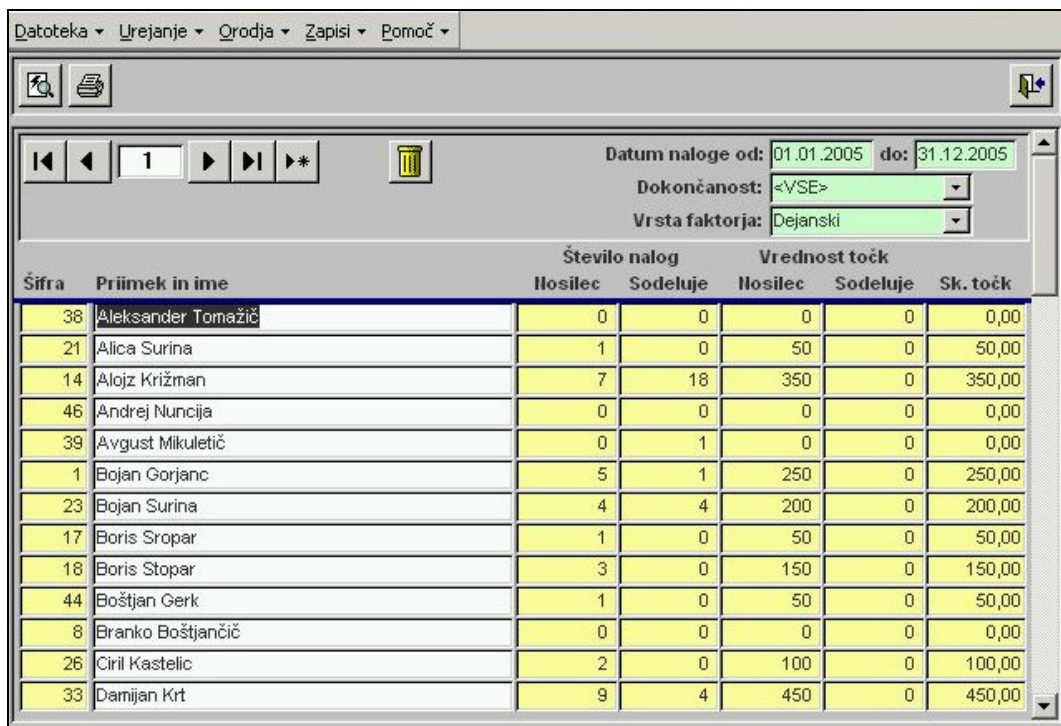
Slika 4: Ekranska slika pregleda nalog v programu Naloge

Iz ekranske slike šifrantov (slika 5) je razvidno da lahko tu dodajamo, spreminjamo ali brišemo različne vire, tematska področja, organizacijske enote. Seveda lahko posamezne podatke iz šifrantov odstranimo le, če ti niso v uporabi.



Slika 5: Ekranska slika šifrantov v programu Naloge

Podoben obrazec je tudi šifrant odgovornih oseb (slika 6). Prav tako tudi tu ne moremo brisati podatkov, ki so v uporabi.

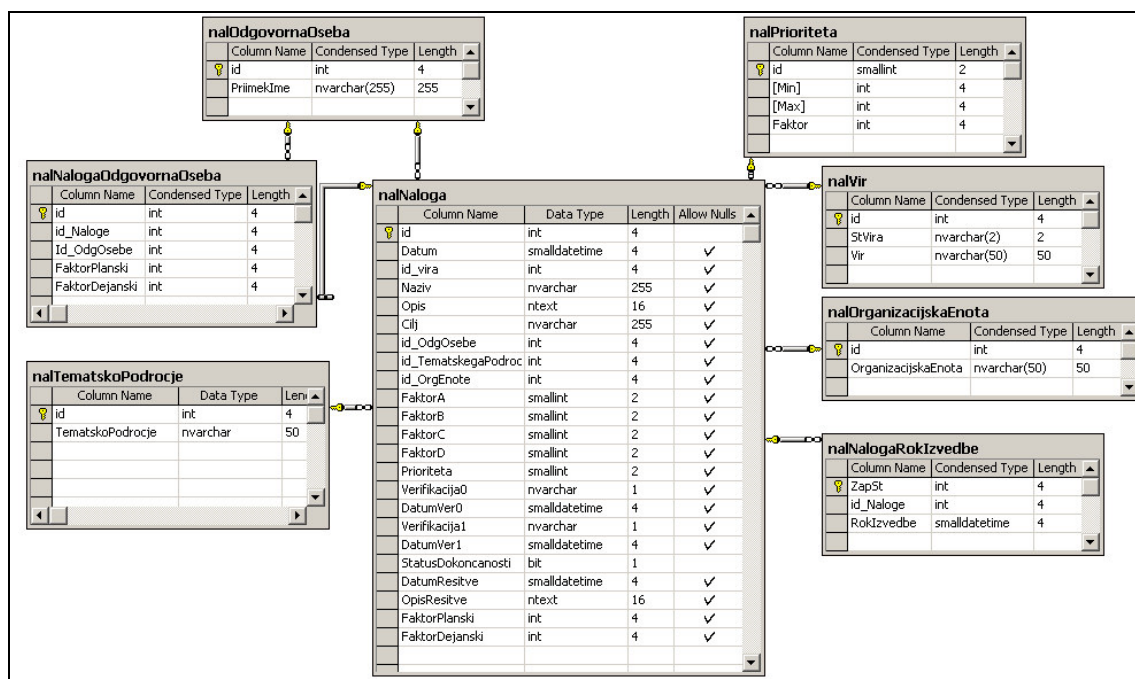


Slika 6: Ekranska slika šifranta odgovornih oseb v programu Naloge

2.3.2. Podatkovna zbirka

Kot smo že omenili, zbirka podatkov obstoječega programa sloni na orodju Microsoft Access. Iz entitetno-relacijskega modela (slika 7) je razvidno, katere šifrante uporablja program Naloge, prav tako so vidne povezave med posameznimi tabelami. Program Naloge uporablja tabele *nalNaloga*, *nalNalogaRokIzvedbe*, *nalOdgovornaOseba*, *nalNalogaOdgovornaOseba*, *nalTematskoPodrocje*, *nalVir*, *nalOrganizajskaEnota* ter *nalPrioriteta*.

V tabeli *nalNaloga* so zbrani podatki, ki povezujejo vse ostale tabele v vsebinsko zaključeno celoto, ki je osnova programa Naloge. Nekateri podatki so vezani na druge tabele, npr. *idOdgOsebe*, ki je vezan na šifrant *nalOdgovornaOseba*, drugi pa so specifični za posamezno nalogo, kot npr. *OpisResitve*. Tabela *nalNalogaOdgovornaOseba* je povezana s tabelama *nalNaloga* in *nalOdgovornaOseba*. Druge tabele so vezane na tabelo *nalNaloga* z relacijo ena proti mnogo, tako da je ena vrstica v posamezni tabeli vezana na več vrstic v tabeli *nalNaloga*.



Slika 7: Entitetno-relacijski model podatkovne zbirke Naloge

3. UPORABLJENA METODOLOGIJA IN PROGRAMSKA ORODJA

Tudi nova programska podpora za spremljanje delovnih nalog bo temeljila na relacijskem podatkovnem modelu. Pri razvoju programske podpore za spremljanje delovnih nalog smo uporabili sistem za upravljanje podatkovnih zbirk Microsoft SQL Server in orodje za programiranje Delphi. Microsoft SQL Server je zmogljiv strežnik za zbirke podatkov, ki z integriranimi orodji zagotavlja prvovrstno upravljanje podatkov. Delphi je visoko zmogljivo vizualno okolje za programiranje v okolju Windows. Tako relacijske zbirke podatkov kot Microsoft SQL Server in orodje Delphi bomo na kratko opisali v nadaljevanju poglavja.

3.1. Relacijske zbirke podatkov

Relacijski podatkovni model sodi med površinske modele in se uporablja za predstavitev podatkovnih zbirk na konceptualnem in zunanjem nivoju, hkrati pa je tudi sredstvo za specifikacijo konceptualne in zunanjih shem. Kot opisni in strukturni mehanizem je vgrajen v sisteme za upravljanje relacijskih podatkovnih zbirk (Mohorič, 1995).

Relacijska podatkovna zbirka se uporabniku kaže kot množica tabel. Vsaka tabela je sestavljena iz čelne vrstice in podatkovnih vrstic. Čelna vrstica pojasnjuje pomen posameznih stolpcev v tabeli, na osnovi kompatibilnega tipa stolpcev pa se dajo tabele tudi povezovati med seboj. Uporabniku so dostopne osnovne tabele, katerih vrstice so fizično shranjene v fizični podatkovni zbirki, na voljo pa so mu tudi nad osnovnimi tabelami definirane navidezne tabele, ki nimajo neposredne fizične predstavitve.

Vsebina tabel je uporabniku dostopna s pomočjo povpraševalnih jezikov, ki so osnovani na relacijski algebri ali pa na relacijskem računu. Predstavnik nepostopkovnih jezikov, ki je doživel standardizacijo v svetovnem merilu (ANSI, ISO), je SQL (Structured Query Language). SQL ni le povpraševalni jezik, pač pa je posplošen do te mere, da obsega tudi sredstva za definicijo podatkovnih struktur, specifikacijo integritetnih omejitev, podeljevanje pristopnih dovoljenj in zaščito podatkovne zbirke. Relacija v podatkovni zbirki je model nekega dogajanja ali stanja

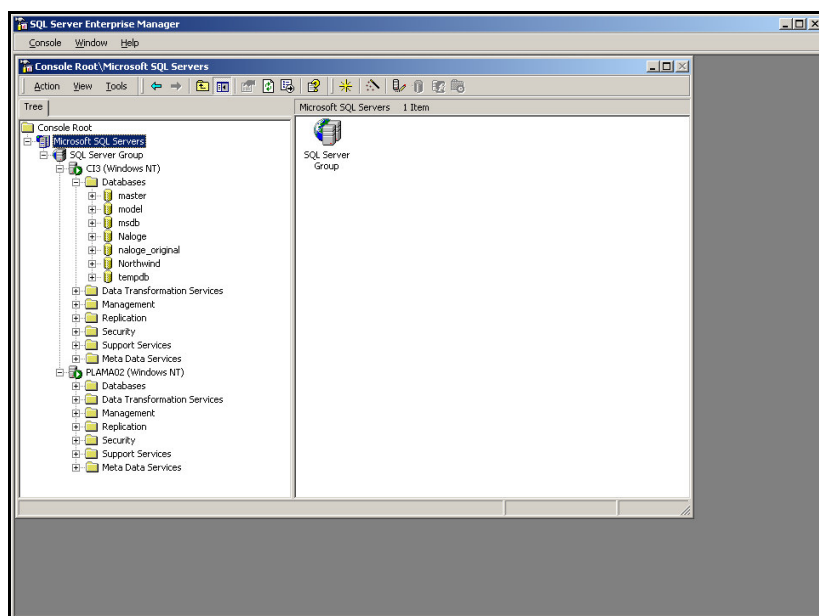
v svetu. Z relacijo lahko povežemo dve tabeli na osnovi kompatibilnega tipa stolpcev.

3.2. Sistem za upravljanje podatkovnih zbirk Microsoft SQL Server

Microsoft SQL Server zagotavlja varno in zanesljivo shranjevanje tako relacijskih kot strukturiranih podatkov in tako omogoča zmogljivo upravljanje podatkovnih zbirk. Jezik Microsoft SQL je prav zaradi svoje vloge upravljanja s podatki ključen del sistema Microsoft Server. Poleg tega SQL Server vsebuje funkcije analize, poročanja, integracije in obveščanja.

3.2.1. Upravljanje strežnika

Enterprise Manager (slika 8) je grafično orodje, ki omogoča administracijo SQL strežnikov. Z njim lahko urejamo podatke o uporabnikih in njihovih pravicah, upravljamo s podatkovnimi bazami, izvajamo arhiviranje podatkov, upravljamo s strukturami baz, tabel itd. Podobno lahko v njem registriramo tudi oddaljeni strežnik in se tako preko omrežja povežemo s takim sistemom. S pomočjo orodja Enterprise Manager lahko s podatki v podatkovnih zbirkah tudi manipuliramo. Posamezne podatke lahko dodajamo, brišemo ali posodabljam (System Administration for Microsoft SQL Server 7.0, 1998).



Slika 8: Ekranska slika Enterprise Managerja

Zelo uporabno orodje v sklopu strežnika Microsoft SQL Server je tudi SQL Server Query Analyzer, s katerim lahko izvajamo SQL stavke, analiziramo in na osnovi grafičnih analiz optimiziramo njihovo izvajanje. Z njim je mogoče posamezne podatke v tabelah tudi spreminjati. Kot primer poizvedbe smo iz podatkovne zbirke Naloge prikazali vsebino tabele *nalVir* (slika 9).

	id	StVira	Vir
1	1	01	NOTRANJA SAMOOCENA PO MODEL...
2	2	02	NOTRANJA PRESOJA
3	3	03	ZUNANJA PRESOJA
4	4	04	ZUNANJA SAMOOCENA
5	5	05	AMDEC/FMEA
6	6	06	PREGLED S STRANI VODSTVA
7	7	07	DIREKTRO DRUŽBE
8	8	08	TEHNIČNI DIREKTOR
9	9	09	SUGESTIJA
10	10	10	TEMATSKA DELAVNICA
11	11	11	POSLOVODNI KOLEGIJ
12	12	12	VODJA KAKOVOSTI
13	13	13	POSAMEZNIK-DELAVEC
14	14	14	DRUGO

Slika 9: Ekranska slika poizvedbe v SQL Server Query Manager

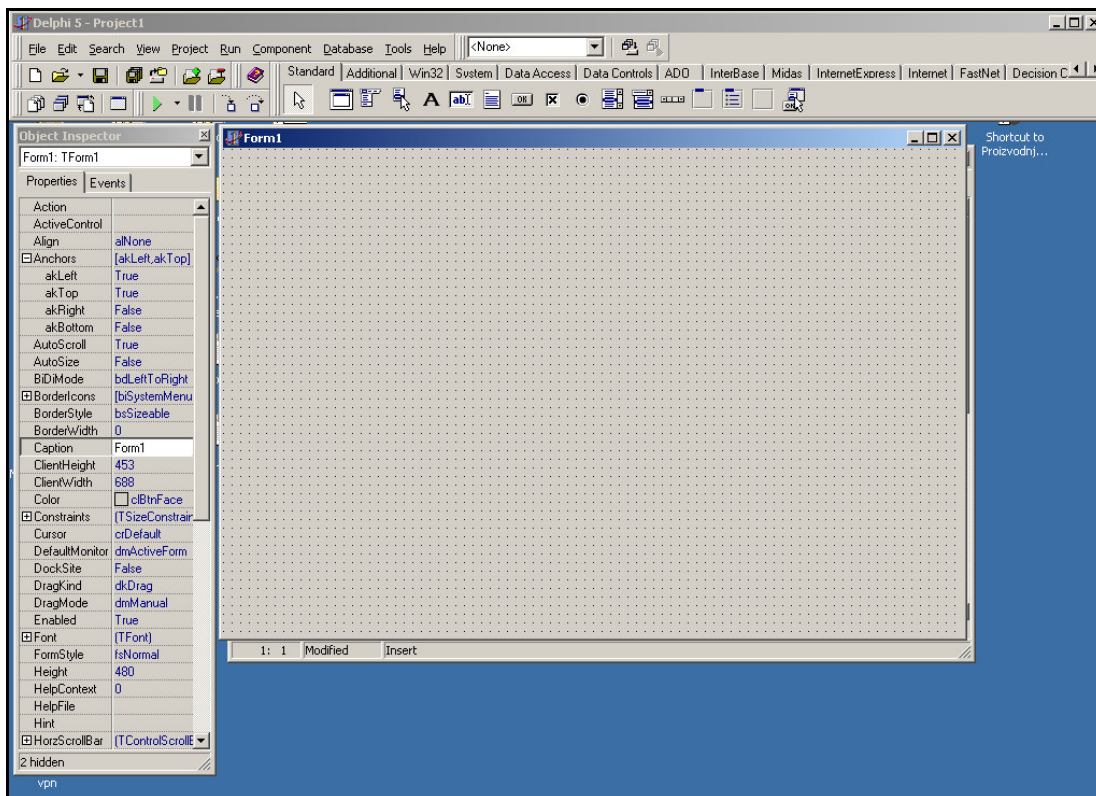
3.2.2. Arhitektura podatkovnih zbirk

Podatki na strežniku Microsoft SQL Server so shranjeni v podatkovnih bazah. Podatki v teh bazah so logično urejeni in vidni uporabnikom z zadostnimi pravicami. Fizično pa predstavlja podatkovno bazo ena ali več datotek na disku. Pri delu s podatkovno bazo v največji meri uporabljamo njene komponente, kot so tabele, pogledi, procedure in uporabniki. Fizična implementacija na nivoju datotek je za uporabnike popolnoma nevidna, dostop do nje pa ima samo administrator podatkovne baze. Vsak strežnik ima več podatkovnih baz. Strežnik ima štiri systemske podatkovne zbirke (*master*, *model*, *tempdb* in *msdb*) in eno ali več uporabniških podatkovnih zbirk. Strežnik omogoča hkratni dostop več uporabnikom do več podatkovnih baz in elementov v njih, seveda v skladu s pravicami, ki so določene na nivoju posameznega uporabnika.

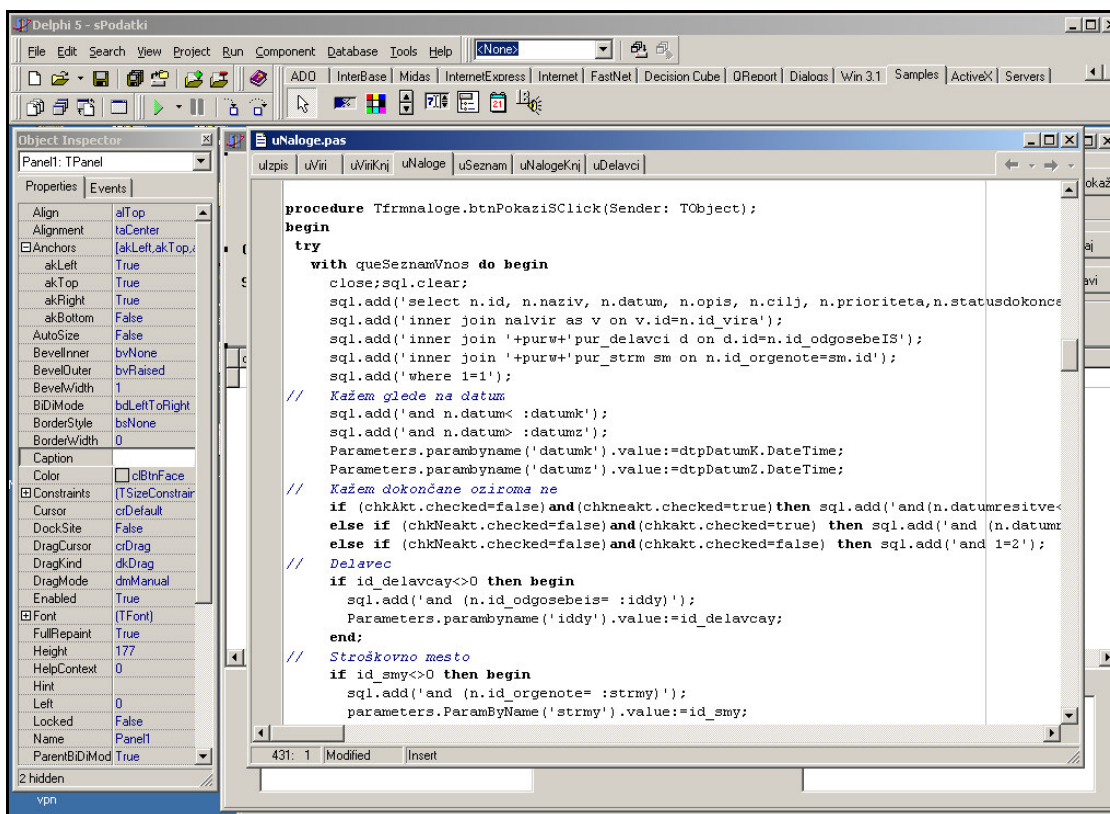
Vsi podatki, ki jih hranimo v strežniku Microsoft SQL Server, so organizirani v objektih – tabelah. Vsaka tabela ima stolpce in vrstice. Stolpci določajo strukturo hranjenih podatkov v tabeli, vrstice pa hranijo podatke. Ker vsak stolpec vsebuje podobne podatke, mu določimo podatkovni tip. Microsoft SQL Server pozna naslednje podatkovne tipe: binary, bit, char, datetime, decimal, float, image, int, money, nchar, ntext, nvarchar, numeric, real, smalldatetime, smallint, smallmoney, sysname, text, timestamp, tinyint, varbinary, varchar in uniqueidentifier (Programming a Microsoft SQL Server 2000 Database, 2000).

3.3. Razvojno programsko okolje Borland Delphi

Borland Delphi je visoko zmogljivo vizualno okolje za programiranje v operacijskem sistemu Windows (slika 10). V podjetju so se za izbor tega orodja odločili, ko so ocenili razmerje med njegovo zmogljivostjo in enostavnostjo. Izbrali so ga predvsem zaradi zagotovitve skladnosti s programsko opremo in informacijskim sistemom v podjetju. Za pisanje programske kode v okolju Delphi se uporablja jezik predmetni pascal (slika 11). Njegova največja prednost je v načinu ustvarjanja računalniških aplikacij. Delphi se navzven kaže kot univerzalno orodje z mnogo obrazi. Enostaven računalniški program lahko izdelamo skoraj brez vsakih programerskih izkušenj, z malo več znanja pa hitro izdelamo tudi profesionalno aplikacijo.

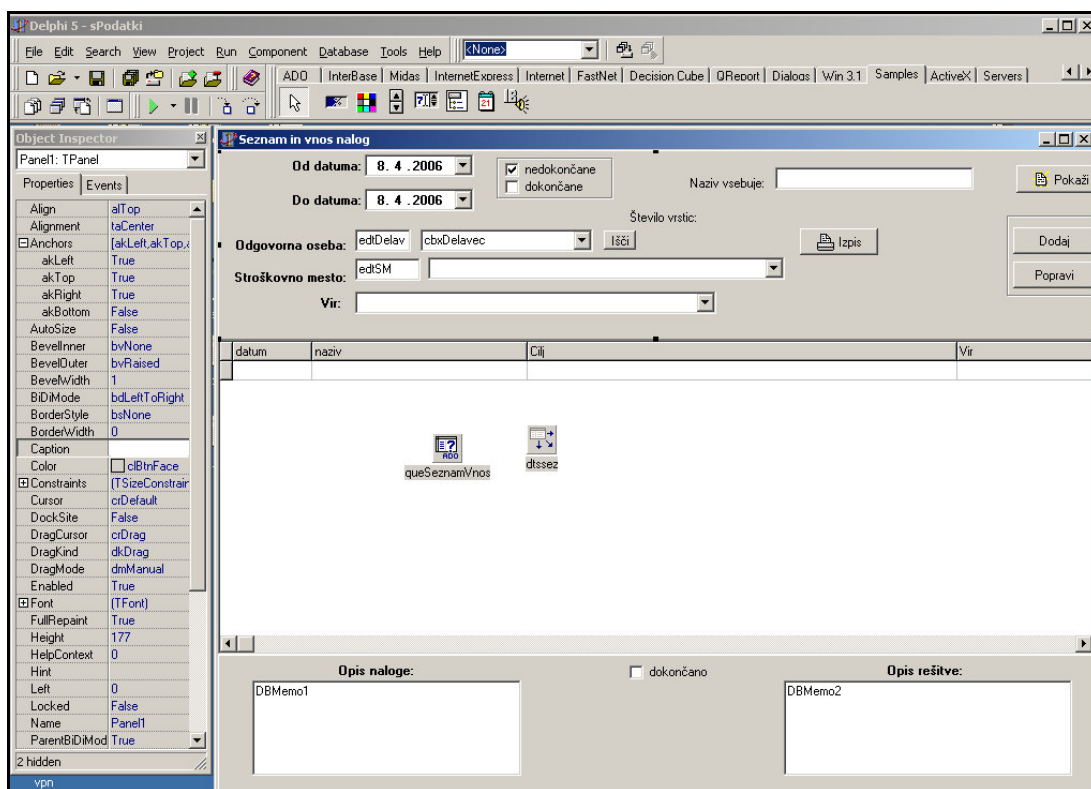


Slika 10: Ekranska slika razvojnega okolja Borland Delphi

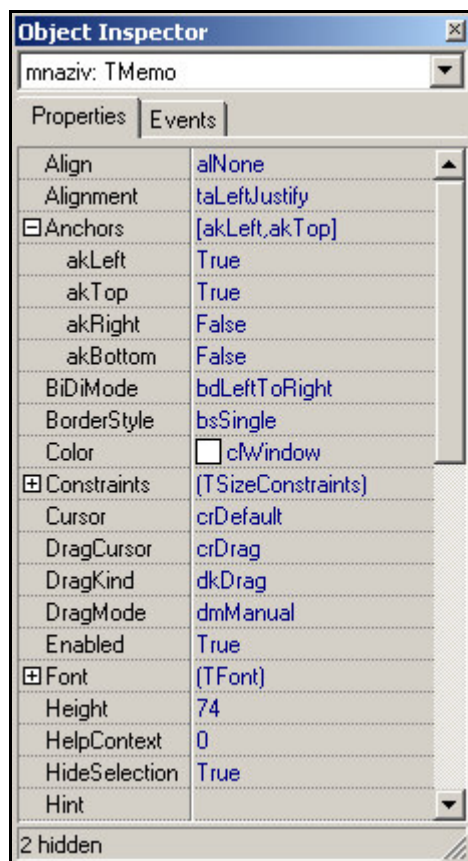


Slika 11: Ekranska slika razvojnega okolja s pogledom na izvorno kodo

Jezik je objektno usmerjen (object oriented) in podpira razrede, dedovanje, polimorfizem in enkapsulacijo. Ponuja dinamične in virtualne metode, objavljene (published) lastnosti, konstruktorje in destruktorje z večkratnim poimenovanjem, ter dostop do privatnih delov razreda preko posameznih lastnosti (Borland Delphi 5 – Object Pascal Language Guide, 1999). Orodje ponuja čudovito razvojno okolje z vizualnimi gradniki, izvedenci (experts) in vodiči (tutors) (slika 12). Orodje Delphi je enostavno in omogoča programerju, da začne s preprostimi programi, postopoma pa orodje izkoristi v vseh njegovih razsežnostih (Borland Delphi 5 – Client/Server Foundations, 1999). Skladnja programov v Delphiju je striktna, čista in enostavna za branje. Prevajalnik, ki je vgrajen v Delphi, je svetovni rekorder v hitrosti prevajanja. Tako hiter je, da po privzeti nastavitvi sploh ne prikazuje poteka prevajanja (sicer pa po želji prikazovanje lahko vključimo). Posamezne lastnosti vizualnih gradnikov lahko spreminjamo z orodjem Object Inspector (slika 13), določene lastnosti pa lahko spreminjamo tudi iz izvorne kode programa (slika 11).



Slika 12: Ekranska slika razvojnega okolja s pogledom na vizualne gradnike



Slika 13: Ekranska slika orodja Object Inspector

Če združimo podatke in kodo, ki nad njimi deluje, dobimo objekte – predmete. Predmeti združujejo stanje in obnašanje entitet, ki jih predstavljajo. Stanje je množica vrednosti atributov, se pravi podatkov, ki so shranjeni v predmetu, in povezav, ki jih ima predmet z drugimi predmeti, obnašanje pa je programska koda, ki deluje nad podatki. Pri svojem delovanju podatke pogosto spremeni, zato rečemo, da se je spremenilo stanje predmeta (Borland Delphi 5 – Developer's Guide, 1999).

Da bi zagotovili, da se stanje predmeta (podatkov v predmetu) ne more spremeniti naključno, so predmeti ograjeni (enkapsulirani). Predmet specificira seznam operacij (metod), ki jih podpira.

Za dostopanje do podatkovne zbirke SQL iz Delphija smo izbrali ADO (ActiveX Data Object). Ker je to Microsoftovo orodje, razvito za podporo njihove podatkovne zbirke, lahko pričakujemo, da so ga dovolj dobro pripravili in ob naslednji posodobitvi strežnika Microsoft SQL Server ne bo potrebno čakati na Borland, da pripravi ustrezen BDE (Borland Database Engine). Perspektivno je ADO zanimiv

tudi zaradi možnosti dostopa do podatkov elektronske pošte in drugih Microsoftovih produktov (Kostrevc, 2000).

Do podatkovne zbirke SQL dostopa kot uporabnik le aplikacija, ki ima tudi pravico branja, pisanja in brisanja podatkov. Tako vsi uporabniki posredno preko nje dostopajo do baze. Razen skupine administratorjev nobeden od uporabnikov na mreži nima pravice dostopa do baze podatkov.

4. RAZVOJ NOVEGA PROGRAMA NALOGE

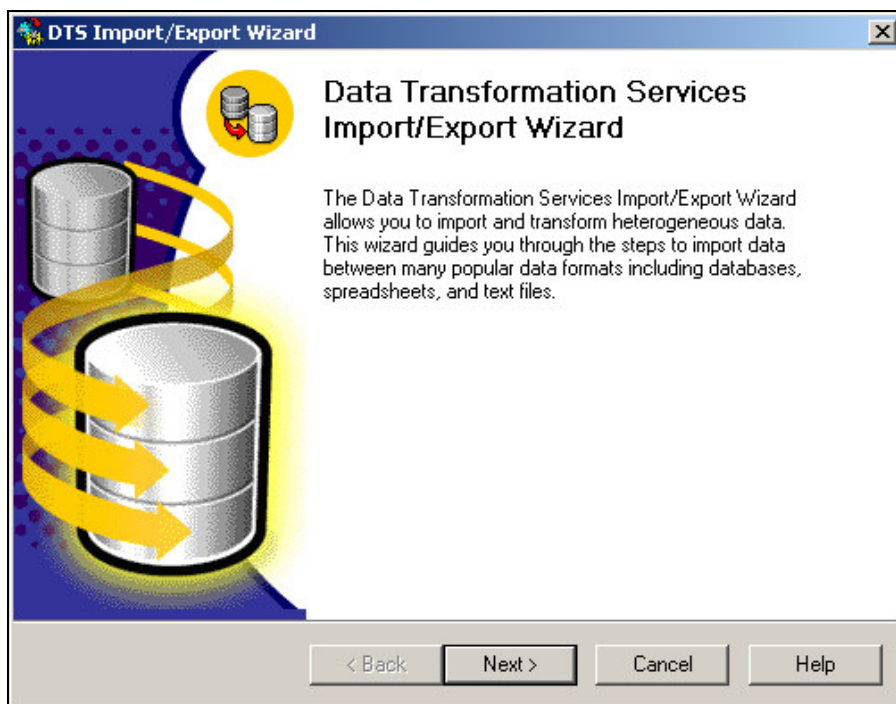
V 3. poglavju opisano metodologijo in programska orodja smo uporabili tudi pri razvoju novega programa Naloga. Pri razvoju smo podatkovne zbirke iz obstoječega programa, ki temelji na orodju za delo s podatkovnimi bazami Microsoft Access, uvozili v strežnik Microsoft SQL Server. Tam smo po potrebi spremenili povezave med tabelami in podatkovni tip stolpcev, nekatere tabele pa smo v celoti odstranili. Odstranjene tabele so bile večinoma nepotrebne zaradi skupnih podatkovnih zbirk podjetja, medtem ko smo programski del v celoti opustili in pričeli s programiranjem z orodjem Borland Delphi.

4.1. Podatkovna zbirka

Delo na podatkovnih zbirkah za novi program Naloga je temeljilo predvsem na uvažanju podatkovnih zbirk v strežnik Microsoft SQL Server in prilagajanju zbirk za ta strežnik. Ponovno smo postavili povezave med tabelami, nekatere tabele pa v celoti odstranili, saj niso več potrebne. Določili smo tudi način dostopa uporabnikov do podatkovne zbirke.

4.1.1. Uvoz podatkov

Podatke smo v strežnik Microsoft SQL Server uvozili s pomočjo orodja Enterprise Manager s funkcijo Data Transformation Services (slika 14). Ta funkcija podpira uvažanje podatkov iz različnih podatkovnih zbirk, npr. dBase, Microsoft Access in Paradox.

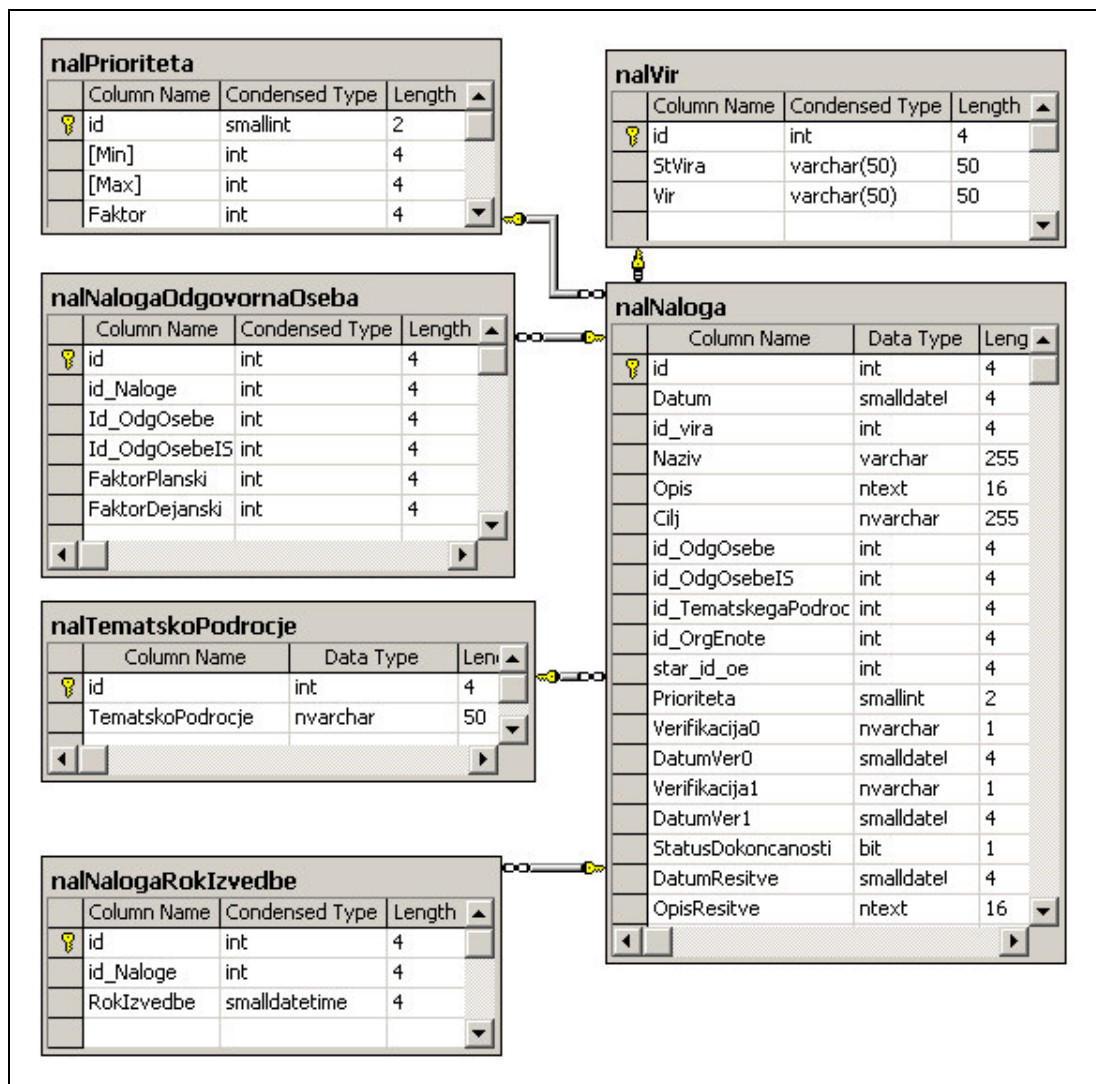


Slika 14: Funkcija Data Transformation Services v orodju Enterprise Manager

4.1.2. Razvoj podatkovne zbirke

Ko smo v strežnik uvozili podatke, smo vzpostavili povezave med tabelami, kajti funkcija Data Transformation Services uvozi le posamezne tabele, ne pa tudi povezav med njimi. Ponovno smo definirali tudi podatkovne tipe nekaterih stolpcev. Zamenjava podatkovnih tipov nekaterih stolpcev je bila potrebna zaradi skladnosti z razvojnim okoljem Borland Delphi. Ker so bile nekatere tabele v podatkovni zbirki programa Naloge odveč, smo jih odstranili. Odstranili smo tabele, ki zaradi povezave s skupnimi podatkovnimi zbirkami podjetja niso bile več potrebne. Vse te spremembe na podatkovnih zbirkah lahko vidimo, če primerjamo entitetno-relacijski model podatkovne zbirke Naloge (slika 7) in entitetno-relacijski model podatkovne zbirke novega programa Naloge (slika 15). Primerjava pokaže, da so bile odstranjene tabele *nalOdgovornaOseba* in *nalOrganizacijskaEnota*. Tabeli *nalNaloga* smo dodali stolpec *id_OdgOsebeIS*, ki je vezan na podatke o delavcih iz podatkovnih zbirk podjetja. Podatke smo najprej iz skupnih podatkovnih zbirk ročno vpisali v tedaj še obstoječo tabelo *nalOdgovornaOseba* v stolpec *id_IS*. Podatke smo v tabelo vpisovali ročno, saj so se polja medsebojno preveč razlikovala, npr. zamenjana

priimek in ime. Nato smo v orodju SQL Query Manager posodobili polja stolpca *id_OdgOsebeIS* tabele *nalNaloga*.



Slika 15: Entitetno-relacijski model podatkovne zbirke novega programa Naloga

Tako smo se izognili vpisovanju novih podatkov o delavcih v šifrante programa, saj bodo programu vedno na voljo sveži podatki. Ob morebitnih modifikacijah programa pa lahko morebitne manjkajoče podatke o delavcih preprosto dodajamo v program, npr. telefonsko številko delavca. Podobno smo storili v tabeli *nalNalogaOdgovornaOseba*. Tudi tam smo dodali polje *Id_OdgOsebeIS*, ki se tako kot v tabeli *nalNaloga* ujema s poljem *id* iz tabele *pur_delavci* v skupnih podatkovnih zbirkah. Tabela *pur_delavci* vsebuje polja *id*, *šifra*, *naziv*, *id_strm* (ki se navezuje na podatke o stroškovnih mestih) ter druga polja s podatki o delavcih. Ker smo v tabeli

nalOdgovornaOseba že imeli podatke, ki se ujemajo s skupno podatkovno zbirko, smo jih le primerjali med seboj in posodobili polja *Id_OdgOsebeIS* v tabeli *nalNalogaOdgovornaOseba*. Tabela *nalOrganizacijskaEnota* smo odstranili, saj so v novem programu Naloge podatki o organizacijski enoti razvidni iz podatkov o delavcu. Vsak delavec pripada določeni organizacijski enoti, tako da je organizacijska enota definirana na podlagi odgovorne osebe. Tudi tu ne bo več potrebno v šifrantu programa vpisovati podatkov o organizacijskih enotah, saj so ti definirani v skupnih podatkovnih zbirkah podjetja.

Iz entitetno-relacijskega modela podatkovne zbirke novega programa Naloge (slika 15) je razvidno, da so v podatkovni zbirki ostale le še tabele: *nalNaloga*, *nalNalogaOdgovornaOseba*, *nalPrioriteta*, *nalVir*, *nalNalogaRokIzvedbe* in *nalTematskoPodročje*. Tabela *nalNaloga* združuje podatke o delovnih nalogah in je povezana z vsemi drugimi tabelami, saj se nanje navezujejo podatki v njej. Tu so tudi podatki, ki v skupnih šifrantih podjetja kažejo na podatke o odgovornih osebah in organizacijskih enotah. Tabela *nalNalogaOdgovornaOseba* združuje podatke o posameznih delovnih nalogah in sodelujočih osebah pri delovnih nalogah. *NalPrioriteta* je šifrant prioritete delovnih nalog. Podobna je tabela *nalVir*, v kateri so podatki o virih. Obe tabeli imata povezavo s tabelo *nalNaloga* s števnostjo ena proti mnogo. V tabeli *nalNalogaRokIzvedbe* so zapisani podatki o rokih izvedbe, posamezna naloga ima lahko vpisanih več različnih datumov. Tabela *nalTematskoPodročje* je šifrant tematskih področij delovnih nalog. Te lahko spreminjamo preko uporabniškega vmesnika novega programa Naloge.

4.1.3. Uporabniki in pravice dostopa do podatkovne zbirke

Večina programov v podjetju je zgrajena tako, da pri dostopanju do podatkovne zbirke ne uporablja imena in gesla uporabnika, ki je v program prijavljen, ampak uporabljamo za dostop do podatkovne zbirke ime in geslo virtualnega uporabnika. Ta ima pravico branja in pisanja v vseh podatkovnih zbirkah, uporabnikom pa dostop do teh funkcij omogočamo preko uporabniškega vmesnika. Vsak uporabnik se mora za dostopanje do programa prijaviti s svojim imenom (lahko je to tudi šifra delavca) in geslom. Program v skupnih tabelah podjetja preveri, ali ima uporabnik pravico vpogleda oziroma knjiženja v programu. Če nima nobene od teh pravic, ga zavrne in

program se zapre. V nasprotnem primeru mu program ponudi le funkcije, ki mu glede na pravice iz skupnih podatkovnih zbirk o pravicah uporabnikov pripadajo. Če ima uporabnik le pravico vpogleda, torej branja, so v programu skrite vse funkcije, ki omogočajo knjiženje oziroma pisanje v podatkovno zbirko. V primeru, da ima uporabnik le pravico vpogleda, je geslo enako šifri delavca, če pa ima tudi pravico knjiženja, je geslo sestavljeno iz naključno izbranih znakov.

4.1.4. Arhiviranje podatkov

Da bi se zaščitili pred možnostjo izgube podatkov in zagotovili možnost njihove obnove v primeru nepredvidenih nesreč, smo oblikovali tri opravila (Maintenance Plans), ki se prožijo na strežniku in arhivirajo podatke. Podatkovna zbirka novega programa Naloga se arhivira vsak večer ob 21:00, med 1:00 in 21:00 se vsako uro arhivira »transaction log«. Arhiva se zapisujeta na trdi disk strežnika in se na njem hranita dva tedna. Arhiviranje sistemskih baz *Master*, *Model* in *MsdB* poteka vsak večer ob 22:30. Vsako prvo nedeljo v mesecu ob 14:00 se proži optimizacija podatkovne zbirke novega programa Naloga, ki zagotavlja optimalno izkoriščenost prostora na disku in stabilno delovanje podatkovne zbirke.

Vsak večer ob polnoči se na strežniku proži arhiviranje preko orodja ArcServ, ki arhive prepíše na zunanji medij. Medije za dni od ponedeljka do četrтка vsak teden prepíšemo čez stare podatke (jih ponovno uporabimo), petkovih pa ne. Petkovih medijev je pet, torej imamo v vsakem trenutku najstarejše dostopne podatke stare dober mesec dni. Vsako leto se en komplet podatkov na medijih shrani, tako da imamo arhive tudi za vsa pretekla leta. Podatkovni mediji se shranjujejo v ognjevarnem trezorju v upravni stavbi podjetja, za njihovo menjavo pa je odgovoren sistemski administrator.

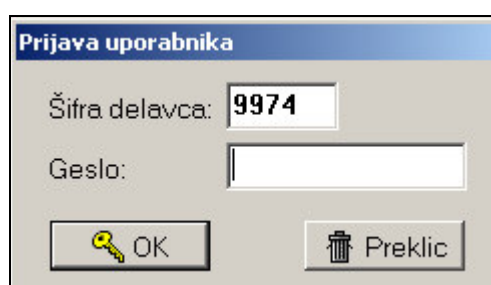
4.2. Uporabniški vmesnik

Vmesnik je izdelan z uporabo orodja Borland Delphi, zmogljivega orodja za vizualno programiranje v okolju Windows. S pomočjo tega orodja ob vsakem prevajanju izvorne kode ustvarimo novo izvedljivo datoteko. Ker se prevajanje kode odvija zelo hitro, je mogoče ob programiranju program tudi zaganjati in opazovati nastajajoči izdelek. Ta lastnost orodja Borland Delphi v praksi zelo pomaga pri delu.

Uporabniški vmesnik je zasnovan tako, da uporabniku omogoča prijazno pregledovanje in zapisovanje v podatkovno zbirko. Ko program zaženemo, se najprej pokaže logotip oziroma zaščitni znak (slika 16), ki je prisoten pri vseh interno izdelanih programih podjetja. Nato se moramo z uporabniškim imenom in geslom prijaviti v obrazec za prijavo (slika 17). Program uporabniško ime že vnaprej predlaga glede na to, kateri uporabnik je prijavljen. Kot je bilo v podpoglavju 4.1.3 o uporabnikih in pravici dostopa do podatkovne zbirke že omenjeno, iz te prijave izhajajo vse pravice do knjiženja ali branja v novem programu Naloge.



Slika 16: Logotip interno izdelanih programov podjetja Plama-pur d.d.

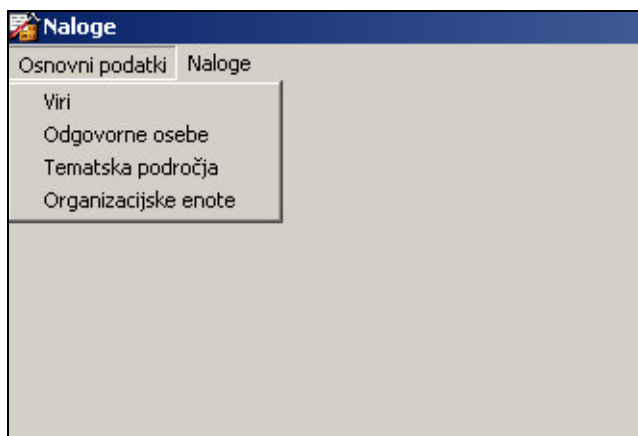


Slika 17: Obrazec za prijavo uporabnika v novi program Naloge

4.2.1. Glavni meni

Celoten program sloni na glavnem meniju, preko katerega dostopamo do vgrajenih funkcij (slika 18). V okviru osnovnih podatkov imamo dostop do šifrantov. To so šifranti virov, odgovornih oseb, tematskih področij in organizacijskih enot. Šifranta

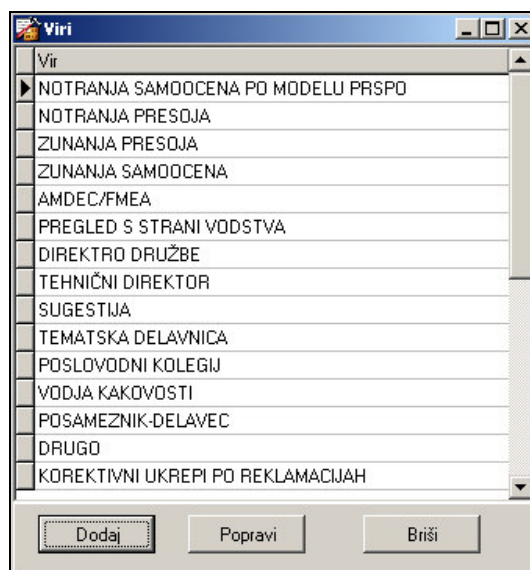
odgovornih oseb in organizacijskih enot se nanašata na podatke v skupnih podatkovnih zbirkah podjetja, zato jih je mogoče samo pregledovati, ne pa tudi popravljati in brisati. Prav tako iz glavnega menija dostopamo do seznama delovnih nalog, ki jih lahko v odvisnosti od uporabniških pravic dodajamo, popravljamo in brišemo.



Slika 18: Ekranska slika glavnega menija novega programa Naloge

4.2.2. Obrazci šifrantov

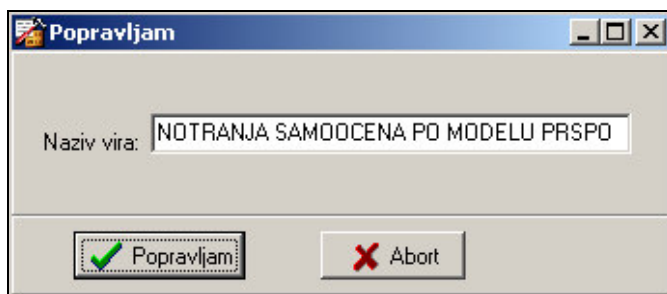
Kot je bilo omenjeno v podpoglavju 4.2.1, pridemo do vseh obrazcev novega programa Naloge skozi glavni meni programa. Če izberemo možnost *viri*, se odpre obrazec za pregled virov (slika 19).



Slika 19: Ekranska slika obrazca za pregled virov

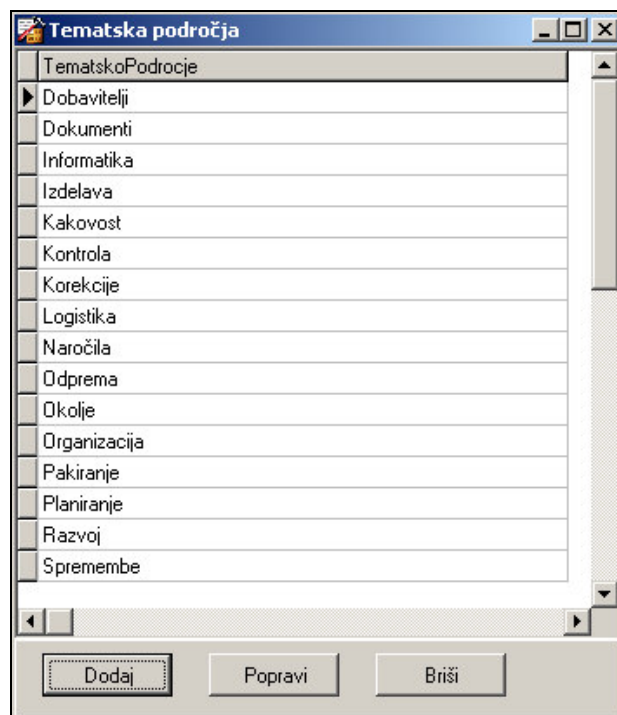
Vsakič, ko se obrazec prikaže, se v tabeli prikažejo viri. Podatke, kot sta *id* in *stvira*, smo v tabeli skrili, saj za uporabnika niso bistveni.

Če na obrazcu za pregled nalog pritisnemo katerega od gumbov, se prikaže obrazec za ažuriranje virov (slika 20). Določeni napisi na obrazcu se izmenjujejo glede na to, katerega izmed gumbov na obrazcu za pregledovanje nalog smo izbrali, v našem primeru je to obrazec za popravljanje naziva vira. Kot spremenljivka se zapiše tudi vrednost polja *id* iz obrazca za pregledovanje virov, saj na ta način program predlaga, kateri vir naj bi popravili oziroma izbrisali. Ob pritisku kateregakoli od gumbov *dodajam*, *popravljam* ali *brišem* se izvrši postopek, ki najprej preveri, če je pisanje, popravljanje ali brisanje mogoče izvršiti, nato pa le-to izvede. V primeru, da želimo zbrisati vnos, ki je še v uporabi, program tega ne dovoli. Kontroliramo tudi dolžino vnesenega podatka, zato ga v primeru, da je predolg, samodejno skrajša in takšnega zapiše v podatkovno zbirko. Uporabnika o tem le obvesti, ta pa lahko nato vnos poljubno popravi.



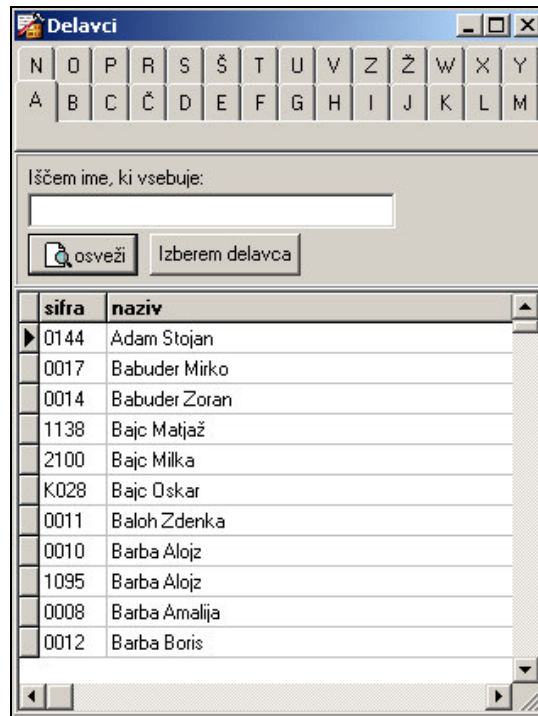
Slika 20: Ekranska slika obrazca za ažuriranje virov

Zelo podoben šifrantu virov je šifrant tematskih področij (slika 21), do katerega pridemo s klikom na *tematska področja* v glavnem meniju. Prav tako z izbiro katerega od gumbov na obrazcu za pregled tematskih področij pridemo do obrazca za knjiženje, preko katerega v podatkovno zbirko zapisujemo, v njej popravljamo ali brišemo podatke. Glede na izbiro gumba definiramo tudi napise v obrazcu za knjiženje tematskih področij. Na podoben način tudi tu nadziramo vsebino, ki jo hočemo v podatkovno zbirko zapisati.



Slika 21: Ekranska slika obrazca za pregled tematskih področij

Šifranta odgovornih oseb (slika 22) in organizacijskih enot se od drugih šifrantov razlikujeta predvsem v tem, da podatkov v njih ni mogoče spreminjati. To ni mogoče, ker podatki v njih izhajajo iz skupnih podatkovnih zbirk podjetja. Podatke je mogoče spreminjati iz nekaterih drugih programov. Tudi oblika obrazca odgovornih oseb je v večini interno izdelanih programov poenotena in je v uporabi vsakič, ko hočemo v nekem programu izbrati delavca. Obrazec vsebuje polje *iščem ime, ki vsebuje*. V polje lahko vpišemo del imena ali priimka odgovorne osebe, program pa prikaže seznam odgovornih oseb, ki ta del vsebujejo.



Slika 22: Ekranska slika obrazca za pregled odgovornih oseb

4.2.3. Obrazec Seznam in vnos nalog

Do obrazca Seznam in vnos nalog (slika 23) pridemo, če v glavnem meniju izberemo možnost *naloge*. Obrazec je namenjen pregledovanju, popravljanju in vnosu podatkov o delovnih nalogah. Na tem obrazcu se, če izberemo možnost *pokaži seznam*, prikaže seznam delovnih nalog. V seznamu lahko vidimo polja *datum*, *naziv*, *cilj*, *vir*, *odgovorna oseba* in *stroškovno mesto*. V spodnjem delu obrazca se v poljih *opis naloge* in *opis rešitve* izpisujejo podatki o izbrani nalogi. Med poljema *opis naloge* in *opis rešitve* je indikator dokončanosti naloge. Tudi ta prikazuje stanje izbrane naloge. Obrazec ima tudi števec vrstic, ki ob vsakem osveževanju seznama prikaže število prikazanih vrstic.

Seznam in vnos nalog

Od datuma: 21. 2. 2000 Do datuma: 20. 5. 2006

nedokončane dokončane

Naziv vsebuje:

Pokaži seznam

Število vrstic: 108

Odgovorna oseba: Išči

Izpis

Dodaj

Popravi

Stroškovno mesto:

Vir:

datum	naziv	Cilj	Vir	strm	odg_oseba
22.2.2000	Etalon za KONTUR	Tehnološko definirati ustrezni profil in izdelati etalon, ki bo na delovnem m	KOREKTIVNI UKREPI PO REKI 453		0534 Turkovič Sergej
22.2.2000	Check liste in preverjanje	Pripraviti CHECK LISTE	KOREKTIVNI UKREPI PO REKI 453		0534 Turkovič Sergej
22.2.2000	Rešitev vzrokov za reklamacije TOM.	Preventi izdelke in tehnologijo za kupca TOM Mima.	KOREKTIVNI UKREPI PO REKI 453		0534 Turkovič Sergej
22.2.2000	Pakiranje sedežev v garniturah in odpremi	Urediti odpreme z računalniško podporo	KOREKTIVNI UKREPI PO REKI 450		0677 Maurič Jadran
23.2.2000	Opozorila na nelogične količine v RPIS	Pregledati teoretične izračune ter izboljšati selektivnost sistema.	KOREKTIVNI UKREPI PO REKI 481		0534 Turkovič Sergej
23.2.2000	Preverjanje odpremi SIRETESILE	Komisjsko odpremljanje za tega kupca in ugotavljanje vzrokov za reklam.	KOREKTIVNI UKREPI PO REKI 171		0436 Zjacić Momčilo
7.3.2000	Dopolnitev tehnologije	Opredeliti zahtevo za podlaganje blokov pri razrezu v tehnologiji	KOREKTIVNI UKREPI PO REKI 120		0534 Turkovič Sergej
7.3.2000	Vnos in korekcija naročil v RPIS	Preventi ter definirati vnos in korekcije naročil	KOREKTIVNI UKREPI PO REKI 178		0135 Stopar Boris
7.3.2000	Etalon za ustreznost kosmita	O dkupca pridobiti vzorce ustreznega in neustreznega kosmita.	KOREKTIVNI UKREPI PO REKI 130		1063 Barba Uroš
10.4.2000	Sprememba tehnologije rezanja plošč	Sprememba tehnologije rezanja plošč	KOREKTIVNI UKREPI PO REKI 120		0534 Turkovič Sergej
21.7.2000	Spoštovanje predpisov	Poučiti delavce o nevarnostih nespoštovanja predpisov o varstvu pri del.	PREGLED S STRANI VODSTVA 450		0677 Maurič Jadran
21.7.2000	Orodja BG/05/00	Redno izvajati preglede orodij v vseh enotah.	KOREKTIVNI UKREPI PO REKI 453		0891 Gojanc Bojan
21.7.2000	Orodja BG/06/00	Preventi sistem označevanja sit v PE ČG	KOREKTIVNI UKREPI PO REKI 453		0534 Turkovič Sergej
21.7.2000	Orodja BG/04/00	Redno voditi knjigo načrtov na FKŽ.	KOREKTIVNI UKREPI PO REKI 453		0534 Turkovič Sergej
21.7.2000	Notranje presoje	Vključiti člane tima Q v notranje presoje	PREGLED S STRANI VODSTVA 470		0891 Gojanc Bojan
21.7.2000	Korektivni ukrepi	Preventi ukrepe po nosilih	PREGLED S STRANI VODSTVA 470		0891 Gojanc Bojan
21.7.2000	Korektivni ukrepi	Prenos vodenja ukrepov na računalnik	PREGLED S STRANI VODSTVA 470		0891 Gojanc Bojan
21.7.2000	Delo z dobavitelji	Oplimiranje števila dobaviteljev	PREGLED S STRANI VODSTVA 100		1332 Boštjančič Branko
21.7.2000	Poročilo NT	Poročilo dodati tudi izgubljene možnosti prodaje zaradi neakovosti	PREGLED S STRANI VODSTVA 178		1332 Boštjančič Branko
9.1.2003	Reklamacija D020158	Opozorilo skladiščnika	KOREKTIVNI UKREPI PO REKI 171		0779 Surina Bojan
9.1.2003	Reklamacija D020133	Opozorilo odpremnemu skladiščniku.	KOREKTIVNI UKREPI PO REKI 171		0779 Surina Bojan

Opis naloge: dokončano

Opis rešitve:

Ob reklamaciji kupca Moderna je bilo ugotovljeno, da izvajalci na liniji ne poznajo, kateri profil je še ustrezen za kupca.

Izdelan etalon in dan v proizvodnjo

Slika 23: Ekranska slika obrazca Seznam in vnos nalog

Prikazani seznam je odvisen od izbire filtrov na obrazcu. Če izberemo možnost *nedokončane*, nam prikazuje samo delovne naloge, ki niso zaključene in obratno, če izberemo možnost *dokončane*, se prikažejo samo dokončane naloge. Podobno deluje tudi datumski filter, ki prikaže le naloge v izbranem intervalu. Delovne naloge določene odgovorne osebe program prikaže, če v polje *odgovorna oseba* vpišemo šifro delavca oziroma v desnem polju izberemo iz seznama ime in priimek odgovorne osebe. Ker je število delavcev in s tem možnih odgovornih oseb veliko, si lahko pomagamo z gumbom *išči*. Tako se odpre obrazec za pregled odgovornih oseb, do katerega pridemo tudi z izbiro možnosti odgovorne osebe v glavnem meniju. Podobno deluje tudi polje *stroškovno mesto*. V polje lahko vpišemo številko stroškovnega mesta ali pa v desnem polju izberemo stroškovno mesto po imenu. Tudi vir delovne naloge lahko izberemo v polju *vir*. Ob izhodu iz večine polj se izvrši postopek, ki preveri vsebino in določene podatke zapiše kot vrednost spremenljivke, npr. *id* delavca v spremenljivko *iddy*.

4.2.4. Obrazec za knjiženje delovnih nalog

Obrazec je namenjen vnašanju in popravljanju podatkov o delovnih nalogah. Obrazec se razlikuje glede na to, kateri gumb na obrazcu Seznam in vnos nalog (slika 23) izberemo. Iz ekranske slike obrazca za knjiženje delovnih nalog (slika 24), je razvidno, da smo izbrali določeno nalogo in nato izbrali možnost *popravi*. Vsakič ko popravljamo podatke o določeni nalogi, se ti podatki prikažejo na obrazcu. V primeru, da hočemo vrednost katerega od polj spremeniti, se ob pritisku na gumb *popravljam* najprej izvede kontrola vsebine, nato pa se podatki zapišejo v podatkovno zbirko. Kontrola vsebine nekaterih polj poteka tudi med vpisovanjem podatkov in se vrši, ko polja zapuščamo (on exit).

Popravljam nalogo

Naloga št.: 15005

Datum vnosa: 10. 2. 2000 Rok: 28. 2. 2000

Popravljam

Odgovorna oseba: 0677 Maurič Jadran Išči

dokončano

22. 2. 2000

Stroškovno mesto: 450 PROIZVODNO TEHNIČNO PODROČJE

Vir: KOREKTIVNI UKREPI PO REKLAMACIJAH

Prioriteta: 3 - pome

Naziv: Pakiranje sedežev v garniturah in odprema

Opis naloge: Pri garniturah za kupce TOM in Razgoršek se pri nekaterih garniturah sedeži pakirajo posebej. Ker ti sedeži niso označeni z etiketami, se dogaja, da jih skladiščniki pozabijo in ne odpremijo k kupcu

Število vrstic: 1

Cilj: Urediti odpreme z računalniško podporo

Opis rešitve: Sedeži so že v odpoklicu vnešeni ločeno in je s tem zagotovljeno, da so ločeni tudi v planu odpreme

Slika 24: Ekranska slika obrazca za popravljanje delovnih nalog

Če želimo v podatkovno zbirko dodati podatke o novi delovni nalogi, se prikaže podoben obrazec, le da je prazen, vse podatke je potrebno vanj vpisati (slika 25). Kot datum vnosa program predlaga tekoči datum.

The screenshot shows a window titled "Dodajam Nalogo" with the following fields and controls:

- Window title: Dodajam Nalogo
- Task number: Naloga št.: 15010
- Submission date: Datum vnosa: 25. 5 . 2006
- Deadline: Rok: 31. 5 . 2006
- Responsible person: Odgovorna oseba: [input field]
- Cost center: Stroškovno mesto: [input field]
- Source: Vir: [input field]
- Priority: Prioriteta: [input field]
- Task name: Naziv: [input field]
- Task description: Opis naloge: [input field]
- Number of rows: Število vrstic: 1
- Goal: Cilj: [input field]
- Resolution description: Opis rešitve: [input field]
- Buttons: Dodajam, Išči, dokončano (checkbox)

Slika 25: Ekranska slika obrazca za dodajanje delovnih nalog

4.2.5. Poročilo o delovnih nalogah

Do poročila o delovnih nalogah (slika 26) pridemo, če na obrazcu *Seznam in vnos* nalog pritisnemo gumb *izpis*. V poročilo so vključeni vsi bistveni podatki o delovnih nalogah. Oblika poročila je povzeta iz starega programa Naloge. Poročilo se, kot je iz njega tudi razvidno, uporablja kot dokument za prevzem oziroma potrditev zaključka posamezne naloge.

PLAMA-PUR d.d.
 Proizvodnja in predelava plastičnih mas
 Podgrad 17
 6244 P ODGRAD
 Služba kakovosti
 +386 5 714 9240

Naloga št.: 15005

Naziv: Pakiranje sedežev v garniturah in odprema

Vir: Urediti odpreme z računalniško podporo
 Opis: Pri garniturah za kupce TOM in Razgoršek se pri nekaterih garniturah sedeži pakirajo posebej. Ker ti sedeži niso označeni z etiketami, se dogaja, da jih skladiščniki pozabijo in ne odpremiijo k kupcu

Cilj: Urediti odpreme z računalniško podporo

Nosilec: Maurič Jadran Sodelujoči: Žgur Irena

Tematsko podr.: 4
 Org. enota: 450
 Rok za izvedbo: 15.3.2000

Naloga dokončana: Končano
 Datum rešitve: 15.3.2000
 Opis rešitve: Sedeži so že v odpoklicu vnešeni ločeno in je s tem zagotovljeno, da so ločeni tudi v planu odpreme

Nalogo prevzel: _____ Odobril: _____
 Rešitev realiziral: _____ Odobril: _____

Slika 26: Ekranska slika poročila o delovnih nalogah

5. UMEŠTITEV NOVEGA PROGRAMA NALOGE V INFORMACIJSKI SISTEM PODJETJA

V tem poglavju bomo najprej opisali testiranje novega program Naloga in pripravo dokumentacije o njem ter predstavili izkušnje uporabnikov in njihove predloge za izboljšanje programa. Nato se bomo posvetili interakciji programa Naloga z informacijskim sistemom podjetja, nazadnje pa bomo opisali učinke novega programa.

5.1. Testiranje programa

Ob uporabi razvojnega programskega okolja Borland Delphi je testiranje funkcij programa potekalo že med programiranjem. To je omogočil Borland Delphi z izredno hitrim prevajanjem izvirne kode. Ker prevajanje poteka hitro, si lahko že med programiranjem ogledamo in preizkusimo obrazce in vanje vgrajene funkcije. Na ta način smo odpravili veliko napak, ki bi jih v nasprotnem primeru opazili šele pri uporabi novega programa.

Čeprav je bila funkcionalna specifikacija določena že nekaj mesecev pred začetkom razvoja programske podpore za spremljanje delovnih nalog in je razvita programska podpora zadostila vsem zahtevam funkcionalne specifikacije, so uporabniki že v prvih dneh redne uporabe predlagali določene izboljšave. Nekaterim prošnjam uporabnikov smo lahko z manjšimi posegi takoj ugodili, nekatere pa bi zahtevale veliko več časa, kajti potrebne bi bile večje spremembe programa, tem pa bomo ugodili v prihodnosti.

Manjše pomankljivosti ali napake v novem programu Naloga, ki so se pojavile v redni uporabi, smo odpravljali sproti, kadar so nas na to opozorili uporabniki. Takšnih napak je bilo v novem programu naloga zelo malo, predvsem zaradi možnosti testiranja programa že med programiranjem.

5.2. Uporabniška dokumentacija

Z željo, da bi uporabnikom olajšali prehod na nov program za spremljanje delovnih nalog, smo pripravili navodila za uporabo novega programa Naloga. Ta vsebujejo

opise ključnih funkcij in ekranske slike novega programa z navodili za delo. Uporabnikom so na voljo v elektronski obliki.

5.3. Uporaba programa in izkušnje uporabnikov

Novi program Naloge je v redni uporabi in trenutno (maj 2006) deluje na dveh računalnikih. Program, kot smo že omenili, omogoča vpisovanje in pregledovanje podatkov o delovnih nalogah. Pravica vnosa oziroma branja podatkov o delovnih nalogah je odvisna od uporabnika, ki je v program Naloge prijavljen. Tako ima oseba, ki je zadolžena za nadzor in spremljanje delovnih nalog, možnost pregledovanja in dodajanja delovnih nalog. Drugi uporabniki pa imajo le možnost vpogleda v seznam delovnih nalog.

Uporabniki so pri novem programu Naloge pohvalili povezavo s skupnimi podatkovnimi zbirkami podjetja, saj jim tako ni več treba vpisovati podatkov o odgovornih osebah in organizacijskih enotah. Še posebej navdušeni so bili nad obrazcem Seznam in vnos nalog (slika 23), saj jim vgrajeni filtri omogočajo preglednejše sezname delovnih nalog. Uporabniki so izrazili tudi željo, da bi bil program širše dostopen. Po možnosti naj bi ga uporabljala vsaka organizacijska enota. Tako bi novi uporabniki pridobili možnost vnosa in branja podatkov o delovnih nalogah, vendar vsak za svojo organizacijsko enoto.

Uporabniki med pregledovanjem delovnih nalog najpogosteje uporabljajo filter po odgovornih osebah (slika 27), saj ponavadi iščejo eno od nalog posamezne odgovorne osebe. Prav tako uporabniki pogosto med delovnimi nalogami iščejo s pomočjo polja *naziv vsebuje* (slika 28).

Seznam in vnos nalog

Od datuma: 21. 2. 2000 Do datuma: 25. 5. 2006

nedokončane dokončane

Naziv vsebuje:

Število vrstic: 20

Odgovorna oseba: 0891 Gorjanc Bojan Išči

Stroškovno mesto:

Vir:

datum	naziv	Cilj	Vir	štm	odg_oseba
21.7.2000	Notranje presoje	Vključiti člane tima Q v notranje presoje	PREGLED S STRANI VODSTVA	470	0891 Gorjanc Bojan
21.7.2000	Korektivni ukrepi	Preveniti ukrepe po nosilcih	PREGLED S STRANI VODSTVA	470	0891 Gorjanc Bojan
21.7.2000	Korektivni ukrepi	Prenos vodenja ukrepov na računalnik	PREGLED S STRANI VODSTVA	470	0891 Gorjanc Bojan
21.7.2000	Drojda BG/05/00	Redno izvajati preglede orodij v vseh enotah.	KOREKTIVNI UKREPI PO REKI	453	0891 Gorjanc Bojan
9.9.2003	Rek. D030109 - luknjice PLS 3842	Ponovni dogovor s kupcem na temo luknjice na eterskih penah	KOREKTIVNI UKREPI PO REKI	110	0891 Gorjanc Bojan
9.9.2003	Rek. D030110 - se ne povrne ITS 2440	Ta tip pene se ne stiska	KOREKTIVNI UKREPI PO REKI	110	0891 Gorjanc Bojan
10.9.2003	Rek. D030114 - luknjičasta pena PLS 384	Ponovni dogovor s kupcem na temo luknjice na eterskih penah	KOREKTIVNI UKREPI PO REKI	110	0891 Gorjanc Bojan
14.10.2003	Rek. D030129 - višina, širina blokov	Dogovor Bojan Gorjanc in Ciril Kastelic glede višin in širin	KOREKTIVNI UKREPI PO REKI	110	0891 Gorjanc Bojan
14.10.2003	Rek. I030058 - višine blokov	Dogovor Bojan Gorjanc in Ciril Kastelic glede višin	KOREKTIVNI UKREPI PO REKI	110	0891 Gorjanc Bojan
14.10.2003	Rek. I030061 - neustrezna zlepljenost čist	Nastavili kontrolno tehnologijo za izdelavo čistilnih gobic v novem obratu.	KOREKTIVNI UKREPI PO REKI	140	0891 Gorjanc Bojan
21.10.2003	Rek. D030134 - pena se ne povrne	Dogovor s kupcem da se pena ITS 2842K, ITS 2440K in ITS 2540 ne stiskajo	KOREKTIVNI UKREPI PO REKI	110	0891 Gorjanc Bojan
21.10.2003	Rek. I030062 - neutr. dim. art. 72353	Opozorilo kooperantom, SK kontrola kooperantov	KOREKTIVNI UKREPI PO REKI	141	0891 Gorjanc Bojan
28.10.2003	Rekl. D030136 - luknjičasta pena ITS 254	Definirali sistem in odgovorne osebe za obveščanje v primeru ugotovljene neustreznosti	KOREKTIVNI UKREPI PO REKI	150	0891 Gorjanc Bojan
29.10.2003	Rekl. I030063 - odstopanje dim gobic	Opozorilo kooperantom, SK kontrola izdelkov narejenih v kooperaciji.	KOREKTIVNI UKREPI PO REKI	141	0891 Gorjanc Bojan
11.11.2003	Rek. D030142-145 razpke blokov PT 254	Najti generalno rešitev za te reklamacije	KOREKTIVNI UKREPI PO REKI	110	0891 Gorjanc Bojan
11.11.2003	Rek. D030148 -149 - neustrezne širine in S	Najti generalno rešitev za te vrste reklamacij	KOREKTIVNI UKREPI PO REKI	110	0891 Gorjanc Bojan
11.11.2003	Rek. I030066-069 - neustrezna višina in S	Najti generalno rešitev za te vrste reklamacij	KOREKTIVNI UKREPI PO REKI	175	0891 Gorjanc Bojan
18.11.2003	Rek. I030070-071 - pena FN 1826 se ne	Sestanek na temo nevrtačenje pene za CARTON PACK	KOREKTIVNI UKREPI PO REKI	450	0891 Gorjanc Bojan
10.3.2004	Rek. I040006 - neustrezna dolžina	Preverjanje dolžin rol na krožni žagi (vrh, sredina, dno) dogovor Bojan Gorjanc z Loric	KOREKTIVNI UKREPI PO REKI	150	0891 Gorjanc Bojan
4.5.2004	Rek. D040045 - luknjice PLS 3842	Dogovor s kupcem glede nadaljnega razreza tega tipa pene. Izdelati pokalkulacijo za	KOREKTIVNI UKREPI PO REKI	110	0891 Gorjanc Bojan

Opis naloge: dokončano

Opis rešitve: V plan presoj za leto 2000 so vključeni člani tima Q.

Slika 27: Ekranska slika obrazca seznam in vnos z iskanjem po odgovorni osebi

Seznam in vnos nalog

Od datuma: 21. 2. 2000 Do datuma: 25. 5. 2006

nedokončane dokončane

Naziv vsebuje: rek. D03011

Število vrstic: 3

Odgovorna oseba: Išči

Stroškovno mesto:

Vir:

datum	naziv	Cilj	Vir	štm	odg_oseba
9.9.2003	Rek. D030110 - se ne povrne ITS 2440	Ta tip pene se ne stiska	KOREKTIVNI UKREPI PO REKI	110	0891 Gorjanc Bojan
10.9.2003	Rek. D030114 - luknjičasta pena PLS 384	Ponovni dogovor s kupcem na temo luknjice na eterskih penah	KOREKTIVNI UKREPI PO REKI	110	0891 Gorjanc Bojan
16.9.2003	Rek. D030117 - neustrezna hitrost gorenj	Za ta tip pene, za tega kupca (TTO) se dodaja aditiv za samougasljivost, preveri logi	KOREKTIVNI UKREPI PO REKI	110	1075 Jagodnik Sandi

Opis naloge: dokončano

Opis rešitve: Dogovor za TTO in OPD in NT, da te pene ne stiskamo več za tega kupca.

Slika 28: Ekranska slika obrazca seznam in vnos z iskanjem po vsebini naziva

5.4. Interakcija z informacijskim sistemom podjetja

Interakcija novega programa Naloge z informacijskim sistemom podjetja se kaže zlasti v povezavi s skupnimi podatkovnimi zbirkami podjetja, iz katerih program črpa podatke za svojo podatkovno zbirko. To so predvsem podatki o odgovornih in sodelujočih osebah, torej delavcih. Podobno so vključeni tudi podatki o organizacijskih enotah.

Interakcija se ne kaže le v eni smeri, ampak so tudi podatki o delovnih nalogah na voljo drugim programom podjetja. Eden takih je program Kadrovska evidenca, katerega izdelava se je časovno prepletala z izdelavo novega programa Naloge. V program Kadrovska evidenca so vključeni tudi podatki iz podatkovne zbirke novega programa Naloge, predvsem podatki o odgovornih osebah in dokončanih oziroma nedokončanih delovnih nalogah. Tako je v Kadrovski evidenci mogoče videti število dokončanih in nedokončanih delovnih nalog ter razmerje med njimi za posamezno odgovorno osebo. To pa je le ena izmed možnosti uporabe podatkov iz podatkovne zbirke novega programa Naloge. Predvidevamo, da bo takih povezav z drugimi programi v prihodnosti še več.

5.5. Učinki programa

Novi program Naloge deluje na osnovi strežnika Microsoft SQL Server, kar v primerjavi s starim programom v veliki meri povečuje njegovo zanesljivost in zmanjšuje možnost izgube podatkov. Na ta način smo pridobili tudi možnost povezave podatkov iz podatkovne zbirke novega programa Naloge z drugimi programi podjetja. Prav tako je strežnik omogočil arhiviranje podatkov in povezavo podatkovnih zbirk z informacijskim sistemom podjetja, kar omogoča uporabo skupnih šifrantov podjetja. Tako smo podatke poenotili in zmanjšali njihov obseg. Obenem smo se izognili zamudnemu vpisovanju podatkov v šifrante programa, saj se nahajajo v skupnih podatkovnih zbirkah podjetja.

Z uporabo strežnika Microsoft SQL Server smo pridobili tudi večuporabniško delovanje programa in osnovo za uporabo spletnega dostopa do podatkov o nalogah s pomočjo tehnologije Microsoft ASP.NET. Kot večuporabniško delovanje razumemo

branje iz podatkovne zbirke in zapisovanje vanjo, ki jo istočasno izvaja več uporabnikov.

Uporabniški vmesnik novega programa Naloge je uporabniku prijaznejši, saj mu s pomočjo vgrajenih filtrov omogoča prijaznejše, predvsem pa hitrejše komuniciranje s podatkovno zbirko. Že samo iskanje delovnih nalog po datumih vnosa, odgovornih osebah, statusu dokončnosti, stroškovnem mestu in virih lahko zelo olajša delo. V starem programu je ta možnost sicer obstajala, vendar se podatki, ki so se pojavljali na filtrih, niso osveževali glede na podatke v podatkovni zbirki.

6. ZAKLJUČEK

V tem poglavju, bomo najprej povzeli opravljeno delo, predstavljeno v prejšnjih poglavjih, in opisali izkušnje z razvojnimi orodji, nato se bomo posvetili možnosti uporabe programa v drugih podjetjih ter definirali nadaljnji razvoj programa Naloge.

6.1. Opravljeno delo

Delo na podatkovnih zbirkah, je temeljilo na uvažanju podatkovni zbirk starega programa Naloge v strežnik Microsoft SQL Server ter prilagajanju zbirk strežniku. Nekatere tabele smo v celoti odstranili, druge pa le prilagodili in vzpostavili relacije med njimi. Določili smo tudi pravice dostopanja uporabnikov do podatkovne zbirke. S pomočjo orodja Borland Delphi smo nato izdelali uporabniški vmesnik, ki preko različnih obrazcev omogoča pregledovanje, popravljanje in vpisovanje podatkov o delovnih nalogah v podatkovno zbirko novega programa Naloge. Program smo sistematično testirali in ga izpopolnili do te mere, da je nadomestil stari program in je sedaj v redni uporabi v podjetju.

Celoten razvoj novega programa Naloge je moje lastno delo, prav tako pa tudi vsi opisani postopki v 4. poglavju. Za izdelavo novega programa, ki obsega okoli 4500 vrstic izvorne kode, sem potreboval okoli 150 ur časa. Ker je bil to moj prvi program izdelan z orodjem Borland Delphi, sem prepričan, da bi naslednjega lahko izdelal v primerno krajšem času.

6.2. Izkušnje z razvojnimi orodji

Obe orodji, tako strežnik Microsoft SQL Server kot razvojno okolje Borland Delphi sta se izkazali kot zelo zanesljivi in stabilni. Ker je prevajanje kode v okolju Delphi zelo hitro, je mogoče ob programiranju program tudi zaganjati in opazovati nastajajoči izdelek. Ta lastnost orodja Borland Delphi v praksi zelo pomaga pri delu. Čeprav ima zelo dobro podprto pomoč pri delu, pa smo pri njej najbolj pogrešali možnost prikazovanja lokacije napake. Orodje seveda sporoči tip napake in ime obrazca na katerem se napaka pojavlja. Ker imamo lahko v določenem obrazcu tudi več tisoč vrstic izvorne kode, pa je iskanje celo najpreprostejše napake zelo zamudno.

Pri strežniku Microsoft SQL Server smo s funkcionalnega vidika pogrešali le relacijo med tabelami podatkovnih zbirk dveh strežnikov. Glede na izkušnje z orodjem pa lahko trdimo, da gre za zelo zanesljiv strežnik.

6.3. Uporabnost v drugih podjetjih

Uporaba programske opreme za spremljanje delovnih nalog je smiselna v večini podjetij, kjer se srečujejo z večjim številom delovnih nalog. Novi program Naloge bi bilo vsekakor mogoče uporabiti v drugih podjetjih. Potrebno bi bilo le prilagoditi povezave do podatkovne zbirke oziroma vzpostaviti podatkovno zbirko s strukturo, ki jo uporablja program Naloge. Tako bi bile potrebne le majhne spremembe v programu. Problematične bi bile le povezave do odgovornih oseb oziroma organizacijskih enot, ki bi jih bi morali prilagoditi informacijskemu sistemu novega podjetja. Medtem ko uporabniškega vmesnika novega programa Naloge skorajda ne bi bilo potrebno spreminjati. Spremeniti bi bilo potrebno le posamezne povezave do podatkovne zbirke programa. Seveda bi bilo mogoče z večjimi spremembami do potankosti prilagoditi program izbranemu podjetju.

6.4. Nadaljnje delo

V bližnji prihodnosti načrtujemo nadgradnjo programa Naloge z uporabo spletne tehnologije Microsoft ASP.NET, ki bo delovala na osnovi strežnika Microsoft SQL Server in spletnega strežnika Microsoft IIS (Internet Information Services). Spletni vmesnik bomo izdelali s pomočjo programa Borland Delphi 2006, ki podpira tehnologijo Microsoft ASP.NET. Preko spletnega vmesnika bomo določenim uporabnikom dovolili vpogled v podatkovno zbirko delovnih nalog. Uporabniki bodo sprva imeli le možnost vpogleda v zbirko, kasneje pa predvidoma tudi možnost vnosa delovnih nalog.

Razviti program Naloge bomo v prihodnosti v podjetju Plama-pur d.d. uporabljali v večjem obsegu. Načrtujemo namestitev programa v vsaki organizacijski enoti. Uporabnik iz vsake organizacijske enote bo imel možnost vpogleda, ne pa tudi knjiženja. Z manjšo spremembo programa bomo lahko uporabnikom posameznih organizacijskih enot omogočili tudi zapisovanje podatkov o delovnih nalogah, vendar le za njihovo organizacijsko enoto. Pravica zapisovanja podatkov za vse

organizacijske enote bo najverjetneje ostala nespremenjena in bo na voljo le osebi, ki je zadolžena za nadzor in spremljanje delovnih nalog v podjetju.

7. LITERATURA

Borland Delphi 5 – Client/Server Foundations (1999). Inprise Corporation.

Borland Delphi 5 – Developer's Guide (1999). Inprise Corporation.

Borland Delphi 5 – Object Pascal Language Guide (1999). Inprise Corporation.

Kostrevc, L. (2000). Uvod v programiranje v programskem okolju Delphi. Ljubljana: Pasadena.

Mohorič, T. (1995). Uvod v podatkovne baze. Ljubljana: BI-TIM.

Plama-pur d.d. Pridobljeno 9.5.2006 s svetovnega spleta:

Programming a Microsoft SQL Server 2000 Database (2000). Microsoft Corporation.

System Administration for Microsoft SQL Server 7.0 (1998). Microsoft Corporation.