

Programi za logistike

Borut Jereb

Dejan Skok

Mirica Šafran

Mateja Škornik

Verzija 10.12.

Univerza v Mariboru
Fakulteta za logistiko
Laboratorij za informatiko

Celje, december 2010

Avtorji:

JEREB, Borut; SKOK, Dejan; ŠAFRAN, Mirica; ŠKORNIK, Mateja.

Naslov:

PROGRAMI ZA LOGISTIKE

Strokovni sodelavec in oblikovanje ovitka:

PLUT, Jalen

Ilustrator:

KOTNIK, Špela

Izdajatelj:

Univerza v Mariboru, Fakulteta za logistiko, Laboratorij za informatiko

Prva izdaja: december 2010, Celje

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

658.78

658.286

PROGRAMI za logistike / Borut Jereb ...[et al.] ; [ilustrator
Kotnik Špela]. - 1. izd. - Celje : Fakulteta za logistiko, 2010

ISBN 978-961-6562-44-7

1. Jereb, Borut

253434112

CIP - Kataložni zapis o publikaciji
Narodna in univerzitetna knjižnica, Ljubljana

656(0.034.2)

658.788(0.034.2)

PROGRAMI za logistike [Elektronski vir] / Borut Jereb ...[et
al.] ; ilustrator Kotnik Špela. - 1. izd. - El. knjiga. - Celje :
Fakulteta za logistiko, 2010

Način dostopa (URL): <http://labinf.fl.uni-mb.si/p4L/>

ISBN 978-961-6562-45-4

1. Jereb, Borut

253506560

Knjiga je urejena s programom L^AT_EX. T_EX je blagovna znamka American Mathematical Society.

To delo je objavljeno pod licenco Creative Commons: "Priznanje avtorstva – Nekomercialno – Brez predelav (verzija 2.5. in več)". Besedilo licence je na voljo na internetnem naslovu <http://www.creativecommons.si>, ali pa na naslovu: Inštitut za intelektualno lastnino, Streliška 1, 1000 Ljubljana.

Pri pošiljanju predlogov za spremembe in dopolnitve te publikacije se STRINJAM Z NASLEDNJO IZJAVO: V kolikor bo Uredniški svet publikacije upošteval moje predloge sprememb publikacije in bodo le te dodane v novejšo verzijo publikacije, se odpovedujem vsem materialnim avtorskim pravicam, ki izhajajo iz mojega avtorskega dela in se strinjam z objavo mojega imena med avtorji publikacije. Naslov za pošiljanje predlogov je programizalogistike@gmail.com.

Naslednji posamezniki so s številnimi popravki, predlogi in drugimi pripombami pomagali izboljšati to knjigo:

Grebenc Damjan
Smolinger Aleksandra

Predgovor

V Laboratoriju za informatiko na Fakulteti za logistiko Univerze v Mariboru smo pripravili knjigo, v kateri smo opisali uporabo prosto dostopnih programskih rešitev. Knjiga je namenjena študentom, predavateljem, inženirjem in ostalim, ki se srečujejo s potrebo po reševanju logističnih izzivov. Želimo pokazati, kako si je mogoče z uporabo prosto dostopne programske opreme olajšati delo na učinkovit način. Gre za način, ali bolj rečeno pristop, v katerega tisoči iz tako imenovane "skupnosti" vlagajo najboljše, kar premorejo. Prispevajo predvsem zaradi svojih prepričanj in delo največkrat opravijo v okviru svojega prostega časa. Zato so rešitve večkrat zelo posebne in imajo v sebi zrna genialnosti, ki jih bolj kot kakršen koli zaslužek motivira priložnost, da uresničijo svoje zamisli.

Pri opisovanju smo se oprli na poenostavljen logistični primer upravljanja vmesnega skladišča pri proizvodnji avtomobilov. Primer, ki ga uporabljamo tudi sicer pri študijskem procesu, opisuje skladišče, v katerem so avtomobilske gume, platišča in vijaki. Skladišče polnimo z izdelki treh dobaviteljev, ki uporabljajo vsak svoje transportno sredstvo (železnica, tovornjak s prikolico in manjše dostavno vozilo), ki vnašajo svoje posebnosti, med katerimi je zelo pomembna časovna dimenzija dobav. Praznimo ga v relativno rednih časovnih intervalih z viličarji, ki zagotavljajo nemoteno oskrbo z avtomobilskimi deli v proizvodnji. Skladišče ima omejeno zmogljivost. Za podjetje predstavljajo skladiščeni avtomobilski deli vezan kapital, ki se s časom dinamično spreminja. Količine se nikoli ne smejo spustiti pod minimalne vrednosti, saj bi bila v tem primeru ogrožena celotna proizvodnja, kar se ne sme zgoditi. Primer ne zahteva posebnega logističnega predznanja in je primeren tudi za študente začetnike in inženirje z drugih področij.

V knjigi smo opisali uporabo šestnajstih programov, vezanih na primer. Ti predstavljajo prav toliko različnih dimenzij problematike upravljanja vmesnega skladišča polizdelkov. Vsak primer podaja svoje poglavje, ki ga je mogoče prebrati neodvisno. Obdelali smo področje upravljanja procesov (projektno vodenje, diagrami poteka, sledljivost izdelkov), vizualizacije (vizualizacija oskrbne verige, vizualizacija geografskega področja in uporaba različnih prostorskih GIS slojev), prostorskega planiranja, odločanja (na osnovi iskanja optimalnih vrednosti, večparameterskega odločitvenega modeliranja, simulacij in finančnih tokov), napovedovanja verjetnega razvoja ter navsezadnje statistične analize.

Vsako poglavje na kratko opisuje teoretično ozadje posamezne dimenzije problema,

katerega upravljanje podpremo z ustreznim orodjem. Nadaljujemo z opisom problema skozi optiko obravnavane dimenzije problema. Sledi kratek opis programskega orodja. Primer njegove uporabe je vezan na primer vmesnega skladišča. Na koncu je vsakemu poglavju dodan kratek povzetek napisanega in seznam virov, ki smo jih pri pisanju poglavja uporabljali. Pred vsemi poglavji je še kratek opis prosto dostopnih programskih orodij in samega primera, ki predstavlja rdečo nit celotne knjige.

Že od vsega začetka je osnovna predpostavka knjige njeno nenehno izboljševanje. Zavedamo se, da je mogoče in potrebno v knjigi marsikaj izboljšati in dodati. Zato ste vabljeni vsi, ki imate predloge v zvezi s knjigo, da nam pišete na naslov programizalogistike@gmail.com. Vse predloge bomo skrbno proučili in vaša uporabna spoznanja vnesli v naslednje verzije. Predvsem si želimo, da bi knjiga s pomočjo "skupnosti" postala tako dobra, da bi jo bilo smiselno prevesti tudi v druge jezike.

Ideja za nastanek "prosto dostopne knjige" v lasti "skupnosti" je zorela kar nekaj časa. Uresničila se je takoj, ko smo s posluhom dekana Fakultete za logistiko Univerze v Mariboru, rednega profesorja dr. Martina Lipičnika, vzpostavili okolje za izvedbo projekta. Za nastanek prve verzije se posebej zahvaljujem podiplomskima študentoma Dejanu Skoku in Mirici Šafran, ki sta z izjemno ustvarjalnostjo, življenjsko močjo in zagnanostjo pripravljala ideje, iskala rešitve in jih tudi zapisala. V nastajanje besedila se je aktivno vključila tudi doktorska študentka Mateja Škornik. Vabilu za pomoč pri izdelavi ilustracij se je odzvala likovna pedagoginja Špela Kotnik in knjigo obogatila s svojimi sporočilnimi slikami. Doktorski študent Jalen Plut je pomagal pri tehnični realizaciji. Projekta in s tem tudi knjige ne bi bilo, če ne bi v preteklih desetletjih imel ob sebi kolega Petra Hitija, s katerim si deliva prepričanje o vlogi odprtokodne "skupnosti" in mnoge ure vročih debat o fenomenu programske opreme nasploh. Zahvaljujem se tudi Kseniji in svojim trem otrokom, ki so tudi zaradi mojega ukvarjanja z odprtokodnimi rešitvami v prostem času, prikrajšani za mnoge pozornosti, ki bi jih sicer lahko bili deležni.

Borut Jereb

Kazalo

1	UVOD	23
1.1	Študija primera	24
1.2	Opis problema	24
1.3	Proces aktivnosti	25
1.4	Zbrani podatki	28
1.5	Cilji	30
1.6	Metodologija	30
1.7	Kako do programskih orodij?	31
1.8	Področja raziskovanja	35
1.9	Nadomestljivost plačljivih programskih orodij	37
1.10	Kategorije programske opreme	41
1.10.1	Prosta programska oprema	41
1.10.2	Odprikodna programska oprema	41
1.11	Vrste licenc	44
1.11.1	Licenca GNU GPL	44
1.11.2	Druge odprtokodne in prosto dostopne licence	47
1.11.3	Študentska licenca	48
1.11.4	Licenca za prosto verzijo	48
1.12	Kako uporabljati programska orodja?	49
2	PLANNER - načrtovanje aktivnosti	53
2.1	Teoretično ozadje	54
2.1.1	Tehnologija črtne kode	54
2.1.2	Ganttov diagram	54
2.2	O programskem orodju	57
2.3	Uporaba	59
3	DIA - Načrtovanje diagramov	69
3.1	Teoretično ozadje	70
3.1.1	Diagram poteka/aktivnosti	70
3.1.2	Standard UML	70
3.2	O programskem orodju	75
3.3	Uporaba	86

4	ZINT- generator črtnih kod	93
4.1	Teoretično ozadje	94
4.1.1	Tehnologija črtne kode	94
4.2	O programskem orodju	97
4.3	Uporaba	99
5	ASDN - integracija oskrbne verige	107
5.1	Teoretično ozadje	108
5.1.1	Oskrbna veriga	108
5.2	O programskem orodju	109
5.3	Uporaba	116
6	GOOGLE ZEMLJA - načrtovanje poti z digitalnim zemljevidom	131
6.1	Teoretično ozadje	132
6.1.1	Digitalni zemljevidi	132
6.2	O programskem orodju	132
6.3	Uporaba	139
7	QUANTUM GIS - geografsko informacijsko planiranje	147
7.1	Teoretično ozadje	148
7.1.1	Geografsko informacijski sistem	148
7.2	O programskem orodju	149
7.3	Uporaba	164
8	QCAD - 2D prostorsko načrtovanje	179
8.1	Teoretično ozadje	180
8.1.1	Računalniško podprto načrtovanje CAD	180
8.2	O programskem orodju	182
8.3	Uporaba	184
9	SIMPLE WAREHOUSE MAPPER- 3D simulacijsko orodje	195
9.1	Teoretično ozadje	196
9.1.1	3D vizualizacija prostora	196
9.2	O programskem orodju	197
9.3	Uporaba	201
10	PETERSEN - odločanje na podlagi teorije grafov	211
10.1	Teoretično ozadje	212
10.1.1	Teorija grafov	212
10.2	O programskem orodju	213
10.3	Uporaba	224

11 LINDO - optimizacija stroškov	239
11.1 Teoretično ozadje	240
11.1.1 Linearno programiranje	240
11.1.2 Metoda simpleksov	241
11.2 O programskem orodju	241
11.3 Uporaba	247
12 DEXI - odločitveni model	257
12.1 Teoretično ozadje	258
12.1.1 Večparametrski odločitveni model	259
12.2 O programskem orodju	262
12.3 Uporaba	272
13 GNUCASH - finančno načrtovanje	287
13.1 Teoretično ozadje	288
13.1.1 Finančno načrtovanje	289
13.2 O programskem orodju	291
13.3 Uporaba	299
14 GPSS WORLD – simulacija “dogajanja” v skladišču	309
14.1 Teoretično ozadje	310
14.2 O programskem orodju	311
14.3 Uporaba	319
15 SCILAB - numerično reševanje in analiziranje podatkov	363
15.1 Teoretično ozadje	364
15.1.1 Napovedovanje povpraševanja	364
15.1.2 Kvantitativne metode napovedovanja	364
15.2 O programskem orodju	365
15.3 Uporaba	378
16 Oo PREGLEDNICA - gibanje zalog	391
16.1 Teoretično ozadje	392
16.1.1 Elektronske preglednice	392
16.2 O programskem orodju	392
16.3 Uporaba	401
17 PSPP - statistična analiza podatkov	409
17.1 Teoretično ozadje	410
17.1.1 Statistična analiza podatkov	410
17.2 O programskem orodju	410
17.3 Uporaba	419
18 DOSTOP DO PROGRAMSKIH ORODIJ	431

19	PREVODI	435
20	PRILOGA	445
20.1	Scilab	446
20.1.1	Brownov model	446
20.1.2	Holtov model	447
20.1.3	Regresijski model	449
20.1.4	Funkcija napovedovanja (Forecast)	451
20.2	GPSS World	453
20.2.1	Model	453
20.2.2	Vhodni podatki	456

Slike

1.1	Opis procesa dela	27
1.2	Zbrani podatki	29
1.3	Zagon programskega središča Ubuntu	31
1.4	Programsko središče Ubuntu	32
1.5	Nameščanje ali odstranjevanje programskega orodja	32
1.6	Upravljanje paketov Synaptic	33
1.7	Nameščanje programskega orodja	34
1.8	Source forge	34
1.9	Povezave med kategorijami programske opreme	42
1.10	Odpri tokodni produkti	42
1.11	Odpri tokodni produkti	43
1.12	Ubuntu	44
1.13	Program Wine	50
1.14	Namestitev programskega orodja s pomočjo programa Wine	51
1.15	Dovoljenje za odpiranje datoteke kot program	51
1.16	Nastavitev za odpiranje z Wine Windows Program Loader	52
1.17	Odpiranje z Wine Windows Program Loader	52
2.1	Spletna stran	58
2.2	Prenos programskega orodja	58
2.3	Programsko okno	59
2.4	Zagon novega projekta	60
2.5	Shranjevanje novega projekta	60
2.6	Odpiranje shranjenega projekta	61
2.7	Koledarske nastavitve	61
2.8	Zapis nove aktivnosti	62
2.9	Določitev trajanja aktivnosti	63
2.10	Povezava aktivnosti	64
2.11	Dodajanje povezave aktivnosti	64
2.12	Oblika povezave med dvema aktivnostma	65
2.13	Kritična pot	65
2.14	Pomoč	66
2.15	Ganttov diagram	66

2.16	Izvoz dokumentov	67
2.17	Pregledovanje dokumenta v HTML obliki s pomočjo spletnega brskalnika	68
3.1	Zgodovinski razvoj specifikacije poenotnega modelnega jezika UML	71
3.2	Pet pogledov na sistem UML	72
3.3	Odvisnost	74
3.4	Asociacija in agregacija	74
3.5	Generalizacija	74
3.6	Realizacija	74
3.7	Urejevalnik diagramov Dia	76
3.8	Prenos programskega orodja Dia	77
3.9	Menijska vrstica	77
3.10	Orodna vrstica - Datoteka	78
3.11	Nastavitev strani	78
3.12	Prikaznih osnovnih in specifičnih orodij	79
3.13	Osnovno okno	80
3.14	Orodna vrstica - Pogled (Windows)	81
3.15	Orodna vrstica - Pogled (Ubuntu Linux)	81
3.16	Listi in predmeti	82
3.17	Možnosti oblik orodij	82
3.18	Specifična orodja	83
3.19	Dodajanje predmetov na platno	84
3.20	Oblikovanje in povezovanje predmetov	84
3.21	Dodajanje besedila v izbrani predmet	85
3.22	Lastnosti diagrama	85
3.23	Pomoč	86
3.24	Orodja UML	87
3.25	Lastnosti diagrama	88
3.26	Diagram aktivnosti z UML	89
3.27	Drevo diagrama	90
4.1	Uvodna spletna stran	98
4.2	Simbologija črtne kode ITF-14.	100
4.3	Izbira črtne kode ITF-14	102
4.4	Kreirana črtna koda	102
4.5	Izbira barvne podlage	103
4.6	Izbira velikosti črtne kode	103
4.7	Generiranje zaporedja	104
4.8	Izbira oblike dokumenta shranitve generiranega zaporedja	105
4.9	Slike generiranega zaporedja črtne kode	105
5.1	Agile Supply Demand Network	110
5.2	Osnovno okno ASDN	110
5.3	Vozlišča in povezave med njimi	112

5.4	Izpolnitev procesa	114
5.5	Atributi za vozlišča	115
5.6	Orodna vrstica	115
5.7	Uporaba programa ASDN	116
5.8	Dodajanje povezav in puščic	117
5.9	Dodajanje povezav in puščic	117
5.10	Izračun zahtevnejših vrednosti – 1. del	118
5.11	Izračun zahtevnejših vrednosti – 2. del	119
5.12	Zapis vrednosti atributov – 1. del	120
5.13	Zapis vrednosti atributov - 2. del	121
5.14	Izbira prevoza	122
5.15	Atributi povezav	122
5.16	Izbira zemljevida	123
5.17	Prikaz na zemljevidu Evrope	123
5.18	Grafični prikaz rezultatov	124
5.19	Ganttov diagram	125
5.20	Zaloge	125
5.21	Vezani kapital na dan	126
5.22	Grafi po meri – 1. del	126
5.23	Grafi po meri – 2. del	127
5.24	Raven storitev	127
5.25	Tabela vozlišč	128
5.26	Transportna tabela	128
5.27	Finančni podatki	129
5.28	Preračun	129
5.29	Pregled opravljenega dela - scenarij	130
6.1	Prenos programskega orodja Google Zemlja	133
6.2	Ogled programskega orodja	134
6.3	Nastavitev zgodovinskih posnetkov	135
6.4	Merjenje razdalj	135
6.5	Orodna vrstica - Datoteka	136
6.6	Orodna vrstica - Urejanje	137
6.7	Orodna vrstica - Pogled	137
6.8	Zagon simulatorja letenja	138
6.9	Simulator letenja	138
6.10	Načrtovanje poti	140
6.11	Iskanje podjetja	141
6.12	Navigacijski gumb	141
6.13	Sprememba reliefnega pogleda	142
6.14	Dodajanje slojev	142
6.15	Dodani sloji	144
6.16	Triglavska cesta X	145

7.1	Rastrski in vektorski podatki	149
7.2	Namestitev programa – 1. del	150
7.3	Namestitev programa – 2. del	151
7.4	Programsko okno	152
7.5	Menijska vrstica – 1. del	153
7.6	Menijska vrstica – 2. del	154
7.7	Orodna vrstica	154
7.8	Preklopi na urejanje	155
7.9	Spletna stran ARSO	155
7.10	Pridobivanje podatkov na WFS ARSO	156
7.11	Agencija Republike Slovenije za okolje	156
7.12	Enostavni izvoz WFSCClient UI	157
7.13	Izbira sloja	158
7.14	Izhodni format	159
7.15	Dodajanje slojev	160
7.16	Prikaz lastnosti slojev	161
7.17	Properties - General	161
7.18	Properties - Symbologyl	162
7.19	Properties - Labels	163
7.20	Properties - Attributes	163
7.21	Število prebivalcev v MOC - januar 2010	165
7.22	Naselja v MOC	165
7.23	Pogled v atributno tabelo cest v MOC	166
7.24	Dodajanja imena cest, ulic in poti	166
7.25	Dodajanja slojev hišnih števil	167
7.26	Spreminjanje oznak	167
7.27	Dodajanje števil stanovanj	168
7.28	Gostota naselitve stanovanj	168
7.29	Stopnja naselitve v bližini Teharja	169
7.30	Zavarovana območja narave	169
7.31	Ekološko pomembna območja	171
7.32	Hidrografska območja	172
7.33	Karta potresne nevarnosti - projektni pospešek	172
7.34	Karta potresne nevarnosti - splošno	173
7.35	Hitrost vetra	173
7.36	Povprečna letna višina korigiranih padavin	174
7.37	Območje onesnaženosti zunanjega zraka	175
7.38	Atributna tabela za območje onesnaženosti zunanjega zraka	175
7.39	Poplavna območja	176
7.40	Hrup, ki ga povzroča železniški promet	177
8.1	Spletna pomoč za uporabo programskega orodja Qcad	183
8.2	Izbira oblike shranitve datoteke	184

8.3	Shranjevanje datoteke s skico	185
8.4	Odpiranje datoteke s skico	186
8.5	Izvažanje datotek	186
8.6	Dodajanje novih blokov	187
8.7	Možnost povečave	187
8.8	Povečava slike z uporabe miške	187
8.9	Zaklepanje seznama	188
8.10	Izbira oblike črte	189
8.11	Natančnost risanja	189
8.12	Natančnost risanja (točke)	190
8.13	Merjenje razdalj	190
8.14	Dodajanje teksta na skico	191
8.15	Brisanje črt	191
8.16	Razveljavitev ukaza	192
8.17	Pomoč	192
8.18	Skica skladiščnega prostora	193
9.1	Spletna stran Simple Warehouse Mapper	198
9.2	Programsko orodje Simple Warehouse Mapper	199
9.3	Spletna navodila za namestitev programskega orodja	200
9.4	Nov dokument	202
9.5	Izbor dokumenta	202
9.6	Menijska in orodna vrstica	203
9.7	Sprememba nastavitve velikosti grafične mreže	203
9.8	Določitev velikosti grafične mreže	204
9.9	Vstavljanje besedila	204
9.10	Različni predmeti za vizualizacijo prostora	205
9.11	Kopiranje predmetov	205
9.12	Povečanje/zmanjšanje pogleda	205
9.13	Kako shraniti datoteko	206
9.14	Oblika shranjene datoteke	206
9.15	Izvoz datoteke	207
9.16	Pomoč	207
9.17	Praktični prikaz vizualizacije procesa skladiščenja	208
10.1	Prenos Jave	214
10.2	Prenos Petersen-ovega programskega orodja	215
10.3	Menijska vrstica	215
10.4	Določitev števila točk – Null Graph	216
10.5	Določitev števila točk – Complete	216
10.6	Complete – Complete Bipartite Graph	216
10.7	Complete-Complete Tripartite Graph	217
10.8	Circuit Graph	217
10.9	Wheel	218

10.10	Windmills	218
10.11	Full n-ary Tree	219
10.12	Različne vrste grafov – 1. del	220
10.13	Različne vrste grafov – 2. del	221
10.14	Menijska vrstica in razdelek Graph	222
10.15	Dodatna pomoč pri uporabi programa	222
10.16	Dodatna literatura na temo minimalnega vpetega drevesa	223
10.17	Klik na direktno povezavo	223
10.18	Posamezna programska okna	224
10.19	Shranjevanje datoteke	225
10.20	Izris grafa s petimi točkami in povezavami	226
10.21	Dodajanje uteži	226
10.22	Zapis uteži	227
10.23	Vrednosti povezav	228
10.24	Možnosti grafa – 1. del	228
10.25	Možnosti grafa – 2. del	229
10.26	Možnosti grafa – 3. del	229
10.27	Properties- Statistics	230
10.28	Matrika sosednosti glede na vrednosti	231
10.29	Matrika sosednosti glede na povezave	231
10.30	Kromatično število	232
10.31	Minimalno vpeto drevo	233
10.32	Minimalno vpeto drevo – Kruskalov algoritem	234
10.33	Postavitev grafa	235
10.34	Graf z omejitvami	236
11.1	Prenos programa Linda – 1. del	242
11.2	Prenos programa Linda – 2. del	243
11.3	Osnovno okno ob zagonu	244
11.4	Menija Datoteka in Urejanje	245
11.5	Meni Solve	245
11.6	Meni Window in Help	246
11.7	Bližnjice (pomen)	246
11.8	Orodna vrstica	247
11.9	Primer napačno zapisanega modela	248
11.10	Shranjevanje datoteke	249
11.11	Odpiranje datoteke	250
11.12	Spletna stran za prenos datotek, ki so v pomoč pri uporabi programa Lindo	254
12.1	Večparametrski odločitveni model	260
12.2	Prenos programskega orodja DEXi	264
12.3	Kako odpremo že ustvarjeni dokument	265
12.4	Primer večparametrskega modela za izbiro viličarja	266
12.5	Izdelava poročila rezultatov vrednotenja	267

12.6	Nastavitve funkcij programskega orodja	267
12.7	Kako shraniti datoteko	268
12.8	Možni načini shranjevanja	268
12.9	Funkcije razdelka Urejanje	269
12.10	Pomoč	270
12.11	Model (primer za izbiro viličarja)	273
12.12	Drevo kriterijev	273
12.13	Določitev uteži kriterijev za izbiro viličarja	274
12.14	Vnos podatkov za attribute	274
12.15	Zaloga vrednosti	275
12.16	Določanje zaloge vrednosti	276
12.17	Funkcija koristnosti	276
12.18	Odločitvena pravila za funkcijo koristnosti	277
12.19	Vrednotenje kriterijev in variant	279
12.20	Analiza in primerjava variant	280
12.21	Primerjava dveh variant	280
12.22	Poročilo o primerjavi variant	281
12.23	Grafični prikaz vrednotenja vrednosti parametrov	282
12.24	Izbira kriterijev za primerjavo	282
12.25	Drevo kriterijev	283
12.26	Določitev zaloge vrednosti	283
12.27	Tabela odločitvenih pravil	284
12.28	Povprečne uteži kriterijev	284
12.29	Grafični prikaz vrednotenja parametrov vseh treh variant	285
13.1	Računovodska enačba	290
13.2	Spletna stran GnuCash	292
13.3	Namestitev programskega orodja GnuCash	293
13.4	Nastavitve	294
13.5	Kreiranje in vnos podatkov	295
13.6	Uvoz datotek	295
13.7	Izvoz podatkovnih datotek	296
13.8	Orodja – 1. del	296
13.9	Orodja – 2. del	297
13.10	Poročila	297
13.11	Pomoč	298
13.12	Zagon programa GnuCash	300
13.13	Nov dokument	300
13.14	Glavna stran	301
13.15	Prihodki	301
13.16	Odhodki	302
13.17	Najem posojila	302
13.18	Sredstva	302

13.19	Lastniški kapital	303
13.20	Pregled računov	304
13.21	Izbira poročila denarnega toka	305
13.22	Glavna knjiga	305
13.23	Osveževanje podatkov	306
14.1	Spletna stran GPSS World	312
14.2	Prenos programskega orodja GPSS World	313
14.3	Primer blokovnega diagrama za izbrano simulacijo	320
14.4	Klicanje tekstovnih datotek v GPSS	321
14.5	Osnovni podatki	330
14.6	Kako odpremo datoteko	331
14.7	Zagon datoteke v GPSS	331
14.8	Zagon simulacije	332
14.9	Transakcije	332
14.10	Določitev številčno izražene vrednosti imena za operiranje transakcij v izbrani simulaciji	332
14.11	Čakalne vrste	337
14.12	Proces skladiščenja	338
14.13	Izbira pogleda poročila	343
14.14	Grafični prikaz stroškov skladiščenja platišč	344
14.15	Osnovni podatki	346
14.16	Sprememba standardnega odklona za dostavo pnevmatik	348
14.17	Čakalne vrste za Situacijo 1	349
14.18	Vhodne količine komponent za Situacijo 1	349
14.19	Stroški skladiščenja platišč za Situacijo 1	354
14.20	Spremembe pri intervalih porazdelitve gibanja zalog – Situacija 1	355
14.21	Sprememba intervalnega časa za dostavo platišč	355
14.22	Čakalne vrste za Situacijo 2	355
14.23	Vhodne količine komponent za Situacijo 2	356
14.24	Strošek skladiščenja platišč za Situacijo 2	361
14.25	Spremembe pri intervalih porazdelitve gibanja zalog – Situacija 2	362
15.1	Uradna spletna stran Scilab	366
15.2	Odpiranje programskega orodja v Ubuntu	367
15.3	Zagon programa	368
15.4	Menijska in orodna vrstica v Scilabu	368
15.5	Menijska vrstica – 1. del	369
15.6	Menijska vrstica – 2. del	369
15.7	Okno za zapis programa	370
15.8	Razdelek Xcos	371
15.9	Razdelek Pomoč	373
15.10	Menijska vrstica – 3. del	373
15.11	Demo programi	374

15.12	Orodna vrstica	374
15.13	Graf funkcije narisane v Scilab	377
15.14	Odpiranje datoteke	379
15.15	Izpis rezultatov	382
15.16	Povpraševanje za sedanje obdobje	386
15.17	Povpraševanje, model in predikcija z upoštevanjem standardnega od- klona – regresija	387
15.18	Povpraševanje, model in predikcija z upoštevanjem standardnega od- klona - Holt	388
16.1	Prenos programskega paketa OOo	393
16.2	Naslovna stran OOo	394
16.3	Osnovno okno	394
16.4	Naslovna vrstica	394
16.5	Menijska vrstica	395
16.6	Menijska vrstica - Datoteka, Uredi in Pogled	396
16.7	Menijska vrstica – Vstavi, Oblika in Orodja	397
16.8	Menijska vrstica – Podatki, Okno	398
16.9	Menijska vrstica Pomoč	398
16.10	Standardna orodna vrstica	399
16.11	Vrstica z orodji	399
16.12	Vrstica stanja	400
16.13	Vnosna vrstica	400
16.14	Stanje zalog na skladišču	401
16.15	Izračun prometa in stanja posameznih komponent	403
16.16	Zasedenost skladišča s posameznimi komponentami	404
16.17	Oblikovanje celic	405
16.18	Določanje veljavnosti	406
17.1	Osnovno okno	412
17.2	Menijska vrstica File	412
17.3	Menijska vrstica Edit	413
17.4	Menijska vrstica Edit	413
17.5	Menijska vrstica Data	414
17.6	Orodna vrstica Transform	414
17.7	Menijska vrstica Analyze v Windows okolju	415
17.8	Menijska vrstica Analyze v Ubuntu Linux	416
17.9	Razdelek v menijski vrstici Help	416
17.10	Anketni vprašalnik	418
17.11	Priprava baze Variable View	419
17.12	Tip spremenljivke	420
17.13	Opis spremenljivk	420
17.14	Manjkajoče vrednosti	421
17.15	Pripravljena baza	422

17.16Vnos podatkov	422
17.17Frekvenčna statistika	423
17.18Izpis rezultatov frekvenčne statistike	424
17.19Opisna statistika	424
17.20Rezultati opisne statistike - Ubuntu in Windows	425
17.21Rezultati opisne statistike - Windows	425
17.22Razdelek Crosstabs	425
17.23Rezultati povezanosti dveh spremenljivk	426
17.24Možnosti izbire analiz	427
17.25T-test za dva odvisna vzorca	427
17.26Rezultati T-testa za dva odvisna vzorca	428
17.27Razdruževanje	428
17.28Rezultati razdruženih datotek	429

Tabele

1.1	Programska orodja – 1. del	39
1.2	Programska orodja – 2. del	40
1.3	Programska orodja	49
2.1	Tabela vseh aktivnosti in njihovih trajanj	56
3.1	Vrste diagramov	75
4.1	Razlaga simbologije črtne kode ITF-14	100
4.2	Primer črtne kode	101
4.3	Ukazi za generiranje zaporedja črtne kode	101
5.1	Atributi za vozlišča	112
5.2	Drugi atributi	113
5.3	Pomen oznak orodne vrstice	113
12.1	Orodja	265
12.2	Osnovni podatki	271
12.3	Kriteriji za izbiro viličarja	272
12.4	Orodja pri opredeljevanju variant	278
12.5	Orodja pri vrednotenju kriterijev in variant	279
14.1	Osnovni elementi za simulacijo v GPSS	315
14.2	Analitičen izračun časovnih intervalov	318
14.3	Kapacitete prevoznega sredstva	318
14.4	Kapacitete prevoznega sredstva	318
14.5	Bloki in stavki	322
14.6	Informacije o simulaciji	331
14.7	Imenovanje in vrednosti blokov	332
14.8	Analiza poteka simulacije	336
14.9	Analiza poteka simulacije	336
14.10	Analiza poteka simulacije	336
14.11	Analiza poteka simulacije	337
14.12	Čakalne vrste	337

14.13	Skladišče	338
14.14	Gibanje zalog	343
14.15	Variabilni podatki	347
14.16	Zanima nas	347
14.17	Podatki	347
14.18	Sprememba variabilnih podatkov	348
14.19	Sprememba variabilnih podatkov	353
15.1	Ikone orodne vrstice	375
15.2	Statistični podatki prodaje v obdobju 1999-2009	378
16.1	Izračun prometa in stanja posameznih komponent	402
16.2	Zasedenost skladišča s posameznimi komponentami	402
18.1	Dostop do programskih orodij	433
19.1	Prevodi A-C	437
19.2	Prevodi C-E	438
19.3	Prevodi E-G	439
19.4	Prevodi G-L	440
19.5	Prevodi L-O	441
19.6	Prevodi P-S	442
19.7	Prevodi S-T	443
19.8	Prevodi T-W	444

Poglavje 1

UVOD



1.1 Študija primera

Obvladovanje logističnih tokov zajema načrtovanje, izvedbo in kontroliranje premikanja ter razmeščanja blaga in s tem povezanih podpornih aktivnosti v okviru sistema, ki sledi specifičnim ciljem. V današnjem času je razvoj logističnih sistemov napredoval do te mere, da je za obvladovanje celotnega sistema oskrbne verige potrebno upoštevati še ostale vidike (management, informatika, komunikacije, optimizacijske metode...).

V preteklosti so se obvladovanja logističnih tokov lotevali brez sodobnih pristopov - temeljil je zgolj na človeškem umu. Sodobna tehnologija omogoča razvoj informacijske tehnologije in telematike, s podporo katere lahko integriramo celostno oskrbno verigo.

Problema obvladovanja logističnih tokov se lahko študentje in podjetja lotijo na več različnih načinov, z različnimi pristopi in znanji. Uporaba prosto dostopnih programskih orodij in njihova integracija v poslovne procese omogoča sistematično reševanje izbranega logističnega problema, predvsem z vidika optimizacije, vizualizacije, simulacije in projektiranja. Prednost tovrstnih orodij je v njihovi dostopnosti, enostavnosti, funkcionalnosti in uporabi na različnih operacijskih sistemih. Z njihovo pomočjo lahko bistveno izboljšamo obvladovanje notranjih in zunanjih logističnih tokov.

1.2 Opis problema

Izvajanje logističnih procesov v oskrbni verigi avtomobilske proizvodnje zahteva znanja iz področja managementa, upravljanja, optimizacije, komunikacije in informacijskih znanj. Integracija vseh deležnikov in področij oskrbne verige rezultira v učinkoviti in optimalni izvedbi vseh procesov. Temelj učinkovitega sodelovanja med njimi je pravilna zasnova procesov, nalog in odgovornosti (npr. informacijski, materialni in finančni tokovi).

Funkcije in naloge so med deležniki oskrbne verige jasno določene in razdeljene. Vsak deležnik mora upoštevati navodila in standarde, in sicer tako, da ne vpliva na ostale procese (zmanjšanje učinkovitosti in zanesljivosti) oskrbne verige. Pri tem mora upoštevati predhodni in naslednji korak. Del učinkovitega izvajanja logističnih aktivnosti predstavlja uporaba informacijske podpore in tehnologije. Izmenjava informacij v realnem času omogoča učinkovito načrtovanje, izvedbo, kontrolo in ukrepanje v primeru pojava nepredvidenih dogodkov.

V celotni oskrbni verigi, kjer sodelujejo ponudniki, dobavitelji, prevozniki, proizvajalci, kupci in potrošniki, se izbrani logistični problem nanaša le na integracijo dobaviteljev, prevoznikov in proizvajalcev. Vključuje obvladovanje in izvajanje logističnih procesov nabave, transporta, prevzema, skladiščenja in odpreme, ki so del celotne oskrbne verige.

Izbran logistični problem v avtomobilski proizvodnji vključuje različna področja in procese:

- planiranje proizvodnih kapacitet glede na trend povpraševanja prodaje novih vozil;
- proces nabave komponent (vijaki, pnevmatike in platišča);
- izbira transporta in transportnih poti (pnevmatike transportiramo z vlakom, paltišča s tovornjakom in vijake s kombijem);
- proces skladiščenja (komponente skladiščimo na treh različnih in ločenih lokacijah v centralnem skladišču);
- proces prevzema in odpreme;
- upravljanje tveganj;
- finančno načrtovanje;
- proces odločanja (večparametrski odločitveni model);
- pregled tveganj lokacijske strategije s pomočjo prostorskega planiranja;
- simulacija materialnih tokov v skladišču;
- statistična analiza podatkov;
- vizualizacija oskrbne verige;
- informacijska tehnologija za označevanje komponent.

1.3 Proces aktivnosti

Različni dejavniki, med katere lahko prištevamo gospodarsko stanje države, finančno stanje družbe in ostale družbene vidike, vplivajo na povpraševanje po novih vozilih. Povečanje povpraševanja po novih vozilih posledično vpliva na obseg proizvodnje vozil in povečanju ponudbe na avtomobilskem trgu.

Povečanje obsega proizvodnje vozil sorazmerno spiralno zajame vse ostale logistične procese v makro in mikro okolju. K makro okolju lahko prištevamo globalne vidike (povečanje gostote prometa in števila transportov, negativen vpliv na okolje, gospodarska rast itd.), k mikro okolju pa prištevamo interne procese podjetij (planiranje procesov nabave, proizvodnje in skladiščenja, razširitev skladiščnih in proizvodnih kapacitet, simulacije itd.).

Spremembe na avtomobilskem trgu imajo dokazano zelo velik vpliv na celotno dogajanje v gospodarstvu, saj ta sektor zajema ogromno število podjetij (avtomobilska industrija predstavlja veliko oskrbno verigo s številnimi deležniki). O vplivu avtomobilske industrije na gospodarstvo nazorno priča tudi zadnja svetovna gospodarska kriza - hitro ukrepanje in pomoč avtomobilski industriji je eden izmed izhodov iz krize. Kako vpliva sprememba ponudbe na trg in povpraševanje po

vozilih na poslovanje avtomobilskega podjetja? Povečanje povpraševanja spremeni ravnovesje na trgu in povzroči manjko pri ponudbi, s čimer se poveča tržna cena vozil, kar lahko negativno vpliva na nadaljnjo povpraševanje po vozilih. Podjetja morajo pravočasno ukrepati in povečati ponudbo - povečati proizvodne količine in s tem zagotoviti ravnovesje med ponudbo in povpraševanjem na avtomobilskem trgu.

Zagotovitev ravnovesja na avtomobilskem trgu sproži spremembo pri obsegu proizvodnih količin. Management podjetja je odgovoren, da nastalo situacijo temeljito preuči in poda navodila za nadaljnje delo. Želja po zagotovitvi in zadovoljitvi potreb kupcev po novih vozilih sproži signal po povečanju proizvodnih količin. Signal "potuje" v nabavni oddelek podjetja, kjer je potrebno zagotoviti ustrezno količino komponent. Pri izbiri dobavitelja komponent je potrebno upoštevati načela cene, kakovosti, dostavnega časa in ostalih pogojev. Način dostave naročenih komponent je stvar dogovora med dobaviteljem in kupcem. Transport z različnimi prevoznimi sredstvi omogoča optimizacijo stroškov, zmanjšanje tveganja in odvisnosti od vrste transportnega sistema. Izbira med železniškim ali cestnim transportom je odvisna od naročenih količin, stanja infrastrukturne, prevoznih stroškov itd. Dobavljene komponente se na podlagi ustrezne dokumentacije kvalitativno in kvantitativno pregledajo, prevzamejo, uskladiščijo in vnesejo v informacijski sistem. Informacije o dostavljenih komponentah se sporočijo nabavnem oddelku, ki o dostavi komponent obvesti proizvodni oddelek. Komponente se nato skladiščijo na ločenih lokacijah. Upravljanje z zalogami je pomembno, saj le te uravnavajo nihanja v proizvodnji. S tem zmanjšamo tveganje morebitnih zastojev proizvodne linije.

Upravljanje skladiščenja je umetnost uravnavanja informacijskih in materialnih tokov. Planiranje skladiščnih kapacitet je eno izmed težjih nalog, saj so lahko nihanja količin precejšnja. V praksi se teorija mnogokrat izkaže v negativni podobi. Polna skladišča vsekakor zagotavljajo nemoteno proizvodnjo in zmanjšujejo tveganje zastoja, vendar pa imajo tudi negativno stran. Zaloge so vezani kapital, ki predstavlja dodaten strošek, zato je potrebno poiskati optimalen nivo zalog, ki upošteva tako finančni kot tudi vidik tveganja.

Pomemben vidik v celotni oskrbni verigi je tudi upravljanje s tveganji, ki lahko bistveno pripomore k učinkoviti realizaciji vseh logističnih procesov.

1.4 Zbrani podatki

Na Sliki 1.2 so zbrani podatki o namišljenem podjetju OpenStorage, kjer se ukvarjamo s skladiščenjem komponent za proizvodnjo osebnih avtomobilov znamke X.

PODJETJE OpenStorage				
Naslov	Triglavska cesta X, 6264 Bohinjska Bistrica, Občina Bohinj			
DEJAVNOST				
Skladiščenje komponent za proizvodnjo avtomobilov (pnevmatike, platišča in vijaki)				
Količina komponent	4x pnevmatike, 4x platišča, 16x vijak; razmerje 1:1:4			
Vrsta transporta	Vlak, tovornjak in kombi			
Dobavitelji	Italija, Francija			
KOMPONENTA		NABAVNA CENA		
Platišča		35 €/kos		
Pnevmatika		60 €/kos		
Vijak		3 €/kos		
DOSTAVA				
Časovni interval dostave	7200 minut = 120 ur = 5 dni			
Frekvenca dostave	Urejena z naključno porazdelitvijo			
Komponente	Sredstvo	PRIHODI	Interval	Odstopanje +/-
Platišča	Tovornjak		180 min	10 min
Pnevmatika	Vlak		360 min	20 min
Vijak	Kombi		170 min	10 min
ODPREMA				
Komponente	Količina	ODHODI	Interval	Odstopanje +/-
Platišča	20 kosov		8 min	1 min
Pnevmatika	20 kosov		8 min	1 min
Vijak	80 kosov		8 min	1 min
<i>Opomba: skupaj sestavlja 5 avtomobilov</i>				
MANIPULACIJE				
Praznjenje viličarja v proizvodnji			Polnjenje viličarja pri prevzemu	
	Trajanje	Odstopanje +/-	Trajanje	
Platišča	6 min	0 min	6 min	
Pnevmatika	6 min	0 min	6 min	
Vijak	6 min	0 min	6 min	
Skladiščne kapacitete	<u>max</u>		Začetno stanje ob simulaciji	
Platišča	800 kosov		200 kosov	
Pnevmatika	800 kosov		200 kosov	
Vijak	2600 kosov		800 kosov	
KAPACITETA prevoznega sredstva	Optimalno	Investicija	Cena viličarja	
Tovornjak	390 kosov/platišča	Nakup viličarja	20.000 €	
Vlak	800 kosov/pnevmatika			
Kombi	1450 kosov/vijakov			

Slika 1.2: Zbrani podatki

1.5 Cilji

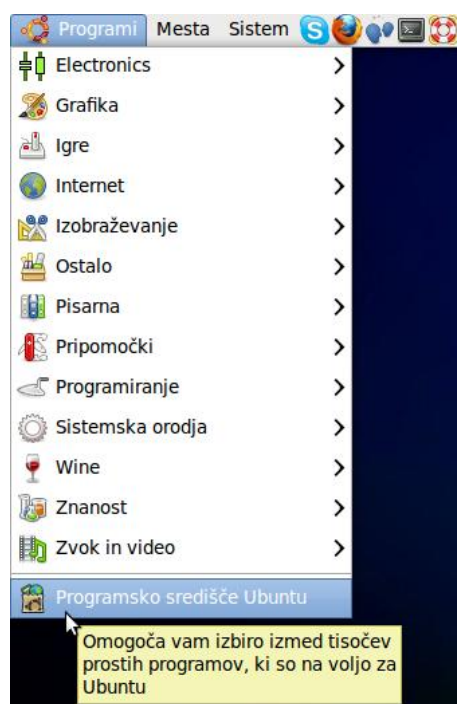
Cilj je prikaz uporabe različnih odprtokodnih in prosto dostopnih programskih orodij na primeru reševanja izbranega logističnega problema. Prikazati želimo pomen in smiselnost uporabe programskih orodij z dodanim teoretičnim pogledom na izbrani problem. Bodočim in že izkušenim logistikom želimo pokazati, kako se lotiti reševanja logističnega problema z različnih vidikov (načrtovanje, optimizacija, finance idr.). Želimo jim približati uporabnost tovrstnih programskih orodij, saj menimo, da v praksi niso dovolj prepoznavni. Zavedamo se potreb po povezanosti logističnega znanja z informacijsko tehnologijo, zato spodbujamo k raziskovanju in reševanju praktičnih problemov, z namenom dosega širšega pogleda na določeno problematiko.

1.6 Metodologija

Pri pripravi knjige uporabimo različno metodologijo dela. *Deduktivno metodo* uporabimo na področju, kjer iz splošnih stališč izvedemo posebne in posamične postavke iz katerih nato prehajamo do konkretnih posamičnih zaključkov. Z *metodo analiziranja* razčlenimo posamezna programska orodja na enostavnejše sestavne dele in prvine in se tako lotimo preučevanja vsakega dela. *Metoda sinteze* omogoča, da na podlagi raziskovanja in pojasnjevanja določene procese in postopke povežemo v smiselno celoto, v kateri so njeni deli vzajemno povezani. S pomočjo *statističnih metod* analiziramo posamezne probleme s prikazom grafikonov, tabel, slik. Pri enem izmed programskih orodij se lotimo osnovnih faz *metod modeliranja*, kjer izdelamo program za reševanje konkretnega problema v praksi. *Metoda proučevanja primerov* je metoda, ki jo uporabimo za preučevanje posamičnega primera iz izbranega področja. Z uporabo te metode na podlagi rezultatov opazovanj in zapisanih primerov izvedemo zakonitosti oz. zaključke.

Kaj pričakujemo od uporabnikov programskih orodij?

Od uporabnikov pričakujemo, da bodo z zanimanjem pristopili k branju knjige in jo morda, kot pripomoček uporabili pri reševanju problemov v praksi. Pričakuje se, da jih dana tematika odprtokodnih in prostodostopnih programskih orodij, navezujoča na logistične probleme, zanima ali pa so zgolj radovedni, kaj jim ponujamo. Želimo, da jim bo raziskovanje skozi posamezna področja prikazano na lahkotnejši in enostavnejši način ter da se pričnejo zavedati, da le ni vse plačljivo, ampak da lahko marsikateri problem razrešimo s pomočjo brezplačnih programskih orodij. Priročnik je namenjen študentom, kakor tudi ostalim uporabnikom, ki jih dana problematika zanima.



Slika 1.3: Zagon programskega središča Ubuntu

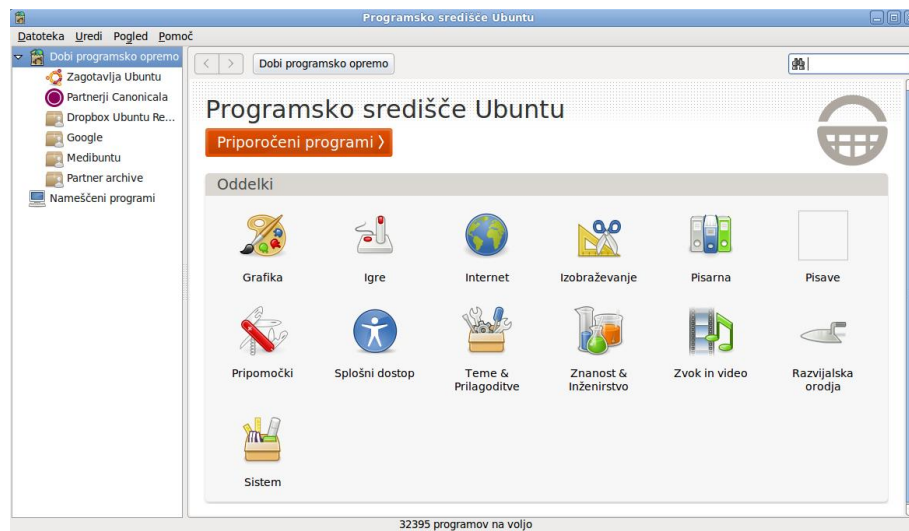
1.7 Kako do programskih orodij?

Dostop do programskih orodij je odvisen od izbranega operacijskega sistema, ki ga uporabljamo. Postopek dostopa v sistemu Ubuntu Linux je drugačen, kakor je le ta v okolju Windows. Za pričetek dela je potrebno namestiti operacijski sistem Ubuntu, ki je dostopen v več kot 25-ih različnih jezikih, tudi slovenščini. Pri dostopu do programskih orodij imamo na voljo dve različni možnosti. Prva omogoča, da v meniju *Programi* izberemo možnost *Programsko središče Ubuntu* (glej Sliko 1.3).

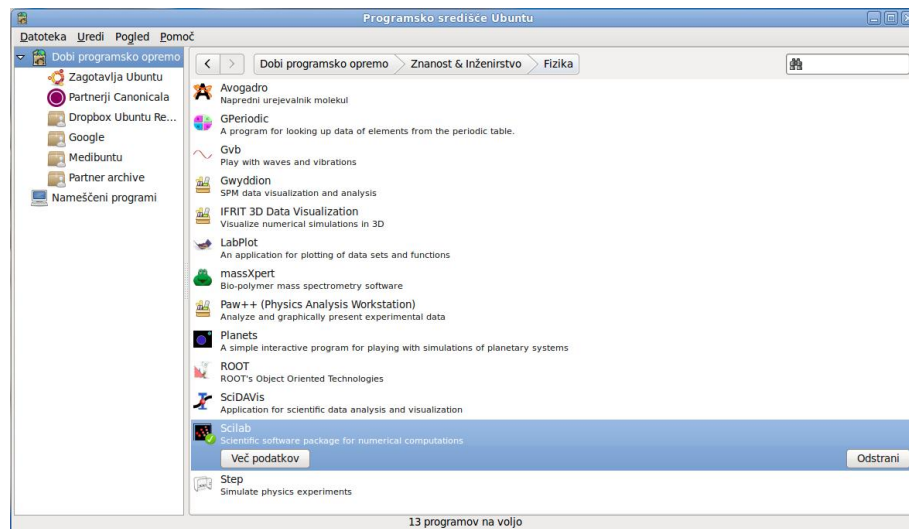
S klikom na dani meni se prikaže okno s priporočenimi programskimi orodji, razvrščenih v trinajst oddelkov. Gre za enega izmed načinov namestitve, ki je preprost in enostaven za uporabnika (glej Sliko 1.4).

Kot primer prikažemo namestitev programskega orodja Scilab. Programsko orodje poiščemo v oddelku *Znanost & Inženirstvo*, pododdelek *Fizika*. S klikom na gumb *Namesti* operacijski sistem namesti program na ustrezno mesto v osnovnem meniju *Programi*. V primeru, če bi želeli določen program odstraniti, ponovimo postopek iskanja določenega programskega orodja, ki ga odstranimo s klikom na *Odstrani* (glej Sliko 1.5).

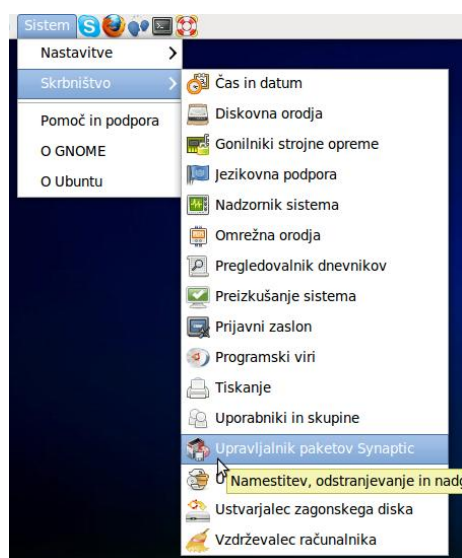
V primeru, da zelenega programa ne najdemo v izbranih oddelkih, so na voljo tudi druge možnosti namestitve. V orodni vrstici *Sistem* izberemo možnost *Skrbnišтво* in v



Slika 1.4: Programsko središče Ubuntu



Slika 1.5: Nameščanje ali odstranjevanje programskega orodja



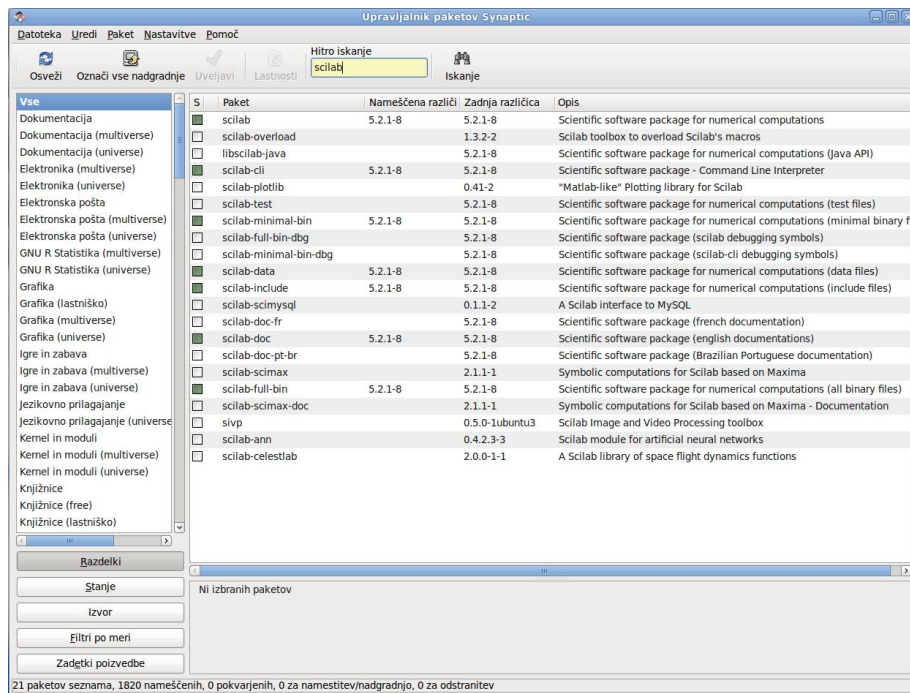
Slika 1.6: Upravljanje paketov Synaptic

nadaljevanju *Upravljanje paketov Synaptic* (glej Sliko 1.6).

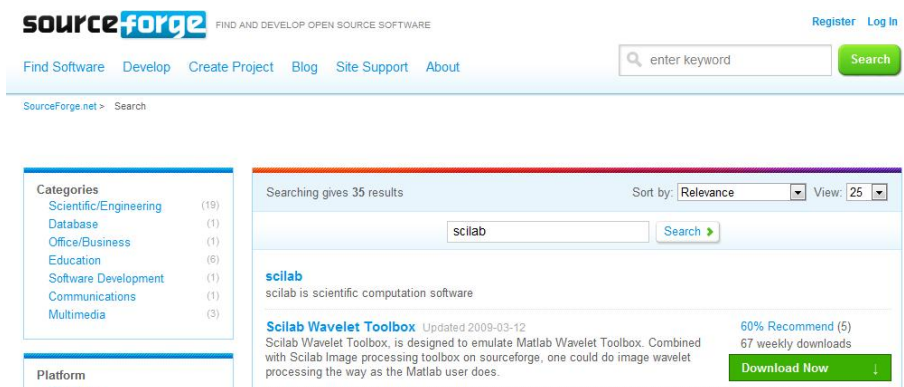
Z izbrano možnostjo lahko dostopamo do številnih drugih programskih orodij, datotek idr. Za pravilno delovanje Ubuntu-ovega sistema je potrebno namestiti številne datoteke povsem samostojno. Za iskanje programskih orodij v okence *Hitro iskanje* vnesemo naslov orodja, ki ga želimo namestiti. V izbranem primeru preverimo, ali lahko programsko orodje Scilab poiščemo še na drug način. V primeru, če še ni nameščen, ga namestimo s klikom na *paket Scilab*. Pozorni moramo biti predvsem, da se program namesti v celoti, brez prekinitev in sočasnega izvajanja posodobitev (glej Sliko 1.7).

V operacijskem sistemu Windows poteka nameščanje programskega orodja na drugačen način. Običajno si posamezna programska orodja namestimo preko njihove uradne spletne strani ali pa preko t.i. "source forge" spletne strani. Vse kar potrebujemo za namestitev programskega orodja, je torej dostop do interneta (glej Sliko 1.8).

Verzije programskih orodij se razlikujejo, odvisno kateri operacijski sistem uporabljamo in katero pot namestitve izberemo. V praksi se dogaja, da uporabniki operacijskega sistema Ubuntu ali drugih Linux operacijskih sistemov, ki namestijo programsko orodje s pomočjo upravljanja paketov Synaptic, dobijo starejšo a delujočo verzijo, ki se razlikuje od tistih, ki jih namestijo uporabniki operacijskega sistema Windows s spleta. V kolikor želimo imeti enake verzije programskega orodja v obeh operacijskih sistemih Ubuntu ali Windows, moramo preveriti verzijo, ki jo je namestil upravljalca paketov Synaptic in jo poiskati na spletu.



Slika 1.7: Nameščanje programskega orodja



Slika 1.8: Source forge

1.8 Področja raziskovanja

Programska orodja smo povezali v šest smiselno razdeljenih področij.

Področje 1 - Procesi

V področje *Procesi* umeščamo tri programska orodja. Prvo programsko orodje *Planner* je v pomoč pri projektnem vodenju, ki opredeljuje prikaz procesa načrtovanja, nadzorovanja in poročanja aktivnosti v podjetju. V fazi načrtovanja je smiselna izdelava Ganttovega diagrama, s pomočjo katerega vizualno prikažemo dogajanje oz. trajanje določenih nalog znotraj projekta.

V nadaljevanju uporabimo programsko orodje *Dia* in izdelamo diagram poteka, ki prikazuje prehode iz ene aktivnosti na drugo aktivnost. Vizualno prikažemo proces oskrbe z izbrano komponento v podjetju (proces naročanja, dostave, izmenjave dokumentov, grobega in finega prevzema platišč, reklamacij in skladiščenja).

Obvladovanje oskrbne verige in zagotovitev sledljivosti komponent na vhodni strani sistema in izdelkov na izhodni strani sistema, predvsem v procesu transporta in skladiščenja, zahteva poznavanje in uporabo standardov označevanja in identifikacije logističnih enot, kar omogoča programsko orodje *ZINT*, s katerim prikažemo implementacijo črtne kode v poslovanju izbranega podjetja. *ZINT* je ustrezen pripomoček za začetnike uvajanja črtnih kod v poslovne procese.

Področje 2 - Vizualizacija

V drugo področje umeščamo *Vizualizacijo* oskrbne verige. S programskim orodjem *ASDN Logistics Analysis* (v nadaljevanju *ASDN*) predstavimo integracijo oskrbne verige - od dobavitelja komponent do izbranega podjetja, saj je združevanje posameznih segmentov nadvse pomembno. *ASDN* je v pomoč pri oblikovanju industrijskih logističnih mrež, ki prispevajo k izboljšanju industrijskih omrežij z optimizacijo zalog in stroškov.

V vsakdanjem življenju se pogosto srečujemo s problemom mobilnosti in pomanjkanjem časa. Pogosto se zgodi, da se nenadoma odločimo za poslovno potovanje v tujino, čeprav niti ne vemo, kako bomo tja prispeli. V takšnih primerih uporabimo programsko orodje *Google Zemlja*, s katerim prikažemo načrtovanje poti z digitalnimi zemljevidi. Izmed številnih segmentov, ki jih ponuja programsko orodje, se osredotočimo na načrtovanje poti od namišljenega poslovnega partnerja v Italiji do našega podjetja.

V izbrano področje vključimo tudi programsko orodje *Quantum GIS*, ki omogoča ustvarjanje, vizualizacijo, poizvedbe in analizo geoprostorskih podatkov. S *Quantum GIS* na podlagi pridobljenih slojev (karta potresnega območja, karta poplavnega območja, karta hrupa ipd.) vizualno opredelimo primernost lokacije za izgradnjo skladiščnega objekta z upoštevanjem tveganj.

Področje 3 - Prostorsko planiranje

Tretje področje je področje *Prostorskega planiranja*, kamor umestimo programski orodji *Qcad* in *Simple Warehouse Mapper*. Logistika kot dejavnost se ukvarja z upravljanjem različnih tokov v oskrbni verigi ali znotraj podjetja. Pri tem zajema vse procese, ki se izvajajo in pri obvladovanju le teh potrebuje različna znanja. Mednje umestimo tudi poznavanje in uporabo *Qcad* orodja, ki pripomore k vizualizaciji problemov, prostorov ali procesov. Z njegovo pomočjo izrišemo skladišni prostor v 2-D skici, kjer so uskladiščene komponente za proizvodnjo avtomobila.

Z odprtokodnim programskim orodjem *Simple Warehouse Mapper* na podlagi že izdelanega skladišnega prostora prikažemo enostaven model ureditve in razporeditve skladišnega prostora. Uporablja preprosto 3D simulacijsko tehniko, znano kot izometrični zemljevid. Z njim ustvarimo vizualno podobo izbranega objekta s pripadajočimi podatki in omogočimo predstavitev dejanskega stanja v realnem svetu.

Področje 4 - Odločitve

Odločitve so pomembno področje na vseh področjih poslovanja v podjetju. Znotraj izbranega področja umestimo pet različnih programskih orodij. *Petersen* programsko orodje je praktično, enostavno in brezplačno s pomočjo katerega prikažemo izračun optimalne dostave komponent od dobavitelja do izmišljenega podjetja *OpenStorage*.

Zaradi velikega števila numeričnih operacij s programskim orodjem *Lindo* predstavimo problem izbire transporta platišč, ki jih je potrebno dostavljati v določenem obdobju. Programsko orodje uporabimo za optimizacijo stroškov z linearnim in celoštevilskim programiranjem. Izdelamo model, ki prikazuje optimalno izbiro prevoznega sredstva ob izbranih omejitvah.

V nadaljevanju z uporabo programskega orodja *DEXi* predstavimo odločitveni model, s katerim določimo optimalni izbor nakupa viličarja. Določimo kateri viličar je najboljša rešitev za naše potrebe in zahteve. Enostavna razčlenitev kriterijev, opredelitev različnih variant in njihovo vrednotenje poda rešitev izbire najboljšega. S kakovostno raziskavo in primerjavo lahko uspešno nastopimo pred vodstvom in predstavimo izbiro za nakup novega viličarja, katerega izberemo s pomočjo večparametrskega odločitvenega modela.

Pri poslovanju nikakor ne moremo mimo financ, ki so povezane z različnimi zahtevami, problemi in ovirami. S programskim orodjem *GnuCash* prikažemo vodenje računa pri investiciji v nakup sodobne tehnologije. Doseganje učinkovitosti in pretočnosti materialnega toka je odvisno od celotnega delovanja sistema oskrbne verige. Časovne zamude v fazi skladiščenja bistveno vplivajo na celotno pretočnost in podaljšanje procesov oskrbne verige. Hitro ukrepanje in investicija v nakup sodobne tehnologija, ponudita uporabniku možnost, tako optimizacije poslovanja, kot dosego višje pretočnosti materialnega toka in zmanjšanje stroškov.

Z uporabo *GPSS World* simulacijskega orodja je mogoče predvideti učinke modeliranja in izbire realnega kompleksnega sistema. Gre za celovito orodje za modeliranje simulacij dogodkov, ki ohranja simulacijsko okolje z dodano visoko

stopnjo interaktivnosti in vizualizacije. S simuliranjem materialnega toka treh različnih komponent (pnevmatike, platišča in vijaki) in spreminjanjem vrednosti (čas, količine) prikažemo pravilno izbiro modela oskrbne verige s finančnim vidikom spreminjanja vrednosti. Sistematično planiranje oskrbne verige v fazi izvedbe dostave in skladiščenja potrebnih komponent, zagotavlja učinkovitost z visoko stopnjo odzivnosti vseh deležnikov v oskrbni verigi, kar rezultira večjo pretočnosti materialnega toka.

Področje 5 - Napovedovanje

Z napovedovanjem ocenjujemo verjetne razvoje v prihodnosti. Planiranje ne oznanja samo verjetnega, ampak na osnovi verjetnega postavlja zaželeno. Peto področje je osredotočeno na *Napovedovanje*, predvsem s programskim orodjem *Scilab*. V namišljenem podjetju *OpenStorage* se ukvarjamo s skladiščenjem treh vrst komponent, namenjenih izdelavi osebnega avtomobila. Na podlagi zbranih podatkov o nakupu v preteklih 11 letih, izvedemo napoved prodaje avtomobilov za obdobje 5 let. Ob zapisu danega programa in dobljenih rezultatov preverimo ali je izbrana metoda pravilna oz. ali jo je za nadaljnjo delo potrebno spremeniti.

Skozi drugačen pogled s programskim orodjem *Open.Office.org Preglednica* (v nadaljevanju *OOo Preglednica*) izdelamo model s katerim prikažemo izračun stanja zalog v skladišču. Upoštevamo začetno stanje zalog, vrednost posamezne komponente in kapaciteto skladišča. Pri tem tedensko vodimo stanje treh komponent. Izdelan model omogoča prihodnjo napoved potreb po komponentah.

Področje 6 - Analiza

Statistična analiza podatkov je ena izmed pomembnih aktivnosti na vseh področjih raziskovanja. V podjetjih jo pogosto uporabljamo, kadar želimo analizirati določeno količino podatkov, se osredotočiti zgolj na en vidik posameznega procesa ipd. Običajno se izvaja na podlagi že zbranih podatkov ali z zbiranjem podatkov z anketnimi vprašalniki. V zadnjem šestem področju s programskim orodjem *PSPP* izvedemo analizo podatkov zbranih s pomočjo anketnega vprašalnika. Zanima nas zadovoljstvo zaposlenih v namišljenem podjetju *OpenStorage* z delovnim mestom v skladišču, natančneje na oddelku, ki se ukvarja s postopkom prevzema pnevmatik. Pomembno je, da se pri izbranem problemu ne osredotočimo zgolj na matematične izračune, ampak da pri izboljšanju delovnih navad prisostvuje tudi mnenje zaposlenih.

1.9 Nadomestljivost plačljivih programskih orodij

Prikazati želimo, da se je moč posluževati številnih odprtokodnih in prosto dostopnih programskih orodij povsem brezplačno, čeprav pri študentih še niso pogosto v uporabi. Prosto dostopna so že dlje časa na spletnih straneh, vendar so kljub temu mnogim še nepoznana. Bralca želimo skozi posamezna poglavja popeljati v določen segment logističnega problema, katerega z ustreznim programskim orodjem razrešimo in v

nadaljevanju prikažemo rezultate, jih natančneje opišemo ter smiselno povežemo v zaključeno celoto.

Na spletu so dostopna številna programska orodja, ki jih uporabimo v študijske namene in delo na projektih. Pogosto se zgodi, da potrebujemo določeno programsko orodje, katerega cene so študentom nedosegljive. S strani izobraževalnih ali drugih institucij je moč pridobiti študentske verzije, ki so pogosto necelostne, kar pomeni, da je programsko orodje moč uporabiti le z z določenimi omejitvami.

Tabeli 1.1 in 1.2 prikazujeta uporabljena programska orodja in primerjalna (nadomestna) programska orodja. Prikazuje še nekaj drugih programskih orodij, ki jih lahko preizkusite sami. Brezplačne programske rešitve so v praksi specializirane predvsem za določena področja in ne zajemajo širših področij, kot plačljive, zato v priročniku povežemo različna brezplačna programska orodja in se tako skušamo približati plačljivim.

Programska orodja				
Področje	Uporabljena	Druga brezplačna	Plačljiva	Uporaba
Procesi	Planner 0.14.4	Open Workbench	Microsoft Project	Načrtovanje projektov
	Dia 0.97.1	OOo Impress, Violet, StarUML	Microsoft Visio, Gliffy, SmartDraw	Načrtovanje diagramov
	ZINT 2.3.2	Online Barcode Generator	DRPU Barcode Label Maker Software	Generiranje črtne kode
Vizualizacija	ASDN 1.217	/	SAP	Integracija oskrbne verige
	Google Zemlja 5.1	TangoGPS, GoogleMaps, Viking, Marble, Earth 3D	/	Načrtovanje poti z digitalnimi zemljevidi
	Quantum GIS 1.01	Grass GIS, Saga GIS	GIS	Geografsko informacijski sistem
Prostorsko planiranje	Qcad 2.0.5	Archimedes, BRL-CAD	Autocad, ProEngineering, Archicad	Projektiranje objektov, lokacij v 2D dimenzijah
	Simple Warehouse Mapper	/	ArchiCAD, AutoCAD	Vizualizacija tokov v 3D dimenzijah

Tabela 1.1: Programska orodja – 1. del

Programska orodja				
Področje	Uporabljena	Druga brezplačna	Plačljiva	Uporaba
Odločitve	Petersen 3.2.3	GraphThing	Matlab 7.0.1	Odločanje na podlagi teorije grafov
	Lindo 6.1	/	Lingo	Optimizacija stroškov z linearnim in celoštevilskim programiranjem
	Dexi 3.02	JDexi	Decision Pad	Odločitveni modeli
	Gnucash 2.2.9	Grisbi, Eqnomize	Quicken, AceMoney	Računovodsko orodje za finančno načrtovanje
	GPSS 5.2.2.	Simula, JGPPS	SIMAN, Arena	Simulacija "dogajanje" v skladišču in opredelitev prihodov
Napovedovanje	Scilab	Maxima	Matlab	Numerično reševanje in analiziranje podatkov
	OOo Calc 3.2.0	/	Microsoft Excel	Uporaba preglednic za prikaz gibanja zalog
Analiza	PSPP 0.602	Google Docks	SPSS	Statistična analiza podatkov

Tabela 1.2: Programska orodja – 2. del

1.10 Kategorije programske opreme

Programsko opremo delimo v več različnih zvrsti - od povsem osnovnih do precej kompleksnih, ki so izdelane s pomočjo diagramov za večjo preglednost. Slika 1.9 prikazuje delitev programske opreme na vizualni podlagi.

1.10.1 Prosta programska oprema

Prosto programsko opremo definiramo kot opremo, ki jo lahko zastonj ali proti plačilu vsakdo uporablja, kopira in dopolnjuje. Prost program je potencialno mogoče vključiti v prost operacijski sistem, kot je GNU, ali proste različice sistema GNU/Linux. Besedo "prost" je potrebno strogo ločiti od besede "brezplačen" (ang. *Freeware*) - tovrstna oprema je zelo podobna odprtokodni programski opremi.

1.10.2 Odprtokodna programska oprema

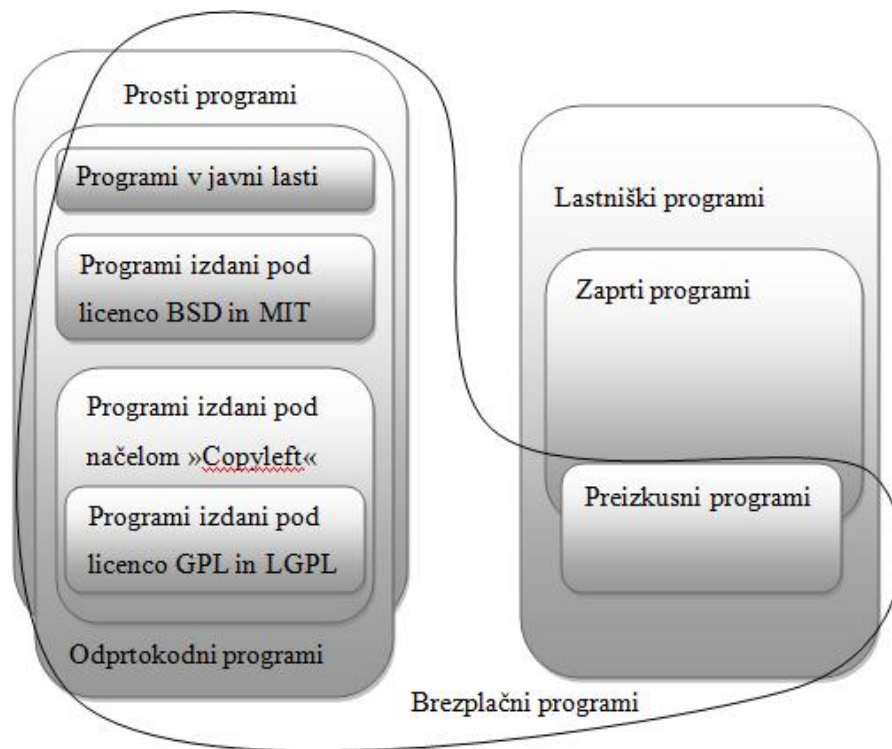
Razvoj odprtokodne programske opreme (*Open source software* - OSS) se prične kot marketinška kampanja za prosto programje. Lahko jo definiramo, kot programsko opremo, za katero je izvorna koda izdana pod zaščitno licenco (ali aranžma, kot npr. javna domena). Odprta koda dovoljuje vsakršno uporabo in spreminjanje programske opreme, njeno razširjanje v spremenjeni ali nespremenjeni obliki. Definiciji proste programske opreme in odprtokodne programske opreme sta skoraj identični. Bistvena razlika je v tem, da prva poudarja svobodno uporabo programske opreme, druga pa dostop do izvorne kode programske opreme [61].

Projekte OSS postavljajo in vzdržujejo prostovoljni programerji. Primeri odprtokodnih produktov so Apache HTTP Server, Internet Protocol in internetni brskalnik Mozilla Firefox. Med najuspešnejšimi programi je operacijski sistem Linux ter odprtokodni operacijski sistem Unix-like [61] (glej Sliko 1.10).

V praksi obstaja veliko splošno znanih ovir za prehod na odprtokodno programiranje. Podjetja so prepričana, da odprtokodne licence vsebujejo viruse, nimajo zadostne formalne podpore in izobraževanj, se hitro spreminjajo in ne sovpadajo z dolgoročnimi načrti. Večina teh ovir je povezanih s stopnjo tveganja. Potrebno je poudariti, da tudi lastniški projekti ne vsebujejo natančnih načrtov za prihodnost, vse odprtokodne licence niso enako virusne, veliko OSS-jevih projektov (še posebej operacijski sistemi) pa prinašajo dobiček iz plačane podpore in dokumentacije. Pogosto uporabljena poslovna strategija komercialnih odprtokodnih programskih podjetij je dvojno-licenčna strategija (*Dual-Licence Strategy*), katero uporabljajo tudi MySQL, Alfresco in drugi.

Linux

Operacijski sistem Linux je ustvaril Linus Torvalds na finski univerzi v Helsinkih. Zanimal se je za Minix, majhen UNIX sistem, in se pri tem odločil, da razvije sistem,



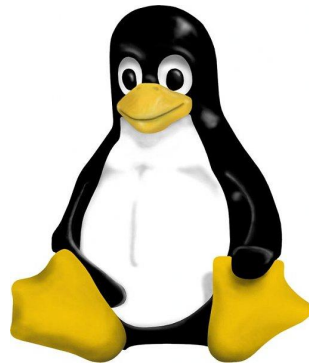
Slika 1.9: Povezave med kategorijami programske opreme

Vir: [108]



Slika 1.10: Odprtokodni produkti

Vir: [3] [36]



Slika 1.11: Odprtokodni produkti

Vir: [49]

ki bo presegel Minix-ove standarde. Z delom je pričel leta 1991 in v letu 1994 izdal prvo verzijo 1.0. Izdana je bila pod GNU licenco, izvorna koda pa je prosto dostopna. V vsakdanjem pogovoru se je ustalilo poimenovanje "Linux" za celoten operacijski sistem plus, kar ni povsem točno, saj se poimenovanje Linux nanaša le na Linuxovo jedro. Samega jedra kot takega ni nemogoče s pridom uporabljati. Uporabno je le kot del celotnega operacijskega sistema, zato se Linux običajno uporablja kot sestavni del operacijskega sistema GNU. Tako govorimo o GNU/Linux, katerega maskota je sedeči pingvin z imenom Tux (Slika 1.11), ustvarjalca Larryja Ewinga [21].

Pridobitev operacijskega sistema Linux je povsem brezplačna. Vsakdo si ga lahko enostavno namesti preko spleta. V svetu so poznane številne distribucije Linuxa, ki so navedene v naslednjih odstavkih.

Ubuntu Linux – Beseda "Ubuntu" je starodavnega afriškega izvora in pomeni "človečnost za druge". Gre za dovršen operacijski sistem, temelječ na distribuciji Debian. Ubuntu skupnost želi, brezplačen dostop do vse programske opreme, uporabnost programskih orodij v lokalnem jeziku in svobodo za prilagajanje programske opreme na kakršen koli ustrezen način.

Pingo Linux - slovenska distribucija Linux. Gre za preprost, poslovenjen in uporabniku prijazen operacijski sistem, ki združuje poslovenjeni namizji KDE in Gnome, poslovenjeno pisarniško zbirko OpenOffice.org, spletni brskalnik in poštni program Mozilla, poslovenjen namestitveni program, multimedijske aplikacije in orodja za upravljanje s sistemom.

Ferix GNU/Linux – slovenska distribucija Linux namenjenega izobraževanju. Združuje poslovenjeni namizji KDE in Gnome, poslovenjeno pisarniško zbirko OpenOffice.org, spletni brskalnik in poštni program Mozilla, poslovenjen namestitveni program, multimedijske aplikacije in orodja za upravljanje s sistemom ter seveda izobraževalne programe za vse starostne skupine. Druge Linux distribucije so še: *Slax*,



Slika 1.12: Ubuntu

Vir: [26]

Konppix, Mandriva Linux in SuSE Linux.

V slovenskem jeziku je izšla prosto dostopna knjiga Linux na namizju, katero uporabniki Linux brezplačno prenesejo iz spletnega naslova: Linux na namizju [30]. Linux na namizju je v pomoč pri preučevanju Linux operacijskega sistema in sami uporabi Ubuntu Linux.

Ubuntu Linux

Kot smo že omenili je Ubuntu Linux ena izmed številnih Linux distribucij. Temelji na znani Linux distribuciji Debian, ki slovi predvsem po varnosti in hitremu sistemu za nameščanje programskih paketov. Uporablja namizno okolje Gnome, na voljo je tudi posebna različica za KDE. Na sistem je kasneje možno namestiti tudi poljubno namizno okolje. Ubuntu Linux je na voljo za 32 in 64-bitne sisteme. Izgled sistema lahko povsem prilagodimo operacijskemu sistemu Windows. Na voljo je tudi v slovenskem jeziku.

V primeru, da ima uporabnik na računalniku že nameščen operacijski sistem Windows (oz. kakšen drug sistem), ga Ubuntu ne "povozi", pač pa omogoči, da na istem računalniku vzporedno uporabljamo več operacijskih sistemov.

1.11 Vrste licenc

1.11.1 Licenca GNU GPL

Kadar govorimo o Linux je potrebno poudariti še t.i. licenco GNU GPL, ki dovoljuje prosto razmnoževanje in urejanje ter izboljševanje programske kode. Uporaba Linux zato pomeni uporabo licenčne in povsem zakonite programske opreme. GNU GPL licenca uporabniku omogoča poganjanje programa za kakršenkoli namen - uporabnik

lahko preučuje, kako program deluje in program prilagaja svojim potrebam, prosto razširja kopije programa, program izboljšuje in daje svoje izboljšave na voljo javnosti.

Običajno so licenčne pogodbe zasnovane tako, da uporabniku preprečujejo njegovo svobodno razdeljevanje in spreminjanje. Za razliko od teh *Splošno dovoljenje GNU* (angl. *GNU General Public License, GPL*) jamči svobodo pri razdeljevanju in spreminjanju prostega programiranja in s tem zagotavlja, da programi ostanejo prosti za vse uporabnike. Dovoljenje GNU uporabniku zagotavlja pravico razširjati kopije prostega programiranja (in zaračunavati za to storitev, če tako želi).

Določitve in pogoji za razmnoževanje, razširjanje in spreminjanje

Licenca je povzeta po spletnem naslovu Lugos.si, prevajalec Roman Maurer [31].

(0) "Licenca se nanaša na vsak program ali drugo delo, ki vsebuje obvestilo lastnika avtorskih pravic... z izjavo, da se lahko distribuira pod pogoji Splošnega dovoljenja GNU... Ta licenca ne pokriva nobenih drugih aktivnosti razen razmnoževanja, razširjanja in sprememb; ostale so izven njenega dometa. Dejanje poganjanja programa ni omejeno in izhod programa je zajet le, če njegova vsebina sestavlja delo, iz katerega je izpeljan program (ne glede na to, da je bil narejen s poganjanjem programa)... " [31].

(1) "Razmnožujete in razširjate lahko dobresedne izvode izvorne kode programa v enaki obliki, kot jo dobite, preko kateregakoli medija, če le na vsakem izvodu razločno in primerno objavite obvestilo o pravicah razširjanja in zanikanje jamstva... Za fizično dejanje prenosa kopije lahko zaračunavate in po vaši presoji lahko ponudite garancijsko zaščito v zameno za plačilo" [31].

(2) "Spreminjati smete vaš izvod ali izvode programa ali katerikoli njegov del, in tako narediti delo, ki temelji na programu, ter razmnoževati in razširjati takšne spremembe ali dela pod pogoji zgornjega razdelka 1, če zadostite tudi vsem naslednjim pogojem: (a) Zagotoviti morate, da spremenjene datoteke nosijo vidna obvestila o tem, da ste jih spremenili in datum vsake spremembe. (b) Zagotoviti morate, da je vsako delo, ki ga razširjate ali izdajate in ki v celoti ali deloma vsebuje program ali katerikoli njegov del ali pa je iz njega izpeljano, licencirano pod pogoji te licence kot celota brez plačila katerikoli tretji osebi (c) Če spremenjeni program ob zagonu navadno bere ukaze interaktivno, morate zagotoviti, da se ob najbolj običajnem zagonu za takšno interaktivno uporabo izpiše ali prikaže najava, ki vključuje primerno sporočilo o pravicah razširjanja in sporočilo, da jamstvo ni zagotovljeno... " [31].

(3) "Program (ali delo, ki temelji na njem, pod razdelkom 2) lahko razmnožujete in razširjate v objektni kodi ali izvedljivi obliki pod pogoji zgornjih razdelkov 1 in 2, če izpolnite tudi kaj od tega: (a) Opremite ga s popolno in ustrezno izvorno kodo v strojno berljivi obliki, ki mora biti razširjana pod pogoji zgornjih razdelkov 1 in 2 na mediju, ki se navadno uporablja za izmenjavo programja; ali, (b) Opremite ga z napisano ponudbo, veljavno vsaj tri leta, da boste katerikoli tretji osebi, za plačilo, ki ne bo presegalo vaših stroškov fizičnega izvajanja izvorne distribucije, dali popoln izvod ustrezne izvorne kode v strojno berljivi obliki, ki bo razširjana pod pogoji zgornjih razdelkov 1 in 2 na mediju, ki se običajno uporablja za izmenjavo programja; ali, (c) Opremite ga z

informacijo, ki ste jo dobili vi, kot ponudbo distribucije ustrezne izvorne kode. (Ta alternativa je dovoljena le za nekomercialne distribucije in le, če ste dobili program v obliki izvorne kode ali izvedljivi obliki s takšno ponudbo, glede na podrazdelek b, zgoraj.)" [31].

(4) "Ne smete razmnoževati, spreminjati, podlicencirati ali razširjati programa drugače, kot to izrecno določa pričujoča licenca. Vsak poskus siceršnjega kopiranja, spreminjanja, podlicenciranja ali razširjanja programa je nič in bo samodejno prekinil vaše pravice pod to licenco. Vendar pa se osebam, ki so svoj izvod ali pravice dobile od vas pod to licenco, licenca ne prekine, dokler se ji popolnoma podrejšo" [31].

(5) "Ni vam treba sprejeti te licence, saj je niste podpisali. Vendar vam razen nje nič ne dovoljuje spreminjanja ali razširjanja programa ali iz njega izpeljanih del. Če ne sprejmete te licence, ta dejanja prepoveduje zakon. Torej, s spremembo ali razširjanjem programa (ali katerokoli dela, ki temelji na programu), pokažete svoje strinjanje s to licenco in z vsemi njenimi določitvami in pogoji za razmnoževanje, razširjanje ali spreminjanje programa ali del, ki temeljijo na njem" [31].

(6) "Vsakič, ko razširjate program (ali katerokoli delo, ki temelji na programu), prejemnik samodejno prejme licenco od izvirnega izdajatelja licence (angl. original licensor) za razmnoževanje, razširjanje ali spreminjanje programa glede na ta določila in pogoje. Ne smete vsiljevati nobenih nadaljnjih omejitev izvajanja prejemnikovih pravic, podeljenih tukaj. Niste odgovorni za vsiljevanje strinjanja tretjih oseb s to licenco" [31].

(7) "Če so vam, kot posledica presoje sodišča ali suma kršitve patenta ali zaradi katerekoli drugega razloga (ne omejenega zgolj na patentna vprašanja), vsiljeni pogoji (bodisi z odlokom sodišča, sporazumom ali drugače), ki nasprotujejo pogojem te licence, vas ne odvezujejo pogojev te licence. . . Ta razdelek namerava temeljito pojasniti, kaj so predvidene posledice nadaljevanja licence" [31].

(8) "Če sta razširjanje in/ali uporaba programa omejena v določenih državah, bodisi zaradi patentov ali vmesnikov s posebno pravico razširjanja (angl. copyrighted interfaces), lahko izvorni lastnik ali lastnica pravic razširjanja, ki postavlja program pod to licenco, doda eksplicitno zemljepisno omejitev razširjanja, ki izključuje te države, tako da je razširjanje dovoljeno le v in med državami, ki niso na tak način izključene. V takem primeru ta licenca vključuje omejitve, kot da so napisane v telesu te licence" [31].

(9) "Ustanova Free Software Foundation lahko od časa do časa izdaja preurejene in/ali nove različice Splošne javne licence (angl. *General Public License*). Nove različice bodo pisane v duhu trenutne različice, vendar se lahko razlikujejo v podrobnostih, ki bodo obdelovale nove težave ali poglede. Vsaki različici je prirejena razločevalna številka različice. Če program določa številko različice te licence, ki se nanaša na njo in "na katerekoli poznejše različice", imate izbiro upoštevanja pogojev in določil bodisi te različice ali katerekoli poznejše različice, ki jo je izdala ustanova Free Software Foundation. Če program ne določa številke različice te licence, lahko izberete katerokoli različico, ki jo je kdajkoli izdala ustanova Free Software Foundation" [31].

(10) "Če želite vključiti dele programa v druge proste programe, katerih pogoji razširjanja so drugačni, pišite avtorju in ga prosite za dovoljenje. Za programje, katerega pravice razširjanja ima Free Software Foundation, pišite na Free Software Foundation;

včasih naredimo izjemo pri tem. Našo odločitev bosta vodila dva cilja: ohranitev prostega statusa vseh izvedenih del iz našega prostega programja in spodbujanje razdeljevanja in ponovne uporabe programja na splošno" [31].

Brez jamstva

(11) "Ker je program licenciran kot brezplačen, ni nobenega jamstva za program do meje, ki jo določa pristojni zakon. Razen, če ni drugače napisano, imetniki pravic razširjanja in/ali druge osebe ponujajo program "tak kot je", brez zagotovila kakršnekoli vrste, neposrednega ali posrednega, kar vključuje, a ni omejeno na posredna jamstva cenovne vrednosti in primernosti za določeno uporabo. Celotno tveganje glede kakovosti in delovanja programa prevzamete sami. Če se program izkaže za okvarjenega, sami nosite stroške vseh potrebnih storitev, popravil ali popravkov" [31].

(12) "V nobenem primeru, razen če tako pravi veljavni zakon ali je pisno dogovorjeno, ne bo lastnik pravic razširjanja ali katerakoli druga oseba, ki lahko spremeni in/ali ponovno razširja program, kot je to dovoljeno zgoraj, prevzel odgovornosti zaradi škode, najsi gre za splošno, posebno, nenamerno škodo ali škodo, izhajajočo iz uporabe ali nezmožnosti uporabe programa (vključno z, a ne omejeno na, izgubo podatkov ali nenatančno obdelavo podatkov ali izgubo, povzročeno vam ali tretjim osebam ali nezmožnost programa, da bi deloval s kakim drugim programom), četudi je bil tak lastnik ali druga oseba obveščen o možnosti nastanka takšne škode" [31].

1.11.2 Druge odprtokodne in prosto dostopne licence

Uporabniki odprtokodnih programov lahko izbirajo še med ostalimi ponudniki programske opreme. Licence z odprto kodo se v veliki meri razlikujejo. Med najbolj uporabljenimi so GPL, katero smo že opisali, BSD, LGPL in MPL.

Berkeley Software Distribution (BSD) licenca je preprosta, saj dovoljuje uporabo programske opreme, distribucijo izdelka in izvirne kode. Dovoljuje spreminjanje in vključevanje v drugo programsko opremo brez omejitev. Zahteva, ki jo je potrebno upoštevati je navedba avtorjev v izvorni kodi in dokumentaciji programa. Imen avtorjev ni dovoljeno uporabljati za promocijo izdelka brez predhodnega pisnega dovoljenja [61].

Licenca *Library General Public License (LGPL)* je variacija GPL licence in je namenjena programskim knjižnicam (na primer DLL), torej programom s funkcijami, ki jih je mogoče izkoristiti v drugih programih. Licenca *Mozilla Public License (MPL)* dovoljuje brezplačno uporabo in distribucijo programske opreme. Vsakdo, ki razpečuje spremembe ali dodatke k izdelku, mora dovoliti enake pravice za njegov kos programske opreme in kombinacijo izvirnega dela in njegove rešitve. Pogoji za distribucijo je tudi izvorna koda, ki je lahko vključena v distribucijo ali pa je dostopna na kakšni spletni strani. MPL je dovoljeno vključevati tudi v licenčno programsko kodo lastnih izdelkov, a mora biti za ta del na voljo izvorna koda [61].

BSD licenca temelji na načelih proste programske opreme (*Free Software*) in je prirejena za istega uporabnika. Edini pogoj, ki ga je potrebno vključiti je informacija o

avtorstvu programske opreme in licenciranje vsebin.

Common Public License (CPL) licenca je prav tako prosto programska oprema. Leta 1988 je bila odobrena s strani *Open Source* in je podobna opisanemu dovoljenju "GNU *General Public License*".

1.11.3 Študentska licenca

Mnogo katero programsko orodje ni prosto dostopno na spletu, vendar ga kljub temu lahko pridobimo, kar omogoča študentska verzija (*Student Version*) licence. Na podlagi pridobljenih t.i. študentskih verzij s strani izbrane fakultete je omogočena enoletna, večletna ali dolgoročna uporaba posameznega programskega orodja. Običajno se pojavi razlika med študentskimi in pravimi verzijami, ki jih je potrebno plačati, v okrnjenosti delovanja programskega orodja. Kot primer navedimo programsko orodje GPSS, namenjeno simulacijam. Študentje imajo možnost pridobitve študentske verzije na spletu ali na fakulteti, pri čemer je število blokov študentske verzije omejeno na 180, plačljiva verzija pa v okviru možnosti vsebuje 2000 blokov.

Študentske verzije so namenjene zgolj k spoznavanju dela z določenim programskim orodjem in k reševanju enostavnih problemov z manjšim številom podatkov. To seveda variira pri vsaki posamezni opremi in lastniškem podjetju posebej. Nekatera podjetja omogočajo pridobitev študentske licence za določene programe zgolj s posredovanjem originalnega potrdila o šolanju, z vsemi vsebovanimi moduli. V določenih primerih je za prijavo potrebno posredovati podatke izbrane fakultete oz. izbrana fakulteta pridobi dovoljenje in nato študentje uporabljajo program s časovno omejenim obdobjem.

1.11.4 Licenca za prosto verzijo

V praksi se pogosto znajdemo v situaciji, ko potrebujemo določeno programsko orodje le za kratek čas. Nakup le tega predstavlja prevelik strošek, glede na dejstvo, da ga potrebujemo npr. zgolj za določen projekt. V takšnem primeru so na voljo t.i. licence za poskusno verzijo. Praksa je, da lahko vsi uporabniki prosto uporabljajo program v poskusnem obdobju trideset (30) dni, s pričetkom štetja od prve aktivacije licence za poskusno verzijo na uporabnikovem računalniku.

Ko se odločamo o bodočem nakupu in želimo preučiti več različnih programskih orodij, lahko uporabimo t.i. *shareware licence*. Dostopne so dovolj časa, da preučimo posamezen program in ga glede na zahtevane značilnosti primerjamo z ostalimi. Gre za licenco s simbolnim plačilom, ki po namestitvi omogoča uporabo vseh funkcij programa za določeno obdobje. V primeru nadalje rabe programskega orodja je po preteku časa potrebno kupiti celotno licenco (npr. protivirusni program NOD32 lahko testiramo 30 dni, nato moramo kupiti polno licenco).

Programska orodja				
Programsko orodje	Vrsta	Licenca GNU GPL	Uporaba	Omejitve
Planner	Odprtokoden	Da	Sistem	Ne
Dia	Odprtokoden	Da	Sistem	Ne
Zint	Odprtokoden	Da	Sistem	Ne
ASDN	Odprtokoden	Da	Interne	Ne
Google Zemlja	Prosto dostopen	Ne	Internet	Ne
Quantum GIS	Odprtokoden	Da	Sistem	Ne
Qcad	Odprtokoden	Da	Sistem	Ne
Simple Warehouse Mapper	Prosto dostopen	Ne	Sistem	Ne
Petersen	Prosto dostopen	Ne	Java	Da
Lindo	Prosto dostopen	Ne	Sistem	Da
Dexi	Prosto dostopen	Da	Sistem	Ne
GnuCash	Odprtokoden	Da	Sistem	Ne
GPSS World	Prosto dostopen	Da	Sistem	Da
Scilab	Odprtokoden	Da	Sistem	Ne
OpenOffice.org Preglednica	Odprtokoden	Da	Sistem	Ne
PSPP	Odprtokoden	Da	Sistem	Ne

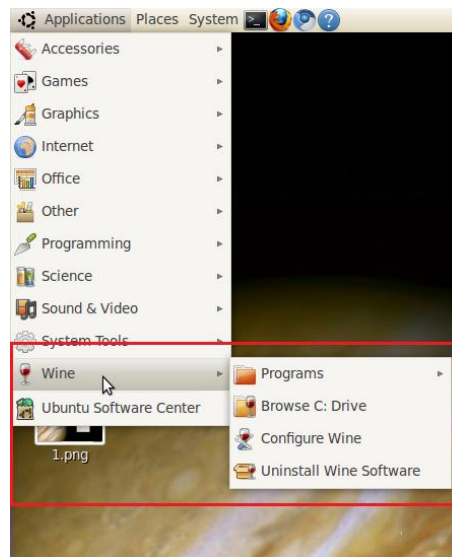
Tabela 1.3: Programska orodja

1.12 Kako uporabljati programska orodja?

V nadaljevanju predstavljamo uporabo programskih orodij napisanih za Windows okolje v operacijskem sistemu Ubuntu (glej Tabelo 1.3).

V izbor programskih orodij uvrstimo tudi nekaj programskih orodij, ki so napisana izključno za uporabnike operacijskega sistema Windows. V kolikor je program napisan zgolj za eno okolje in brez odprtokodne licence, potem izgubi pomen odprtokodnega programa v točki delovanja na obeh platformah (Windows ali Linux) in dostopa do izvirne kode. Ko so programska orodja brezplačna in javno dostopna ustrezajo definiciji proste programske opreme, ki se od odprtokodne razlikuje le v dostopu do izvirne kode.

Preizkusimo delovanje vseh izbranih programskih orodij v različnih okoljih, Windows in Ubuntu. Preizkus potrdi, da nekatera programska orodja napisana za okolje Windows ne delujejo v Ubuntu. Za takšna programska orodja poiščemo rešitev - program Wine. Z uporabo programa Wine vsa izbrana programska orodja napisana za okolje Windows delujejo tudi v Ubuntu.



Slika 1.13: Program Wine

Program Wine

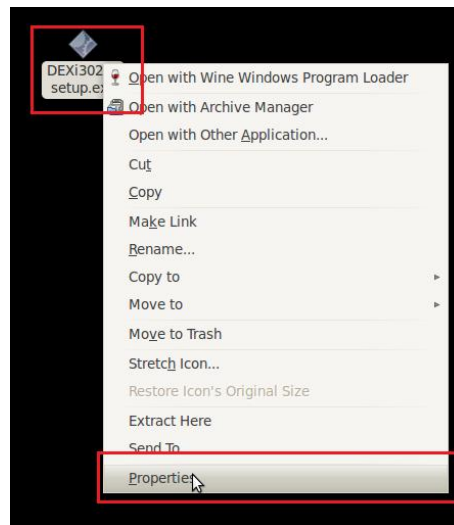
Wine je program, ki ponuja uporabnikom Linux operacijskih sistemov kompatibilnost delovanja programskega orodja napisanega za okolje Windows. Program Wine lahko uporabniki operacijskega sistema Ubuntu enostavno poiščejo, prenesejo in namestijo s pomočjo upravljalca paketov Synaptic ali pa ga poiščejo na spletni strani Wine HQ [51]. Program je odprtokoden in izdan pod GNU GPL licenco [67]. Wine ima tudi svojo knjižnico Winelib, katero razvijalci dopolnjujejo s seznamom delujočih programskih orodij napisanih za okolje Microsoft Windows v Linux okoljih [65] (glej Sliko 1.13).

Uporaba programa Wine je za uporabnike Ubuntu enostavna, vseeno pa opišimo njegovo uporabo pri namestitvi programskega orodja napisanega za okolje Microsoft Windows. V kolikor želimo namestiti takšno programsko orodje, ga s pomočjo spleta ali upravljalca paketov Synaptic poiščemo in prenesemo. Namestitev izvedemo z desnim klikom na *.exe* datoteko, nakar izberemo *Properties* oz. *Nastavitve* (glej Sliko 1.14).

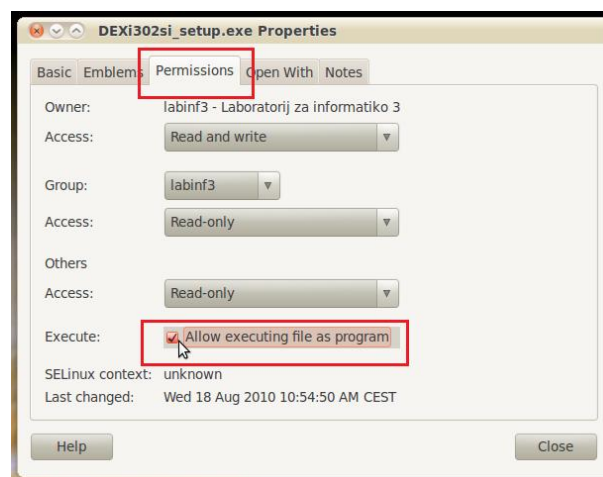
Kliknemo na *Permissions* oz. *Dovoljenja*, kjer izberemo možnost *Execute*, s čimer dovoljujemo namestitev dokumenta kot program (glej Sliko 1.15).

V razdelku *Open With* oz. *Odpri* izberemo *Wine Windows Program Loader* (glej Sliko 1.16).

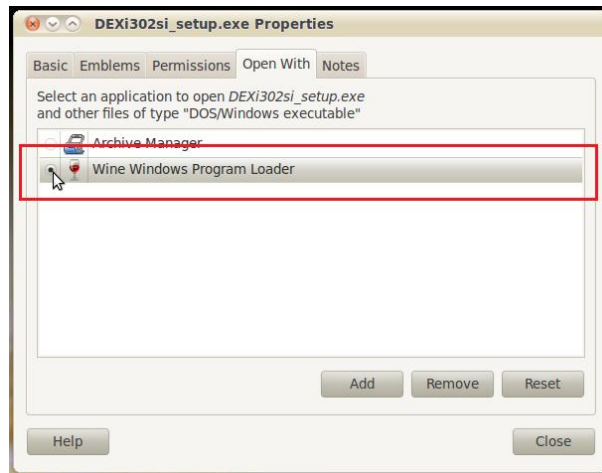
Zapremo in ponovno s desnim klikom kliknemo na *.exe* datoteko. Izberemo možnost *Open with Wine Windows Program Loader*, s čimer se prične namestitev programskega orodja, kakršno smo vajeni v Windows okolju (glej Sliko 1.17).



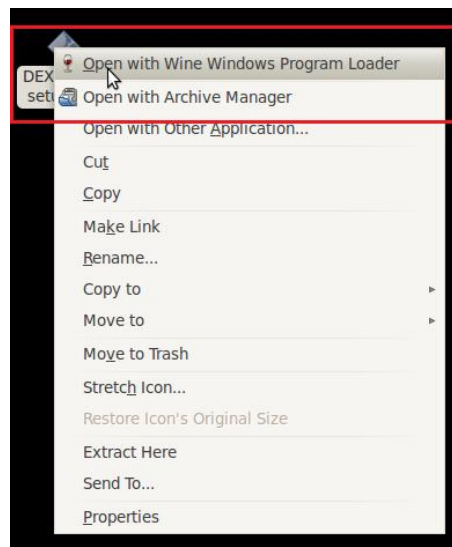
Slika 1.14: Namestitev programskega orodja s pomočjo programa Wine



Slika 1.15: Dovoljenje za odpiranje datoteke kot program



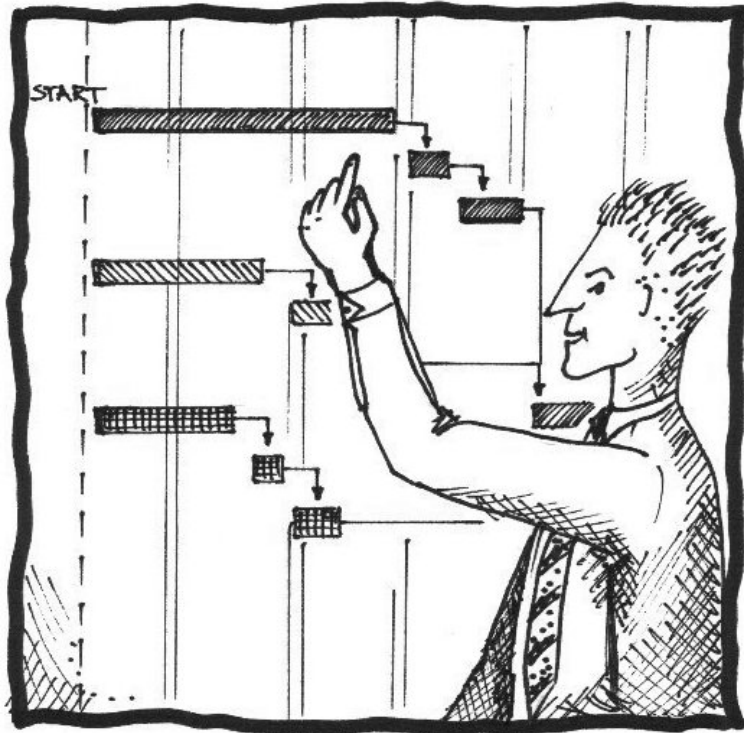
Slika 1.16: Nastavitev za odpiranje z Wine Windows Program Loader



Slika 1.17: Odpiranje z Wine Windows Program Loader

Poglavje 2

PLANNER - načrtovanje aktivnosti



Projektno vodenje
Ganttov diagram
Primer: načrtovanje aktivnosti in procesov

2.1 Teoretično ozadje

2.1.1 Tehnologija črtne kode

Projektno vodenje pomeni proces časovnega načrtovanja, nadzorovanja in poročanja. Njegov pomen narašča z obsegom in kompleksnostjo projektov. Projektno vodenje je predvsem pomembno v fazi načrtovanja, ki je običajno najdaljša in najpomembnejša pri razvoju projektov. Uspešnost projekta določajo trije cilji – čas, stroški in funkcionalnost. Vsak projekt zahteva projektne vodje, ki je zadolžen in odgovoren za uspešnost celotnega projekta in doseganje ciljev. Vsi obsežni projekti vsebujejo predmet projektnega managementa. Projektno vodenje je danes jedro poslovanja marsikatero organizacije [87].

Načrtovanje projekta je ena izmed temeljnih nalog projektne vodje. Vključuje oblikovanje posameznih nalog ter oceno potrebnega obsega dela in predvidenega časa do zaključka dela. Projektne načrt je eden izmed temeljnih orodij za obvladovanje stroškov in časovnega projekta. Načrtovanje se običajno začne s sestavljanjem seznama nalog. Projekt razčlenimo na posamezne naloge, ki so dovolj enostavne in razumljive za izvedbo in merjenje uspešnosti projekta. Vsaka naloga oz. aktivnost ima svoj začetek, konec, opis vsebine in različne vire (čas, denar, človeško delo). Ob nalogah so pomembni element projektnega načrta tudi dogodki, ki pomenijo točke, kjer so doseženi cilj ali spremljajoče aktivnosti.

Z orodji podprto projektno načrtovanje in vodenje je pomembno za zagotovitev uspešnosti projektov. Dandanes uspešna podjetja obvladajo izpopolnjene in razvite metode in tehnike, pri tem pa uporabljajo različna orodja in rešitve. Vodilno in prevladujočo vlogo med komercialnimi izdelki za podporo projektne delu ima programski paket Microsoft Project. Na trgu obstaja veliko primerljivih alternativ. Večina projektov v podjetjih je manjšega obsega in jih lahko enako učinkovito upravljamo z uporabo brezplačnih orodij oz. orodij z odprtokodno licenco, ki jih je možno enostavno najti na internetnih straneh različnih ponudnikov. Eno izmed takšnih orodij je tudi Planner.

2.1.2 Ganttov diagram

Ganttov diagram je grafikon, s katerim vizualno prikažemo dogajanje oziroma trajanje določenih nalog znotraj projekta [8]. Najpogosteje se uporablja kot orodje managementa. Izdelamo ga v fazi načrtovanja in uporabimo v fazi izvedbe projekta. Je eden izmed najenostavnejših in najpogosteje uporabljenih tehnik za prikazovanje aktivnosti projekta. Začetki uporabe Gantt diagramov segajo v zgodnje obdobje 20. stoletja, ko je Frederic W. Taylor uporabil grafični prikaz za planiranje proizvodnje, kar je kasneje Henry L. Gantt razvil v današnji Gantt diagram [91]. Gantt diagram je bil kot orodje uporabljen v začetni fazi evolucije sistema planiranja proizvodnje. Evolucija sistema je temeljila na razvoju uporabnega in trajnostnega sistema planiranja [92]. Prednosti:

- napoved končnega časa projekta;
- razvrščanje dejavnosti;
- pohitritev bodočih nalog;
- pregled nad sredstvi, ki jih potrebujemo za projekt;
- nadzor nad opravljenim delom.

Slabosti:

- ne prikaže najkrajšega in najdaljšega predvidenega časa izvedbe;
- medsebojna odvisnost med dejavnosti ni prikazana;
- ne prikaže posledic zgodnjega ali poznega začetka posamezne dejavnosti.

Jakomin, Zelenika in Medeot [60] opredeljujejo logistiko kot skupek med seboj povezanih aktivnosti in procesov, ki služijo za premikanje surovin, polproizvodov, drugega materiala in gotovih proizvodov od dobaviteljev do podjetja, za premikanje znotraj podjetja in od podjetja do odjemalcev oz. kupcev ter vse z njimi povezane aktivnosti. Vse te aktivnosti in procese je potrebno organizirati in voditi, kar je predmet projektnega vodenja. Planiranje teh aktivnosti in procesov je temelj za nadaljnje delo. Rezultat izvedbe planiranja se odraža v uspešnosti končanega projekta.

Ganttov diagram izdelamo tako, da naprej v tabelo zapišemo vse naloge oz. aktivnosti, njihov začetek, trajanje in konec (Tabela 2.1) ter določimo medsebojne odvisnosti. Prva naloga projektnega vodje je identifikacija konkretnih aktivnosti, ki jih je potrebno izvajati za doseg pričakovanih rezultatov in ciljev projekta. Za vsako aktivnost je potrebna ocenitev trajanja - čas, ki je potreben za izvajanje aktivnosti. Vse aktivnosti je potrebno povezati tako, da vsaki aktivnosti določimo predhodne in naslednje aktivnosti. Pomembni aktivnosti sta še načrtovanje virov in izračun cene posamezne aktivnosti, vendar jih za potrebe našega projekta ne bomo upoštevali.

Problem

Vso potrebno blago za sestavo vozila prispe v skladišče z različnimi transportnimi sredstvi. Predpostavimo, da vsa transportna sredstva odpeljejo ob istem času. Čas transporta z vlakom do skladišča traja 6 ur, čas tovornjaka 3 ure in čas kombija 3 ure. Ob prispetju do skladišča, sledi čakanje na prosti terminal, urejanje dokumentacije, raztovarjanje ter skladiščenje blaga. Za vsako vrsto blaga so zahtevani časi določeni v 2.1. Sestava v enoten izdelek sledi šele, ko je vso blago v skladišču - ko so zaključene aktivnosti 4, 8 in 12. Sledi še skladiščenje končnega izdelka in odprema v proizvodnjo. Ko imamo oblikovano tabelo vseh aktivnosti, izdelamo Ganttov diagram s pomočjo orodja Planner.

Št.	Aktivnost	Začetek	Trajanje	Konec	Predhodna aktivnost
1	Transport pnevmatike z vlakom	0	6	6	/
2	Čakanje in urejanje dokumentacije	6	1	7	1
3	Raztovarjanje	7	2	9	2
4	Skladiščenje pnevmatik	9	3	12	3
5	Transport platišč s tovornjakom	0	3	3	0
6	Čakanje in urejanje dokumentacije	3	1	4	5
7	Raztovarjanje	4	2	6	6
8	Skladiščenje platišč	6	3	9	7
9	Transport vijakov s kombijem	0	3	3	0
10	Čakanje in urejanje dokumentacije	3	1	4	9
11	Raztovarjanje	4	1	5	10
12	Skladiščenje vijako	v 5	3	8	11
13	Komisioniranje	12	1	13	4, 8, 12
14	Natovarjanje	13	1	14	13
15	Odprema v proizvodnjo	14	1	15	14

Tabela 2.1: Tabela vseh aktivnosti in njihovih trajanj

2.2 O programskem orodju

Planner je orodje za planiranje, organiziranje in sledenje različnim projektom. Ponuja večino potrebnih orodij, parametrov in možnosti upravljanja projektov, ki jih projektni vodja potrebuje pri vodenju projektov [41](glej Sliko 2.1). Omogoča: definiranje nalog, njihovih podnalog in virov; vpogled v odvisnosti med posameznimi nalogami; prikaz kritične poti; koledarje s prikazom ali brez prikaza delovnih dni; Ganttov diagram; prikaz porabe virov; HTML izvajanje projektni planov.

Še nekaj prednosti: odprtokodna licenca; enostavna in intuitivna uporaba; širok nabor vgrajenih orodij in poročil.

Prenos in namestitve

Planner prenesemo s spletnega naslova Planner [41] v razdelku *Downloads* (glej Sliko 2.2), kjer lahko izbiramo med različnimi verzijami programskega orodja, odvisno kateri operacijski sistem uporabljamo. Uporabniki operacijskega sistema Ubuntu programsko orodje enostavno poiščejo, prenesejo in namestijo na računalnik z orodjem Synaptic.

Programsko okno

Vsa potrebna orodja najdemo v menijski in orodni vrstici. V menijski vrstici so razdelki *File, Edit, View, Actions, Project* in *Help*. Vsa orodja, najdena v teh razdelkih so tudi v orodni vrstici. V kolikor želimo odpreti ali shraniti dokument, lahko to storimo s klikom na *Save* v orodni vrstici ali s klikom na *File* in nato *Save*. Najpomembnejša orodja pri izdelavi Gantt diagrama oz. kritične poti so opravila *Tasks*, katere lahko v orodni vrstici vstavimo, odstranimo, povežemo ali premikamo.

Na levi strani programskega okna imamo na izbiro različne poglede, odvisno kaj potrebujemo in s čim se ukvarjamo. Za prikaz povezav med aktivnosti in kasnejšega prikaza kritične poti izberemo pogled *Gantt*, za urejanje opravil oz. aktivnosti *Tasks*, vire *Resources* itd. (glej Sliko 2.3).

Planner news

[News](#)
[Screenshots](#)
[Downloads](#)
[Development](#)
[Community](#)
[About](#)

Bounties

A bounty is being offered for an implementation of resource allocation/leveling. [Read the offer here](#) and send mail to the [planner-dev list](#) for more information.

Current Development

Here's some features we're looking at implementing in v0.15:

- Resource leveling [132917](#)

New in the v0.14.4 Release - 15 April 2009

- Rewrote printing to use the GTK printing API, fixing most problems with printing on Windows (thanks to Francisco Moraes)
- Added display of minutes to duration format
- Added creation of Windows installer to Makefiles
- Dropped support for libgda < 3.0

Slika 2.1: Spletna stran

Planner downloads

[News](#)
[Screenshots](#)
[Downloads](#)
[Development](#)
[Community](#)
[About](#)

Get Planner (binary)

Binary releases for Linux are provided by the various Linux distributions themselves.

Ubuntu users can just [click here to install the Planner package](#)

Planner 0.14.4 for Microsoft Windows:

planner-0.14.4.news	Summary of changes per release
planner-0.14.4.exe	Planner 0.14.4 installer for Windows
planner-0.14.4.md5sum	MD5 sum

** the -r1 suffix does not signify a new release of Planner. Some bugs in the installer had to be fixed to properly install Planner 0.14.4.*

Earlier releases are available from the [Planner on Windows page](#).

If you feel upto it you can also [build a version for Windows yourself](#).

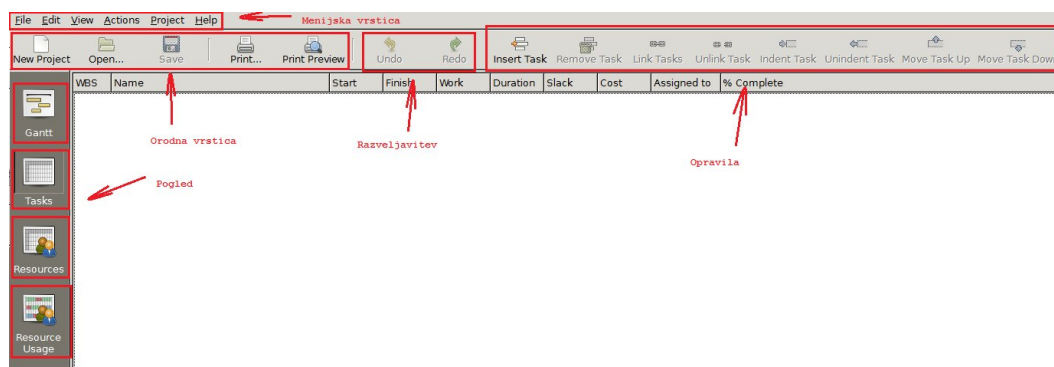
Source code

The following files are available for Planner 0.14.4:

planner-0.14.4.news	Summary of changes per release
planner-0.14.4.changes	Detailed changelog
planner-0.14.4.tar.gz	gzipped tarball with source code of Planner 0.14.4
planner-0.14.4.tar.bz2	bzipped tarball with source code of Planner 0.14.4
planner-0.14.4.md5sum	MD5 sums

Earlier releases are also available from the [Gnome FTP server](#).

Slika 2.2: Prenos programskega orodja



Slika 2.3: Programsko okno

2.3 Uporaba

Z zagonom programskega orodja se odpre nov projekt, prav tako pa ga lahko naknadno odpremo s klikom na gumb *New Project* v orodni vrstici (glej Sliko 2.4).

Že izdelan projekt lahko shranimo v poljubno mapo na računalniku. To storimo tako, da v menijski vrstici kliknemo *File* in izberemo *Save As*. Odpre se novo okno, kjer najprej poimenujemo projekt ter na levi strani izberemo mapo v katero ga želimo shraniti. S klikom na *Save* se projekt shrani na želeno mesto v računalniku (glej Sliko 2.5).

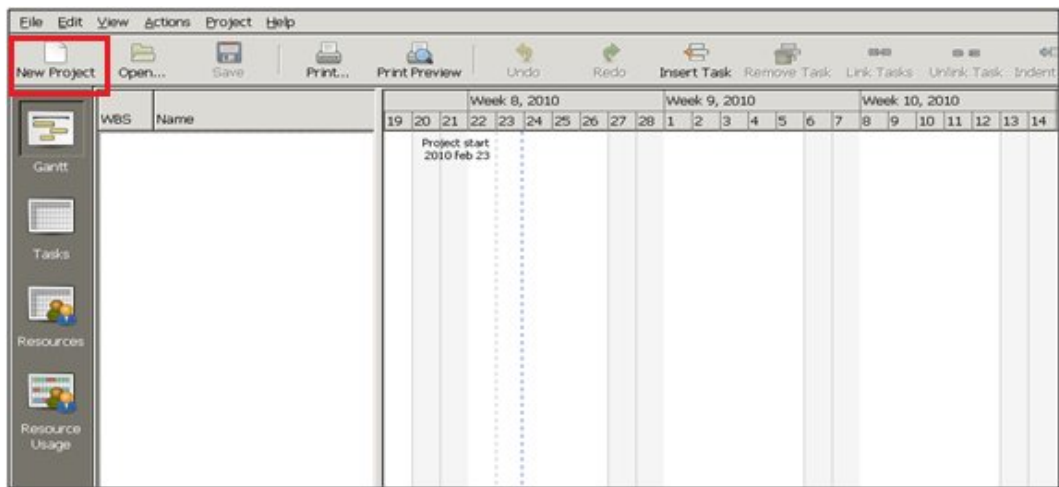
Projekt, ki je že shranjen v sistemu, odpremo s klikom na gumb *Open*. Odpre se novo okno, kjer poiščemo shranjen projekt. S klikom na *Open* potrdimo svoj izbor (glej Sliko 2.6).

Koledarske nastavitve

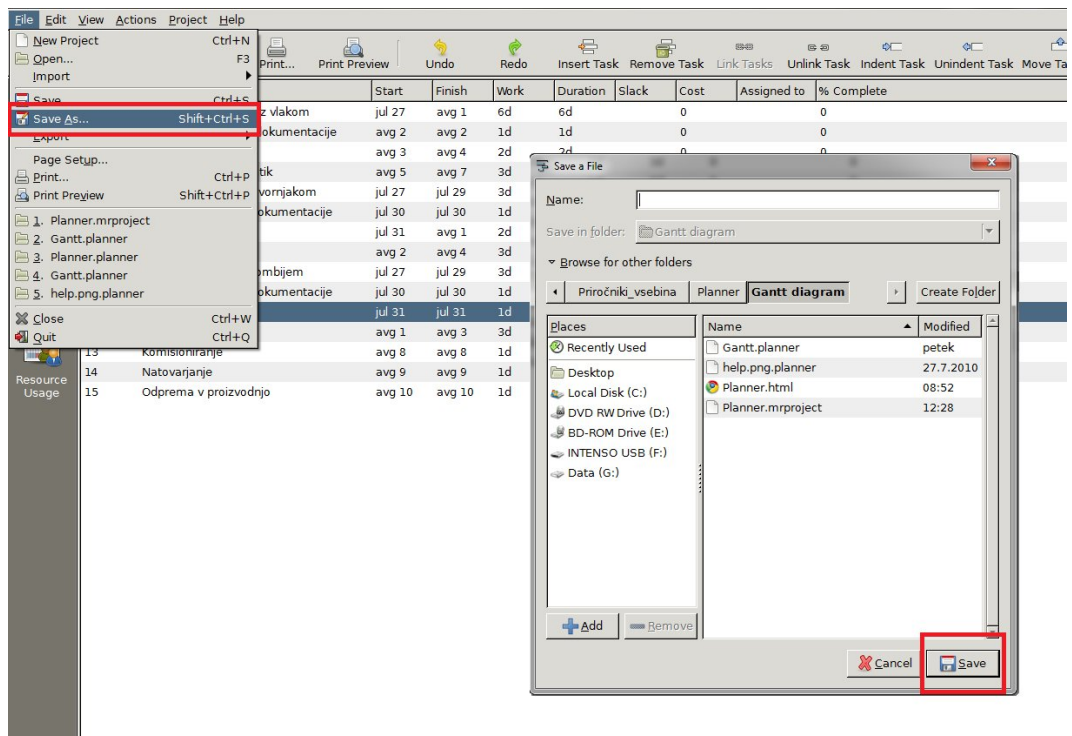
Najprej določimo koledarske nastavitve. V glavni menijski vrstici kliknemo na *Projects > Manage Calendars*. Odpre se novo okno (glej Sliko 2.7), kjer nastavimo zeleni začetni dan projekta, delovni čas (*From, To*) ter tip delovnega dne (*Working, Nonworking*) za vsak dan posebej. Svoje nastavitve potrdimo s klikom na gumb *Apply*.

Opazimo določeno omejitev. Nastavitve upoštevamo le takrat kadar naš projekt oz. naše aktivnosti trajajo dlje časa (dnevi, meseci, leta). Problem se pojavi, kadar želimo diagram aktivnosti izdelati za aktivnosti, ki ne trajajo več kot en dan. Te možnosti program Planner ne dopušča.

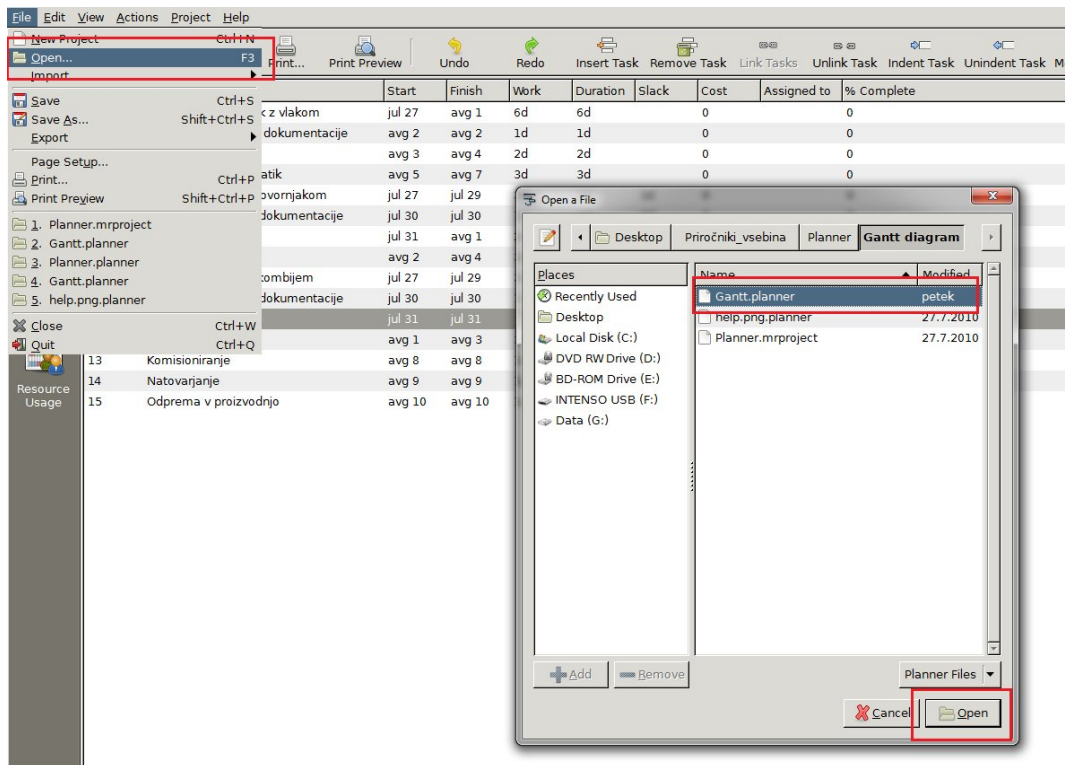
Omejitev se pojavi tudi pri izdelavi Ganttovega diagrama za izbrani primer. Aktivnosti ne trajajo več dni, ampak so del enega dneva, zato predpostavimo, da je ena ura enaka enemu koledarskemu dnevju. Aktivnost ki traja 3 ure, program prikaže kot trajajočo aktivnost 3 dni.



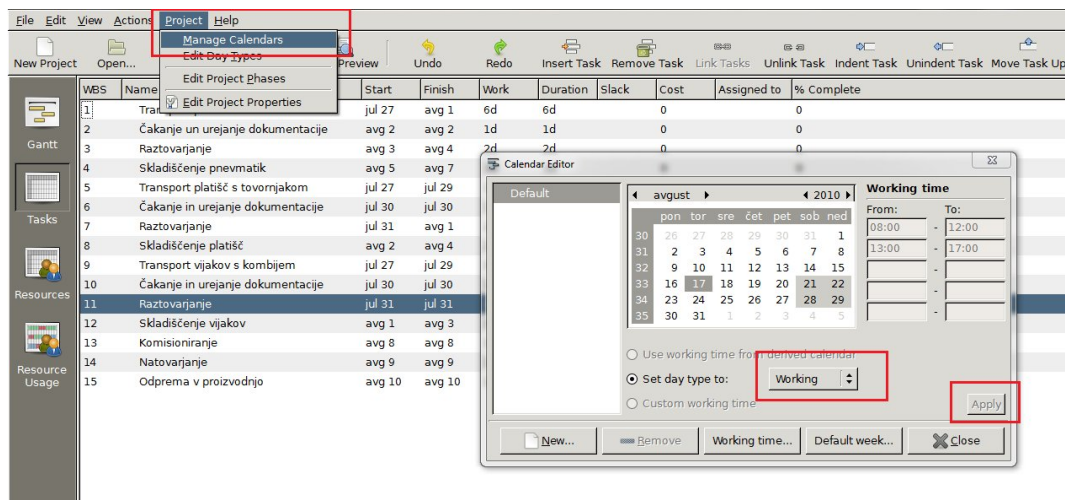
Slika 2.4: Zagon novega projekta



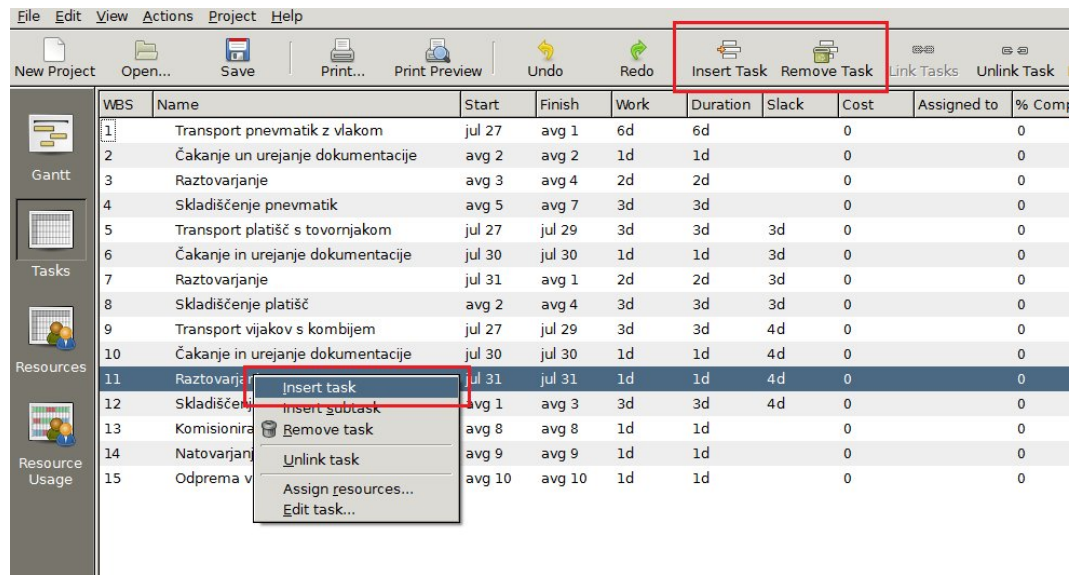
Slika 2.5: Shranjevanje novega projekta



Slika 2.6: Odpiranje shranjenega projekta



Slika 2.7: Koledarske nastavitve



Slika 2.8: Zapis nove aktivnosti

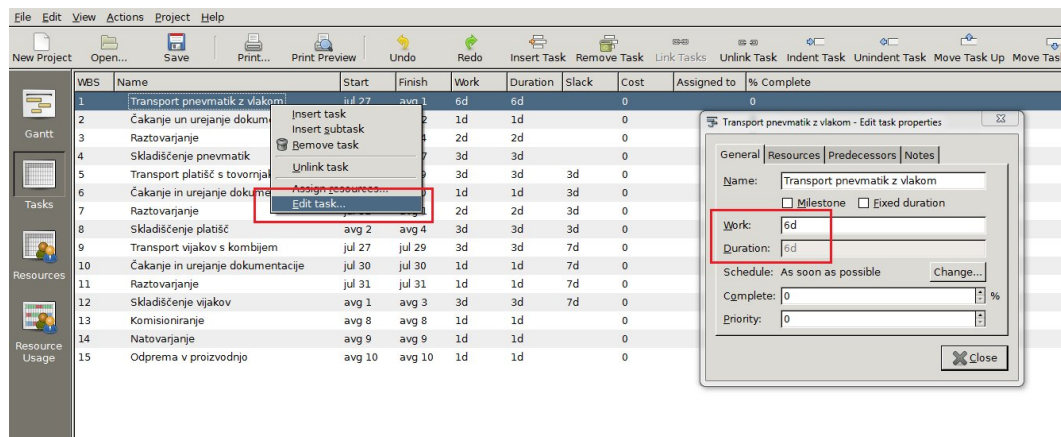
Zapis aktivnosti/podaktivnosti in njihovega trajanja

Opravila oz. aktivnosti projekta zapišemo v okno na levi strani, kjer vsako posamezno aktivnost dodamo tako, da s klikom na desni miškin kazalec izberemo možnost *Insert Task* (glej Sliko 2.8). V stolpec *Name* zapišemo naziv aktivnosti, v stolpec *Work* pa trajanje. Aktivnosti lahko izbrišemo s klikom na *Remove task*.

Določene aktivnosti imajo tudi svoje podaktivnosti. V okno aktivnosti lahko zapišemo tudi te. To storimo tako, da označimo nalogo in z desnim klikom nanjo izberemo *Insert subtask*.

Eden izmed pomembnejših parametrov, ki jih lahko nastavimo, je trajanje posamezne aktivnosti. Čas trajanja posamezne aktivnosti zapišemo že med samim zapisovanjem aktivnosti, lahko pa ga določimo tako, da označimo želeno aktivnost in z desnim klikom nanjo izberemo *Edit task*. Odpre se novo okno (glej Sliko 2.9), kjer pod *Work* spreminjamo čas izvajanja naloge, pod *Duration* pa čas celotne naloge, kamor spadajo morebitne zakasnitve ali pa rezervni čas, ki ga zagotovimo za izvedbo dane naloge. Drug način je klik na aktivnost pod tabelo *Work*, kjer je aktivnost opredeljena s časom trajanja. Tabela *Slack* predstavlja ostanek časa aktivnosti in ne leži na kritični poti. Zakasnitve lahko določamo tudi v jezičku *Predecessors*, pod *Type* spremenimo vrsto povezave med nalogami, pod *Lag* pa vnesemo zakasnitev.

Sam postopek torej ponuja več možnosti za določitev trajanja projekta, saj vključuje tudi možnost določitev rezervnih časov ali morebitnih zakasnitev.



Slika 2.9: Določitev trajanja aktivnosti

Povezava aktivnosti in kritična pot

Za povezavo nalog med seboj najprej označimo naloge, katere želimo, da so med seboj povezane, nato pa s klikom na gumb *Link Tasks*, ki se nahaja v glavni menijski vrstici, spremenimo vrsto povezave med nalogami (glej Sliko 2.10).

Lahko pa izberemo možnost desnega klika na aktivnost, kjer izberemo *Edit task*. V *Edit task* izberemo *Predecessors*, kjer izberemo aktivnost s katero ju bomo povezali (glej Sliko 2.11).

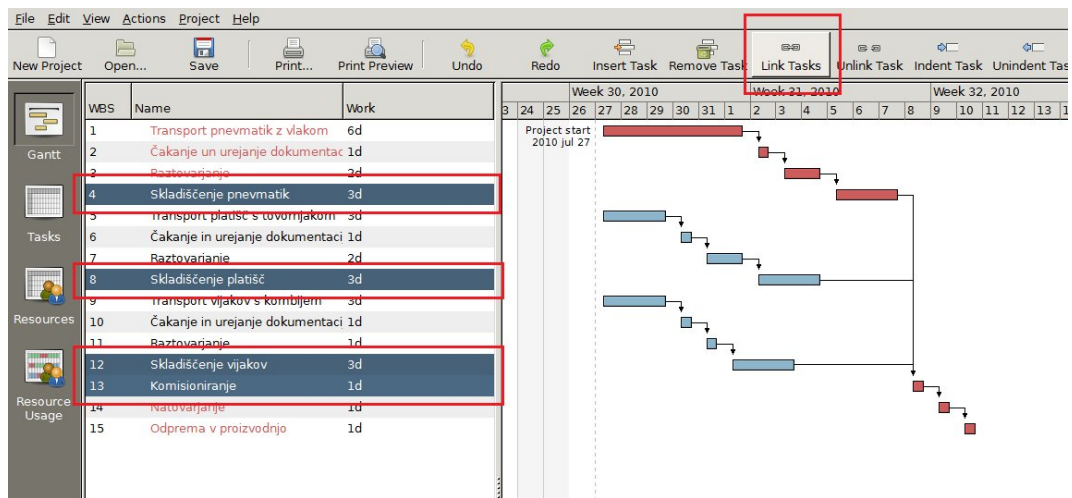
Pri tem lahko določimo tudi obliko povezave (glej Sliko 2.12).

Oblike povezave

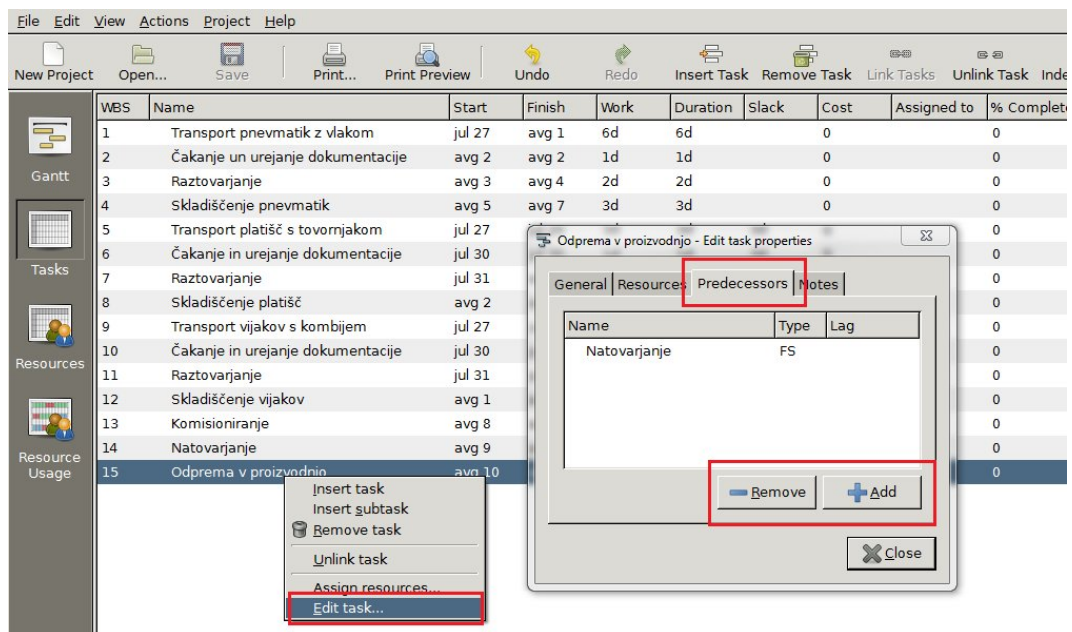
FS Finish to start	Konec - Začetek
FF Finish to finish	Konec - Konec
SS Start to start	Začetek - Začetek
SF Start to finish	Začetek - Konec

Predstavimo še funkcijo, ki koristi predvsem pri večjih projektih, kjer se naloge med seboj prepletajo in potekajo vzporedno. Pomembno je, da poznamo kritično pot, saj je od nje odvisno celotno trajanje projekta. Projekt preprosto skrajšamo s tem, da zmanjšamo trajanje kritične poti, ki jo označimo na naslednji način: pod *View* izberemo možnost *Highlight critical Tasks* (Slika 2.13).

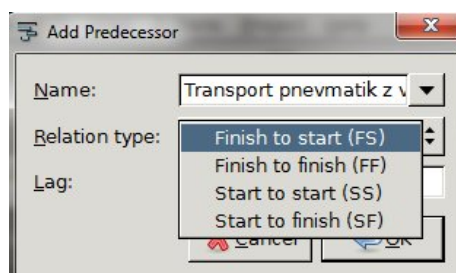
Kritična pot je najdaljše nespremenljivo zaporedje aktivnosti, ki so vzročno-posledično povezane. Njihovega trajanja po eni strani ne moremo skrajšati, po drugi pa bi vsaka zakasnitev aktivnosti s te poti pomenila tudi zakasnitev



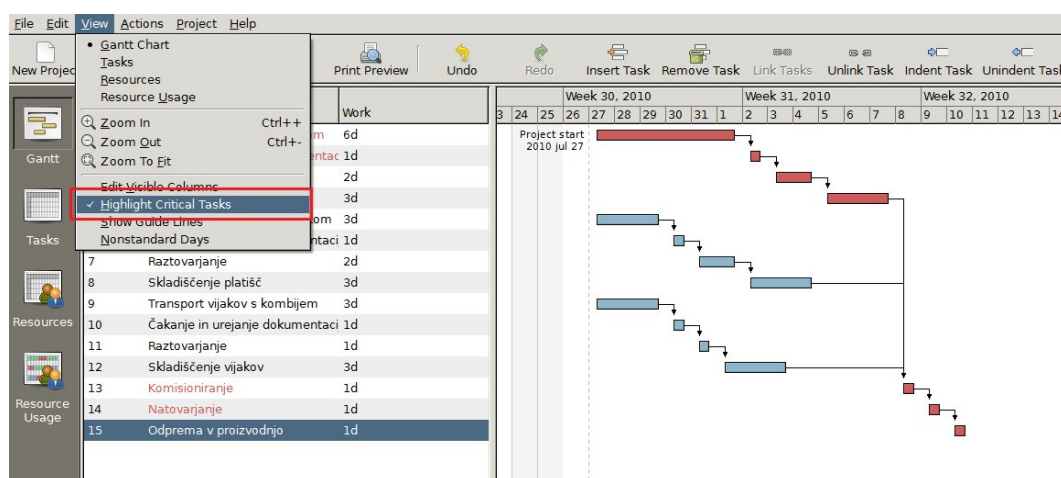
Slika 2.10: Povezava aktivnosti



Slika 2.11: Dodajanje povezave aktivnosti



Slika 2.12: Oblika povezave med dvema aktivnostma



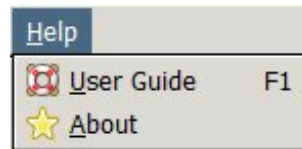
Slika 2.13: Kritična pot

projekta. Dolžina (trajanje) te poti je tisti čas, ki ga za projekt porabimo v vsakem (tudi najboljšem) primeru.

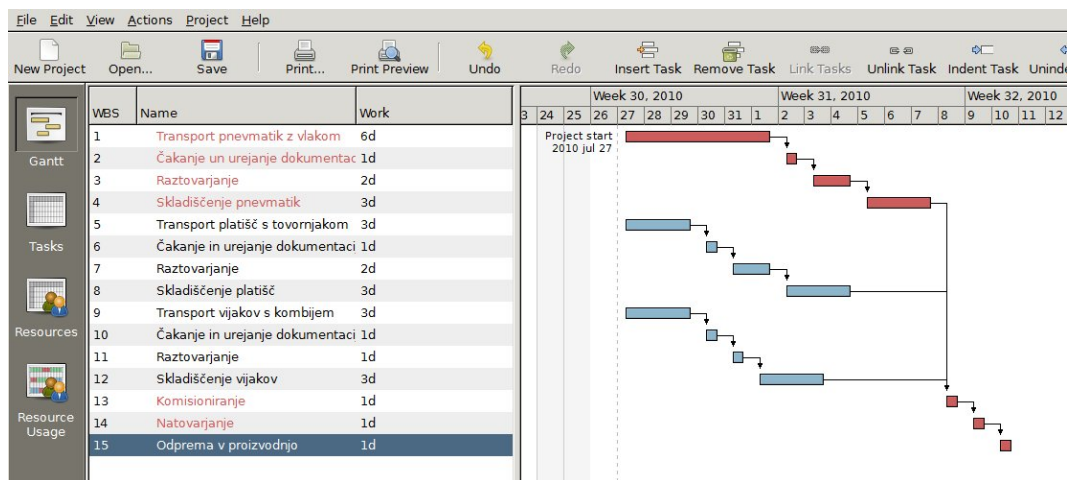
Uporabnik lahko več o programu izve v razdelku *Help > User Guide* (hitra bližnjica: F1), ki se nahaja v menijski vrstici. Razdelek nudi hitro pomoč v primeru, če se uporabnik sreča s kakšno oviro ali pa morda želi zgolj izpopolniti svojo znanje. Pot do pomoči oz. uporabniškega vodiča prikazuje Slika 2.14.

Z natančno opredelitvijo vseh aktivnosti v izoblikovanem Ganttovem diagramu ali diagramu aktivnosti smo prikazali kritično pot naših procesov oz. aktivnosti. Slednje ni možno skrajšati, vsaka dodatna zakasnitev pa bi pomenila podaljšanje celotnega procesa. Na Sliki 2.15 je z rdečo barvo jasno opredeljena kritična pot.

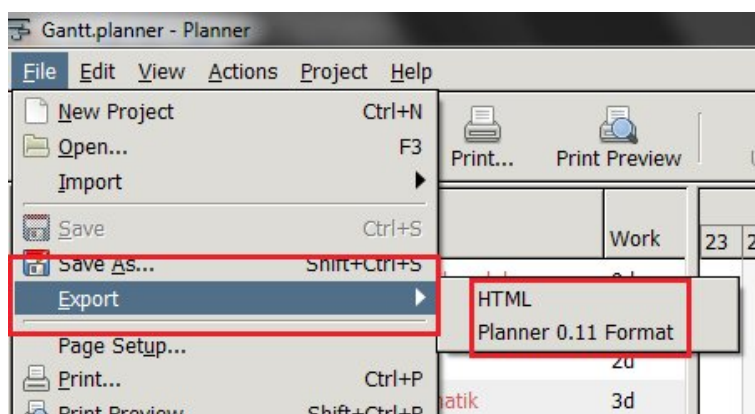
Programsko orodje Planner je kompatibilno z drugimi podobnimi programskimi orodji, kot je na primer plačljivi Microsoft Project, ki omogoča izvoz datotek oz. projektov drugim podobnim programskim orodjem v berljivi



Slika 2.14: Pomoč



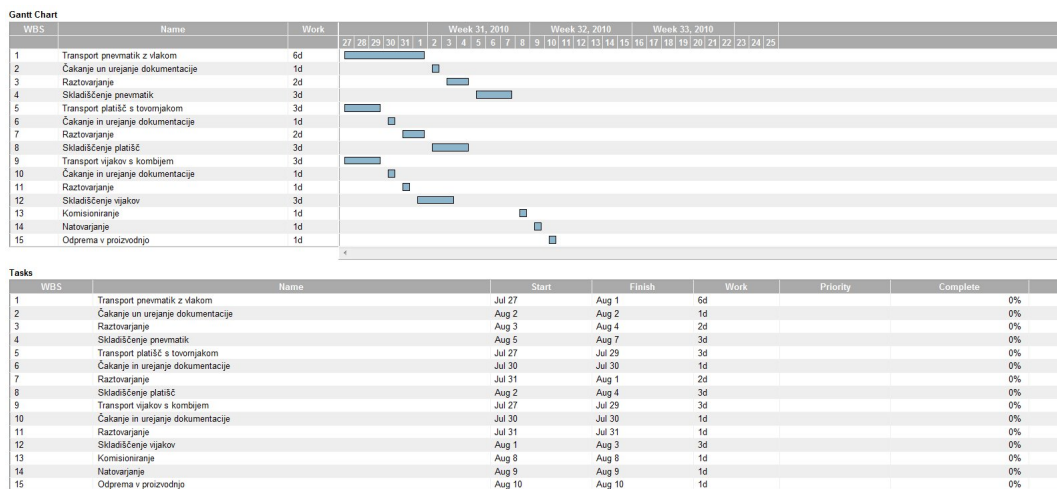
Slika 2.15: Ganttov diagram



Slika 2.16: Izvoz dokumentov

obliki. Dokumente oz. končane projekte je možno izvoziti v obliko HTML ali Planner 0.11 format (glej Sliko 2.16).

V kolikor se odločimo za izvoz dokumenta v obliko HTML, je možno dokument brati in pregledovati s spletnim brskalnikom (glej Sliko 2.17). V kolikor se odločimo za izvoz dokumenta v obliko Planner 0.11 Format, ga lahko odpremo tudi v programskem orodju Microsoft Project.



Slika 2.17: Pregledovanje dokumenta v HTML obliki s pomočjo spletnega brskalnika

Povzetek

S programskim orodjem Planner izdelamo diagram aktivnosti za naše procese. Končni rezultat, Ganttov diagram z označeno kritično potjo izdelamo po naslednjem postopku:

- na list papirja zapišemo tabelo aktivnosti ter čase (začetek, trajanje, konec);
- zaženemo program Planner;
- določimo koledarske nastavitve;
- zapišemo vse aktivnosti/podaktivnosti ter njihove čase trajanja;
- med sabo povežemo soodvisne aktivnosti;
- ugotovimo kritično pot.

Program Planner je odlično orodje za vodenje projektov, ki ga lahko uporabljajo tako študentje pri svojem študiju kot podjetja pri svojem delu. Je brezplačno in enostavno za uporabo, zato ga priporočamo vsem, ki se pri svojem delu srečujejo z nezahtevnimi in preprostejšimi projekti.

Poglavje 3

DIA - Načrtovanje diagramov

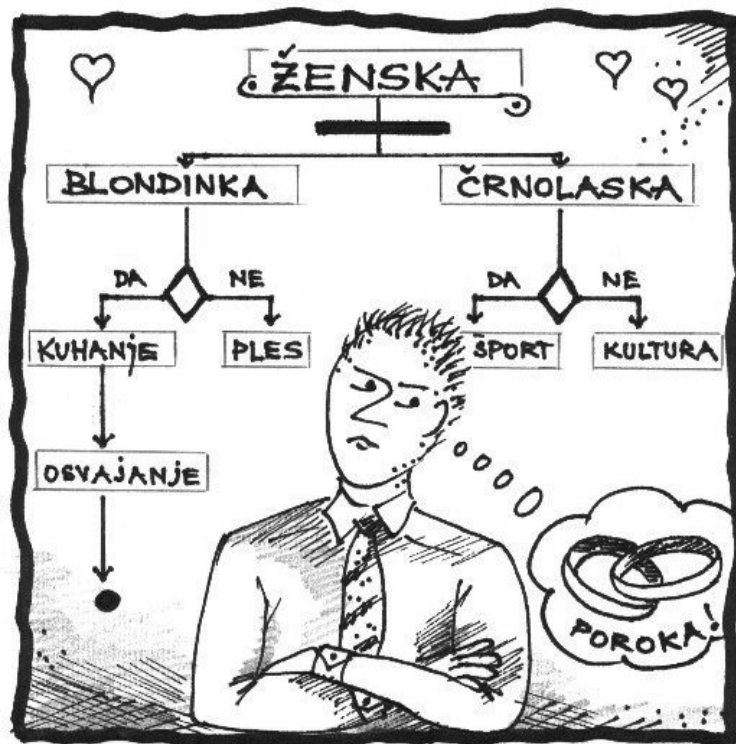


Diagram poteka (aktivnosti)
UML jezik
Primer: načrtovanje diagrama z UML

3.1 Teoretično ozadje

3.1.1 Diagram poteka/aktivnosti

Diagram poteka (*Flow chart*) je diagram za prikaz možnih poti podatkov skozi sistem oz. je eden izmed načinov zapisa algoritma. Diagram prikazuje natančno zaporedje operacij, ki jih programsko orodje pri obdelavi podatkov izvede. Različni grafični simboli predstavljajo vnos in izpis podatkov, odločitve, razvejitve in podprogramska orodja. Uporablja se v računalništvu, matematiki, pravu, logistiki in v mnogih drugih vedah. Danes se diagrami poteka oz. diagrami aktivnosti rišejo predvsem v standardiziranem opisnem jeziku UML (*Unified Modeling Language*).

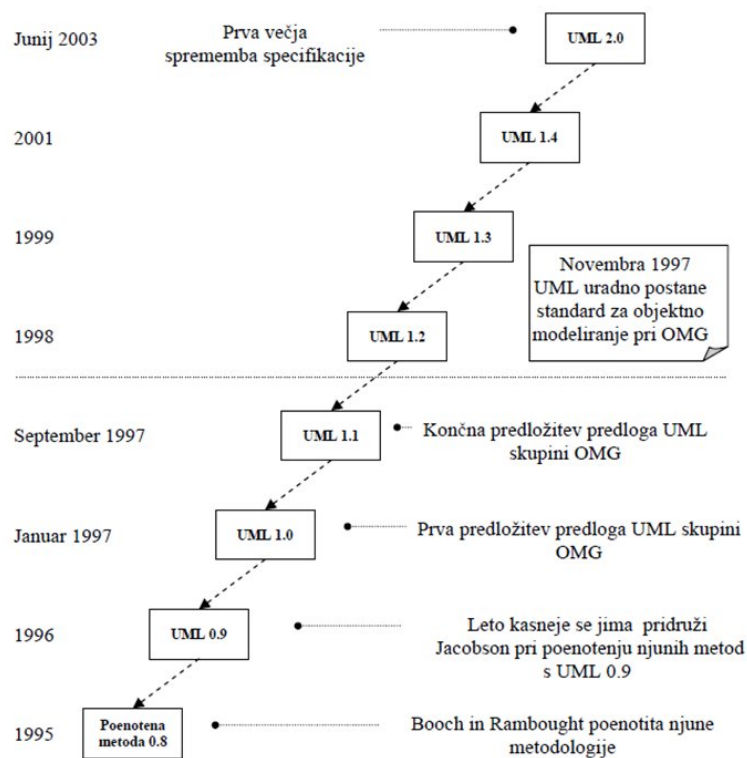
Diagram poteka prikazuje prehode iz ene aktivnosti na drugo. Elementi diagrama aktivnosti so: aktivnosti, prehodi med aktivnostmi, razvejitve (sočasne razvejitve), sočasno stičišče, linija toka dogodkov in objektni tok. Kot rezultat aktivnosti je akcija, katere posledica je prehod v drugo aktivnost ali pa le vračanje vrednosti. Iz diagrama je jasno razvidna odgovornost za posamezne aktivnosti. Če so v tok dogodkov vpleteni pomembni objekti, jih lahko s pomočjo objektnega toka dodamo v diagram aktivnosti.

3.1.2 Standard UML

Programsko orodje UML je grafični jezik za vizualizacijo, modeliranje, specifikacijo, konstruiranje in dokumentiranje programske opreme. UML predstavlja zbir najboljših postopkov v praktično objektno orientiranem modeliranju. Vedeti moramo, da UML ni metodologija za razvoj informacijskega sistema, ampak jezik, skupek tehnik, ki jih lahko uporabljamo skupaj s poenotenim procesom ali katero drugo metodologijo.

Začetek razvoja UML umeščamo v leto 1994. Takrat je bilo na voljo okoli 50 različnih metodologij za objektno usmerjeno analizo in načrtovanje. Ideja raziskovalcev, ki so raziskovali na tem področju je bila združitev metodologij v enoten jezik za modeliranje, ki bi vseboval prednosti različnih metodologij. Tako je nastal UML. Med novimi metodami, ki so se pojavile v tistem času so izstopale Boochova metoda, Jacobsonova metoda OOSE (*Object-Oriented Software Engineering*) in Rumbaughtova metoda OMT (*Object Modeling Techniq*). V sredini 90-tih let so avtorji teh treh metod zasnovali nov modelni jezik, ki naj bi bil podprt v vseh treh metodah (glej Sliko 3.1).

Cilj UML je uporabniku ponuditi standarden, izrazno močan vizualni modelirni jezik; omogočiti mehanizme razširjanja in specializacije osnovnih konceptov; biti neodvisen od določenega programskega jezika in razvojnega procesa; ponuditi formalno osnovo za razumevanje jezika za modeliranje;



Slika 3.1: Zgodovinski razvoj specifikacije poenotene modelnega jezika UML

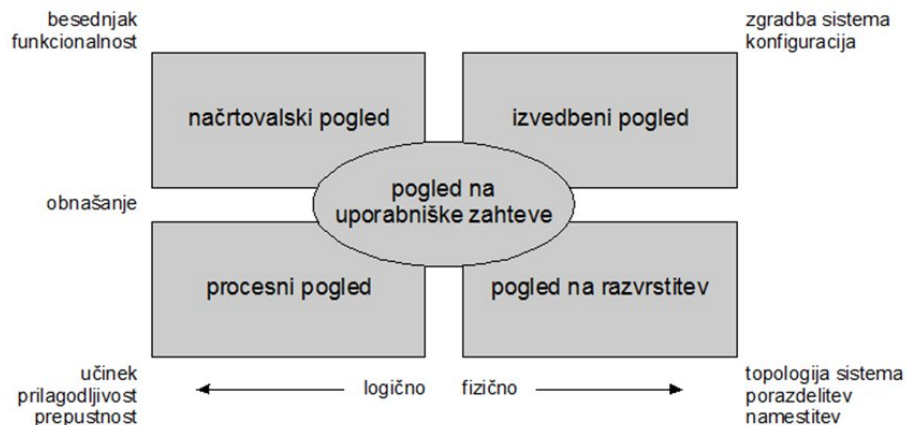
Vir: [75]

spodbuditi rast tržišča objektnih orodij; podpirati visoko nivojske razvojne koncepte, kot so sodelovanje, ogrodja, vzorci in komponente ter integrirati najboljše prakse.

Na sistem UML je moč gledati skozi pet različnih pogledov, kar prikazuje Slika 3.2.

Modeliranje z UML

Pri modeliranju z UML, vrstni red ni predpisan. Koristno je, da najprej modeliramo pogled na uporabniške zahteve, nato načrtovalski in procesni pogled, na koncu pa še pogled na komponente in razvrstitev. Da lahko razumemo delovanje UML, je potrebno poznati pomen treh osnovnih gradnikov, in sicer: elementov, povezav in diagramov.



Slika 3.2: Pet pogledov na sistem UML

Vir: [97]

Elementi

V UML-ju se pojavljajo štiri vrste elementov: strukturni elementi, elementi obnašanja, elementi združevanja in elementi opomb. Strukturni elementi so največkrat statični deli modela, ki predstavljajo element shematično ali fizično. Mednje prištevamo: razrede, vmesnike, sodelovanje, primere uporabe komponente in vozlišča [96].

- Razred je opis nabora objektov, ki delijo enake attribute, operacije, povezave in semantiko. Vmesnik definira povezavo med specifikacijo, s pomočjo katere ugotovimo kaj abstrakcija počne in kako to počne med implementacijo. Gre za zbirko operacij, ki opisujejo kako razred ali komponenta deluje. Sodelovanje je skupina razredov, vmesnikov in drugih elementov, ki delujejo skupaj v kooperativnem obnašanju, katerega rezultat je izboljšano delovanje sistema. Primer uporabe je opis niza zaporednih akcij, ki jih sistem izvaja z namenom, da določenemu akterju prinese rezultat, ki ga lahko opazujemo [96].
- Komponenta je fizični in zamenljivi del sistema, ki omogoča izvedbo niza vmesnikov. Vozlišče je fizični element, ki obstaja v času izvajanja programa in predstavlja programski vir, z lastno rezervacijo pomnilnika [96].
- Elementi obnašanja so dinamičen del UML modela in predstavljajo

časovno in krajevno obnašanje modela. Elemente obnašanja delimo na interakcije in diagrame stanj. Interakcija je obnašanje, ki vključuje nabor sporočil, ki se izmenjujejo med objekti v okviru določenega konteksta za doseganje določenega cilja. Diagram stanja je obnašanje, ki opisuje zaporedje stanj objekta ali interakcij med objekti, kot odgovor na dogodke v sistemu. Vključuje številne elemente stanja, prehode med posameznimi objekti, dogodki in aktivnosti [96].

- Element združevanje je organizacijski del UML, katerega glavni element imenujemo paket, ki omogoča združevanje elementov. Služi za združevanje modelnih elementov na nivoju arhitekturnega načrtovanja v večje skupine, s katerimi je lažje upravljati. Zadnji izmed elementov je element opomb, katerega uporabljamo za pojasnjevanje in opisovanje delovanja posameznih elementov v modelu [96].

Povezave

Kadar govorimo o povezavah, govorimo o spoju dveh elementov. V UML se pojavljajo štiri vrste povezav: odvisnosti, asociacije, generalizacije in realizacije.

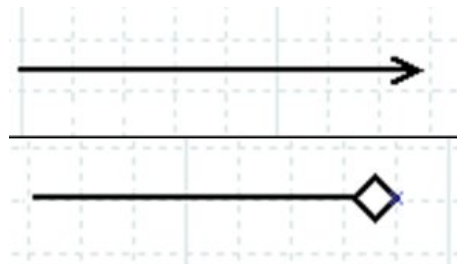
- Odvisnost je semantična povezava dveh elementov, pri katerih lahko sprememba neodvisnega elementa povzroči spremembo odvisnega elementa. Odvisnost se uporablja takrat, ko poizkušamo pokazati, da ena stvar uporablja drugo (glej Sliko 3.3).
- Asociacija je strukturirana povezava s katero pojasnujemo nabor vezi, katere so povezave med objekti. Kot enega izmed posebnih primerov lahko izpostavimo agregacijo, ki predstavlja povezavo med celoto in deli te celote (glej Sliko 3.4).
- Generalizacija je povezava, ki predstavlja povezavo specializacije in generalizacije, v kateri so objekti specializiranega elementa nadomestljivi z elementi generaliziranega elementa (glej Sliko 3.5).
- Realizacija je semantična povezava med elementoma, pri kateri eden element določi dogovor, ki ga mora drugi izvršiti. Le ta se pojavi med vmesniki v razredih ali med komponentami, ki jih realizirajo (glej 3.6).

Diagrami

Diagram je vizualni model, ki predstavlja pogled na določenem nivoju abstrakcije. V UML ločimo strukturne diagrame, ki predstavljajo statično



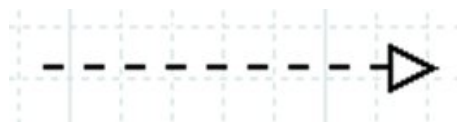
Slika 3.3: Odvisnost



Slika 3.4: Asociacija in agregacija



Slika 3.5: Generalizacija



Slika 3.6: Realizacija

Strukturni diagrami	Diagrami, ki prikazujejo delovanje in uporabo	Diagrami, ki prikazujejo interakcije
Razredni diagram (<i>Class Diagram</i>)	Diagram primerov uporabe (<i>Use Case Diagram</i>)	Diagram zaporedja (<i>Sequence Diagram</i>)
Objektni diagram (<i>Object Diagram</i>)	Diagram aktivnosti (<i>Activity Diagram</i>)	Komunikacijski diagram (<i>Communication Diagram</i>)
Diagram komponent (<i>Component Diagram</i>)	Diagram načrtovanja stanj (<i>State Machine Diagram</i>)	Časovni diagram (<i>Timing Diagram</i>)
Diagram razvrstitve (<i>Deployment Diagram</i>)		Pregledni diagram interakcij (<i>Interaction Overview Diagram</i>)
Strukturni diagram (<i>Composite Structure Diagram</i>)		
Diagram paketov (<i>Package Diagram</i>)		

Tabela 3.1: Vrste diagramov

strukturo sistema, diagrame, ki prikazujejo delovanje in uporabo ter diagrame, ki prikazujejo interakcije. Diagrami, ki prikazujejo delovanje in uporabo ter diagrami, ki prikazujejo interakcijo, predstavljajo tako imenovano dinamično strukturo sistema (Tabela 3.1).

3.2 O programskem orodju

Urejevalnik diagramov Dia (v nadaljevanju Dia) je odprtokodno programsko orodje za kreiranje tehničnih diagramov. Deluje na različnih operacijskih sistemih (Linux, Windows itd.). Je enostavno za učenje in dovolj prožno, saj uporabnikom omogoča ustvarjanje številnih diagramov. Programsko orodje uporabljamo v različnih industrijskih panogah. Primer: električar uporablja Dia za ustvarjanje diagrama, kjer prikaže delovanje vezja; programer ustvarja diagrame za prikaz izvedbe poti.



Slika 3.7: Urejevalnik diagramov Dia

Prenos in namestitvev

Programsko orodje Dia lahko prenesemo in namestimo iz spletnega naslova Live.gnome [6] s klikom na razdelek *Download* na uvodni spletni strani. Pred prenosom lahko izbiramo med različnimi verzijami programskega orodja, odvisno kateri operacijski sistem uporabljamo. Uporabniki operacijskega sistema Ubuntu lahko programsko orodje Dia enostavno poiščejo, prenesejo in namestijo s pomočjo orodja Synaptic (glej Sliko 3.8).

Diagram je lahko sestavljen iz predmetov, ki so različnih oblik, črt, barv, velikosti itd. V razdelku *Orodja* se nahaja paleta vnaprej določenih predmetov, vključno s preprostimi oblikami, črtami in specifičnimi predmeti. Če želimo dodati določen predmet na platno, preprosto kliknemo na zeleni predmet in ga dodamo s klikom na miško. Po platnu premikamo predmete z miško in jih na to med sabo povezujemo (glej Sliko 3.9).

Dia vključuje nabor standardnih oblik in linij, ki so že vnaprej določene. Vključujejo diagram poteka, UML diagram, diagram omrežja, kronogram in mnoge druge diagrame. Glede na področje raziskovanja si lahko izberemo predmete, ki jih potrebujemo za oblikovanje diagrama.

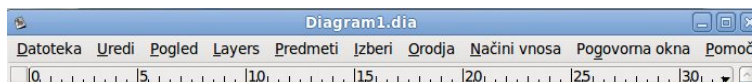
Že izdelan diagram lahko shranimo v poljubno mapo v svojem računalniku. To storimo tako, da v orodni vrstici *Datoteka* kliknemo možnost *Shrani kot*. Odpre se novo okno, v katerem napišemo ime naše datoteke in jo shranimo na zeleno mesto (glej Sliko 3.10).

Da v nadaljevanju zagotovimo normalno shranjevanje in tiskanje dokumenta, v razdelku *Datoteka* izberemo možnost *Nastavitev strani*, kjer izbiramo med velikostjo papirja, usmerjenostjo, robovi in povečavo. Dia podpira izvoz v številne formate, kot so: *.cgm*, *.eps*, *.dia*, *.jpeg*, *.svg*, *.pdf*, *.jpeg ipd.*, katera izberemo iz orodne vrstice *Izvozi* (glej Sliko 3.11).

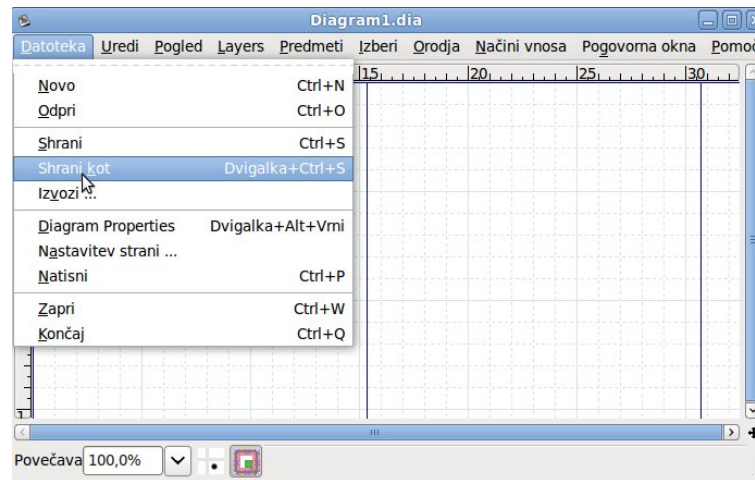
Ob zagonu Dia se ustvari prazno okno. V primeru, ko želimo novo okno za ustvarjanje v razdelku *Datoteka* izberemo možnost *Novo*. Odpre se novo polje za risanje diagramov. Osnovno programsko orodje Dia je sestavljeno iz dveh oken. V prvem oknu so zapisana programska orodja, za izdelavo diagramov



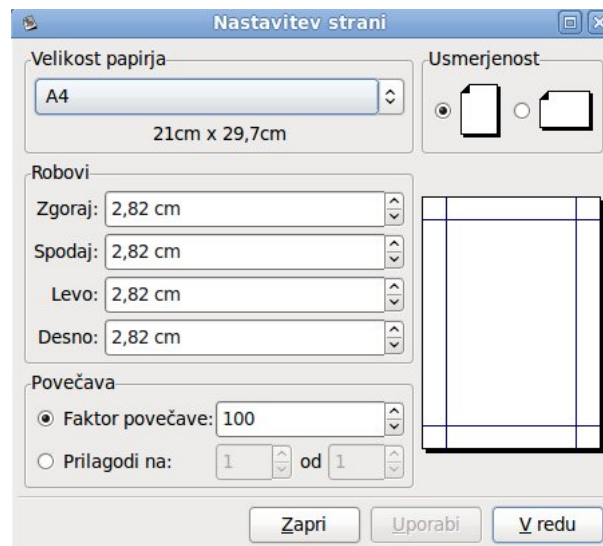
Slika 3.8: Prenos programskega orodja Dia



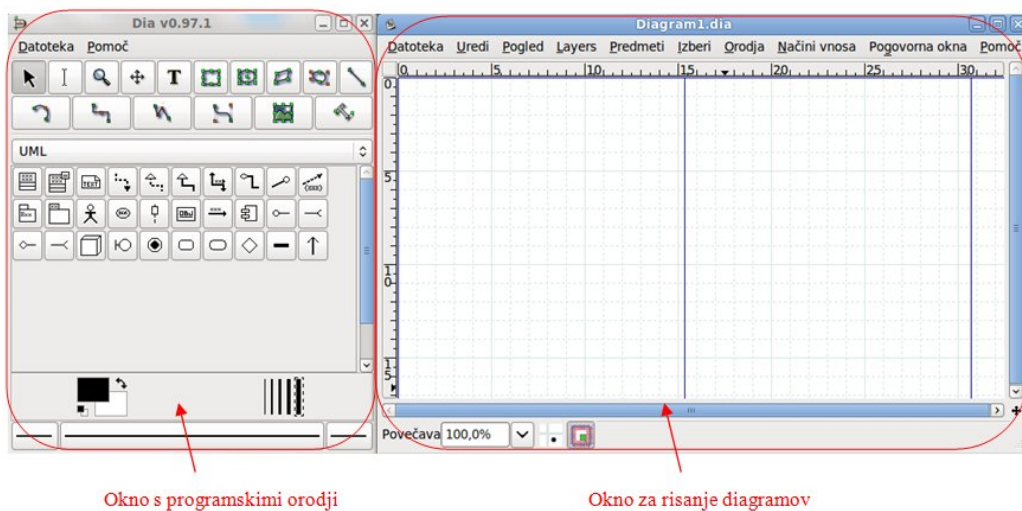
Slika 3.9: Menijska vrstica



Slika 3.10: Orodna vrstica - Datoteka



Slika 3.11: Nastavitev strani



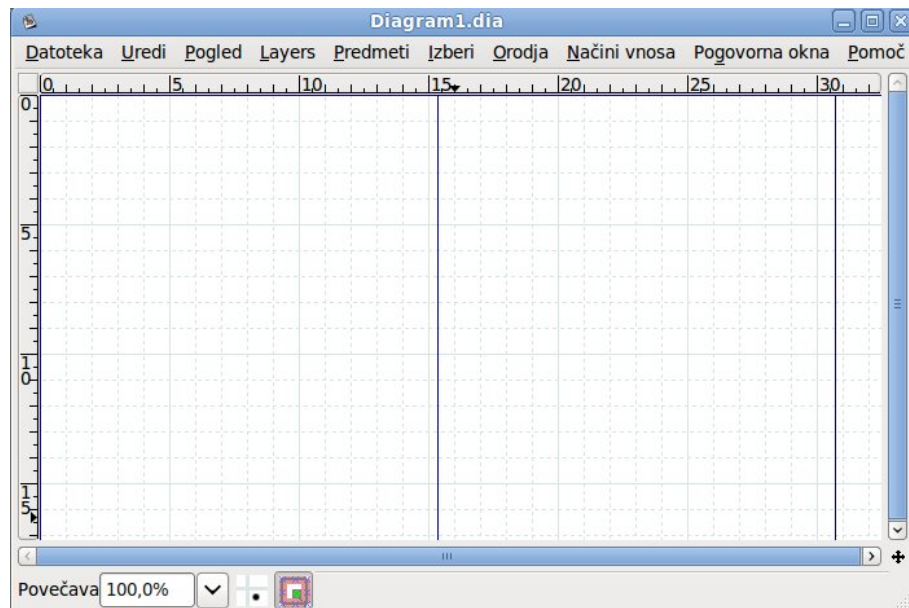
Slika 3.12: Prikaznih osnovnih in specifičnih orodij

(okno s programskimi orodji). V drugem pa je prostor za oblikovanje le teh. Prvi sklop orodij uporabljamo za prilagoditev diagrama (spremeni predmet, urejanje besedila, povečava in drsenje po diagramu). Naslednje ikone so vgrajene za vstavljanje osnovnih predmetov (besedilo, škatla, elipsa, mnogokotnik, črta, slika). Srednji del predstavljajo specifična orodja, torej tista, ki smo jih izbrali iz seznama, glede na področje diagrama (npr. UML, diagram procesa...). Kvadrata, ki se nahajata pod specifičnimi orodji, omogočata nastavitve privzetih barv ospredja in ozadja za vse nove predmete diagrama ter debelin črt, ki jih lahko poljubno spreminjamo.

Slika 3.12 prikazuje osnovni okni, ki jih vidimo v Ubuntu. V podobni zgradbi se programsko orodje prikaže tudi v okolju Windows, le da okni nista ločeni - pojavljata se kot celota.

Na osnovnem oknu je prikazan osnovni meni, ki je namenjen oblikovanju diagrama. Uporabniku je omogočeno, da poveča pogled osnovnega okna maksimalno na 2500 % oz. da ga pomanjša minimalno do 5 %. Povečanje služi predvsem zato, da si lahko natančneje ogledamo dele diagrama. S pomanjšanjem vidimo velikost izdelanega diagrama in kaj je potrebno spremeniti, da ga lahko natisnemo oz. shranimo v pdf. obliko. Na osnovnem oknu je dodano še merilo (ravnilo), ki omogoča natančnejši izris diagrama (glej Sliko 3.13).

V orodni vrstici *Pogled* (v okolju Windows) imamo možnost uporabe številnih funkcij, kot so: uporaba celotnega zaslona (F11), preklopi na mrežo in izključni



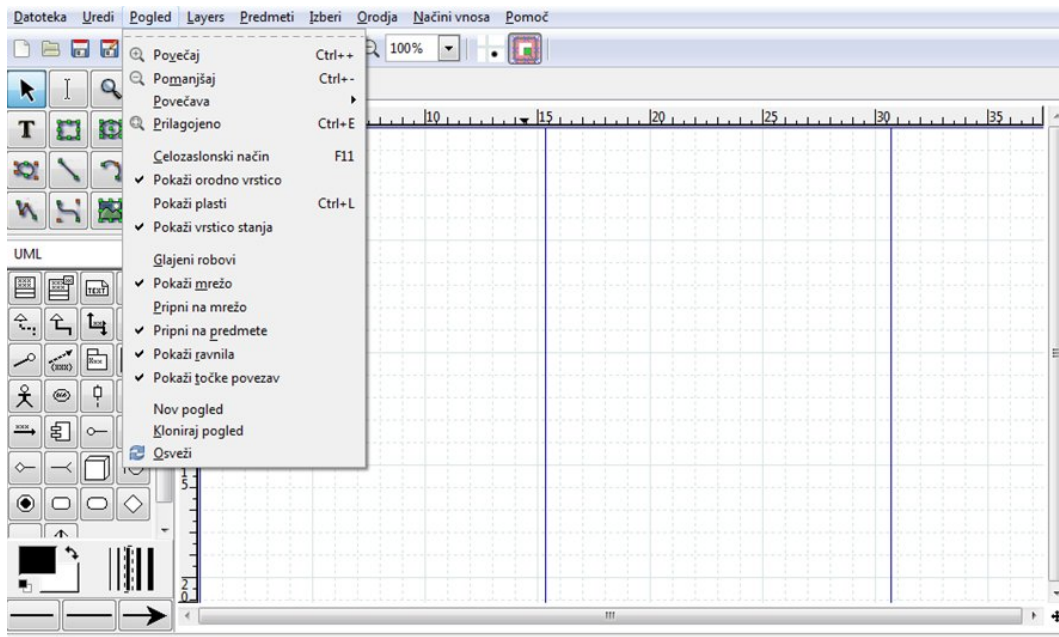
Slika 3.13: Osnovno okno

iz mreže, pokaži ravnilo, nov pogled, glajeni robovi itd. V okolju Ubuntu Linux je zapisanih manj možnosti, kar je razvidno tudi iz Slik 3.14 in 3.15.

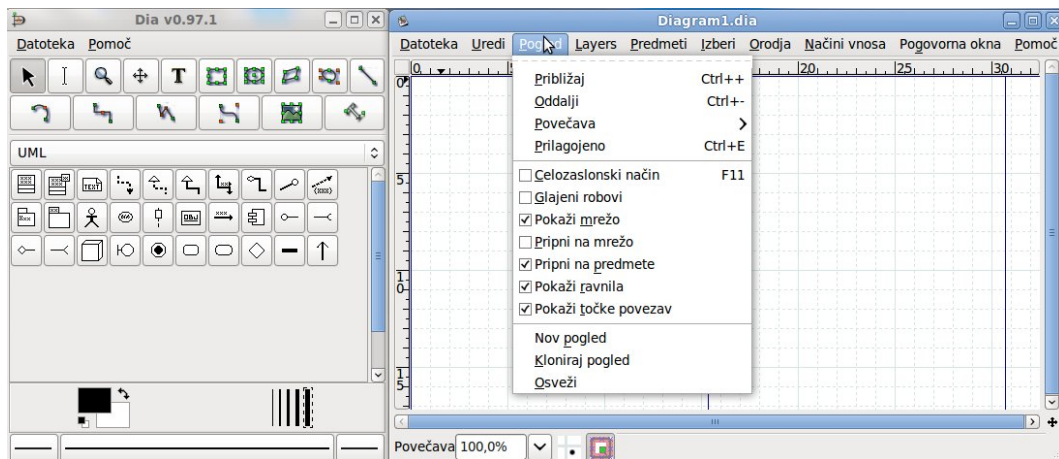
Če uporabnik oceni, da pri svojem delu potrebuje tudi druga orodja, kot so zapisana v osnovnih skupinah razvrščenih datotek, lahko to stori na preprost način. V razdelku *Datoteka* izberemo možnost *Listi in predmeti*, kjer označimo določen predmet, izberemo tipko *Kopiraj* in ga prenesemo v drugo kategorijo (glej Sliko 3.16).

Program DIA vsebuje 36 različnih oblik orodij iz najrazličnejših področij (glej Sliko 3.17).

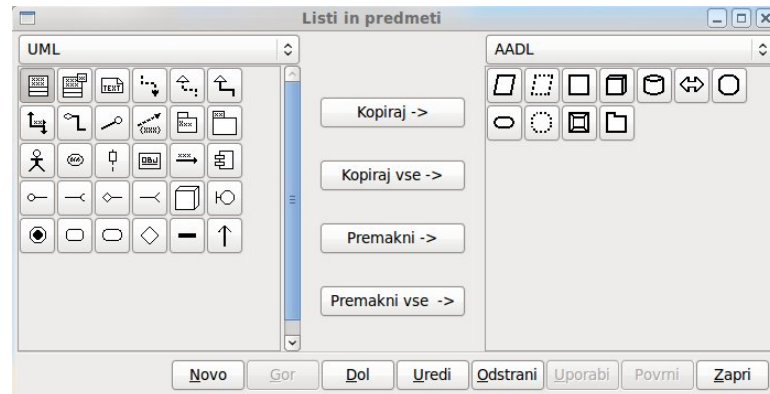
Na Sliki 3.18 je prikazan primer specifičnih standardnih UML orodij za prikaz diagrama aktivnosti ter ostala znana orodja (diagram poteka, strojništvo, luči...).



Slika 3.14: Orodna vrstica - Pogled (Windows)



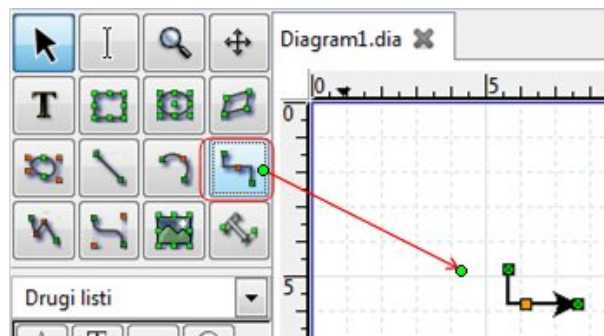
Slika 3.15: Orodna vrstica - Pogled (Ubuntu Linux)



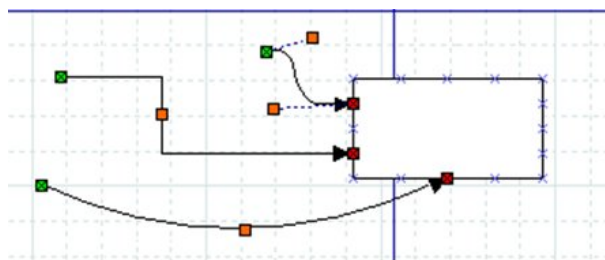
Slika 3.16: Listi in predmeti



Slika 3.17: Možnosti oblik orodij



Slika 3.19: Dodajanje predmetov na platno



Slika 3.20: Oblikovanje in povezovanje predmetov

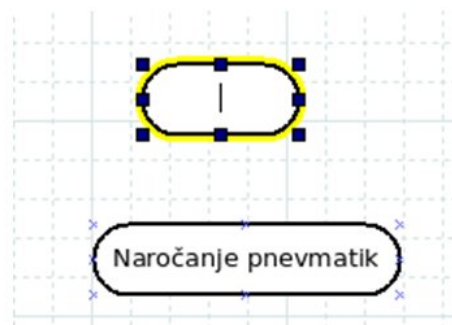
Dodajanje predmetov na platno se izvede s klikom na ikono predmeta, ki ga želimo dodati. Prenesemo ga iz orodij na osnovno okno, kjer ga nato oblikujemo. Če želimo predmet premakniti, kliknemo kjerkoli znotraj predmeta (ali nekje na črto) in ga z miško povlečemo na zeleno mesto (glej Sliko 3.19).

V številnih diagramih so oblike med seboj povezane z eno od osnovnih linij predmetov (oznaka "x"). Obstajajo pa tudi povezane točke v sredini vsake oblike, s pomočjo katerih lahko poljubno spreminjamo obliko predmeta oz. črte. Zelen kvadrček "x" omogoča, da linijo povežemo še z drugimi linijami (glej Sliko 3.20).

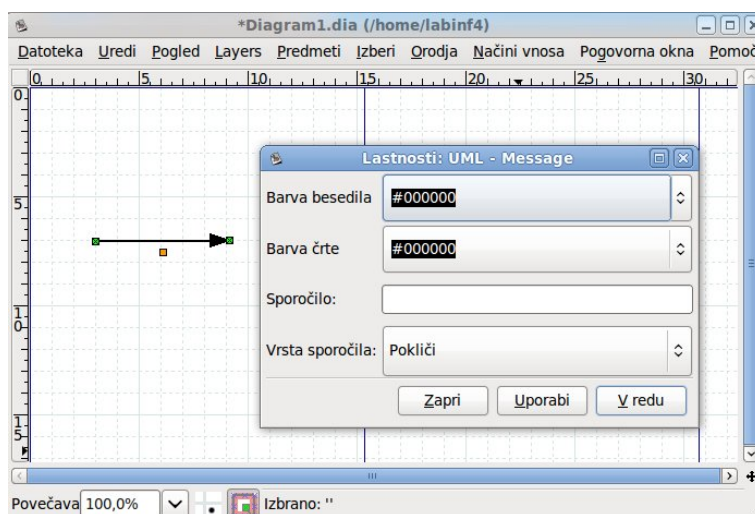
Besedilo vnašamo tako, da najprej izberemo predmet, ga prenesemo v polje risanja in kliknemo nanj ter vnesemo besedilo. Velikost pisave, poravnave in druge elemente, spreminjamo tako, da dvokliknemo na predmet (glej Sliko 3.21).

Program Dia vsebuje tudi možnost spreminjanja barv predmetov in črt. Barvo spremenimo tako, da z desnim klikom na miško, izberemo možnost lastnosti (glej Sliko 3.22).

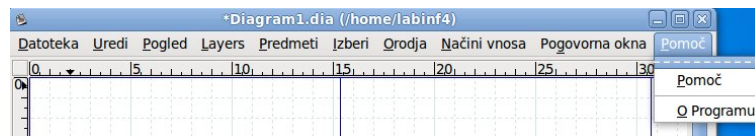
Uporabnik lahko več o programskem orodju izve v razdelku *Pomoč*, ki se nahaja v menijski vrstici v orodnem oknu. Razdelek *Pomoč* omogoča hitro



Slika 3.21: Dodajanje besedila v izbrani predmet



Slika 3.22: Lastnosti diagrama



Slika 3.23: Pomoč

seznanitev s potekom dela, v primeru, da naletimo na kakšno oviro ali pa če želimo zgolj izpopolniti svoje znanje. Pot do pomoči je prikazana na Sliki 3.23.

Problem

V namišljenem podjetju se vsakodnevno srečujemo s procesi dela, ki jih je potrebno medsebojno usklajevati in dopolnjevati. Zavedati se je potrebno, da pravilno načrtovanje diagramov aktivnosti pripomore k učinkovitosti delovanja podjetja.

S programskim orodjem Dia prikažemo diagram aktivnosti v podjetju. Osredotočimo se na proces oskrbe, ki vključuje prikaz procesa naročanja, dostave, izmenjave dokumentov, grobega in finega prevzema platišč, reklamacij in skladiščenja. Smiselnost izdelave diagramov se pokaže predvsem v zahtevnejših primerih, ko aktivnosti ne moremo obravnavati ločeno, saj se medsebojno prepletajo in dopolnjujejo. V podjetju se vsakodnevno srečujemo s procesi dela, ki jih je potrebno medsebojno usklajevati in dopolnjevati. Zavedati se je potrebno, da pravilno načrtovanje diagramov aktivnosti pripomore k učinkovitosti delovanja.































3.3 Uporaba

Za prikaz danega problema preučimo diagram aktivnosti s standardnim jezikom modeliranja UML. Program Dia na Sliki 3.24 prikazuje predmete s katerimi izdelamo diagram aktivnosti. Na vsakem posameznem predmetu so zapisane njegove lastnosti in njegov pomen.

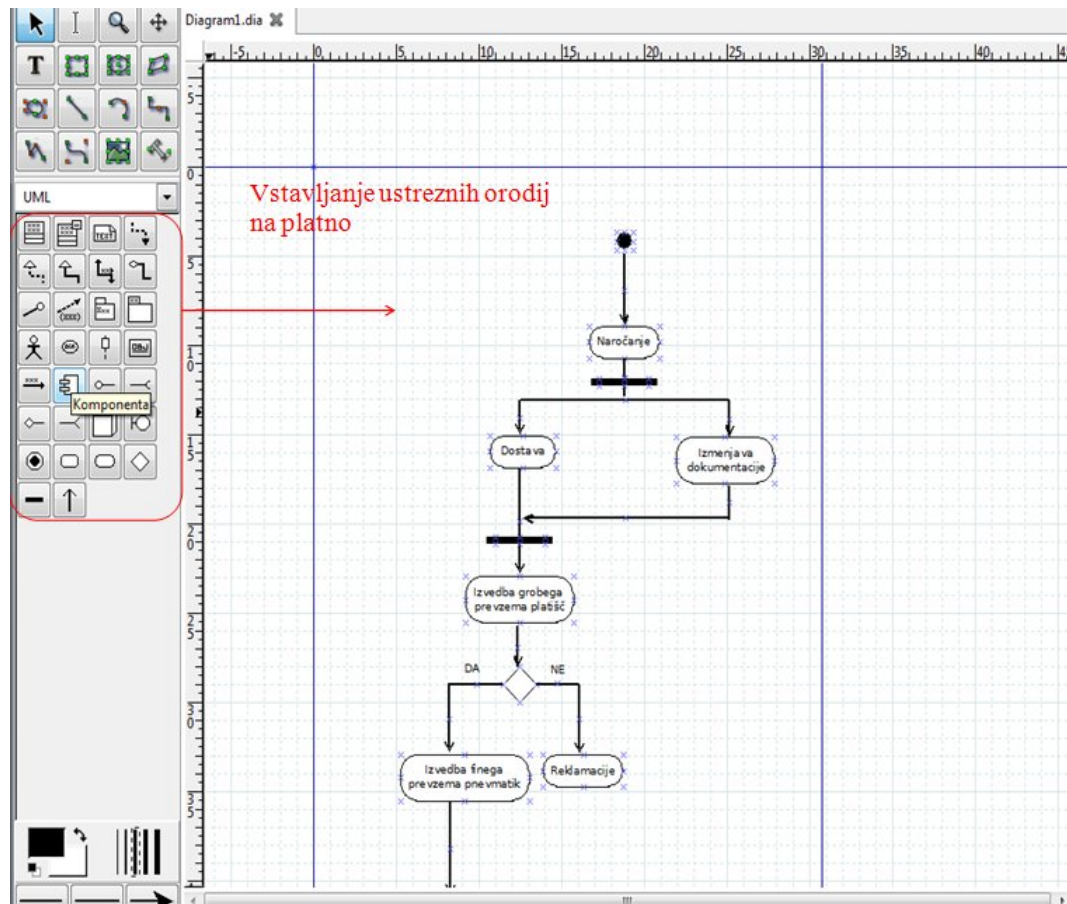
V nadaljevanju je prikazan preprost primer izdelanega diagrama z UML jezikom. Delo pričnemo tako, da na podlagi zastavljenega problema želimo narisati diagram aktivnosti za platišča, vse od naročanja, dostave, prevzema do skladiščenja.

Diagram aktivnosti je nazorneje prikazan na Sliki 3.26.

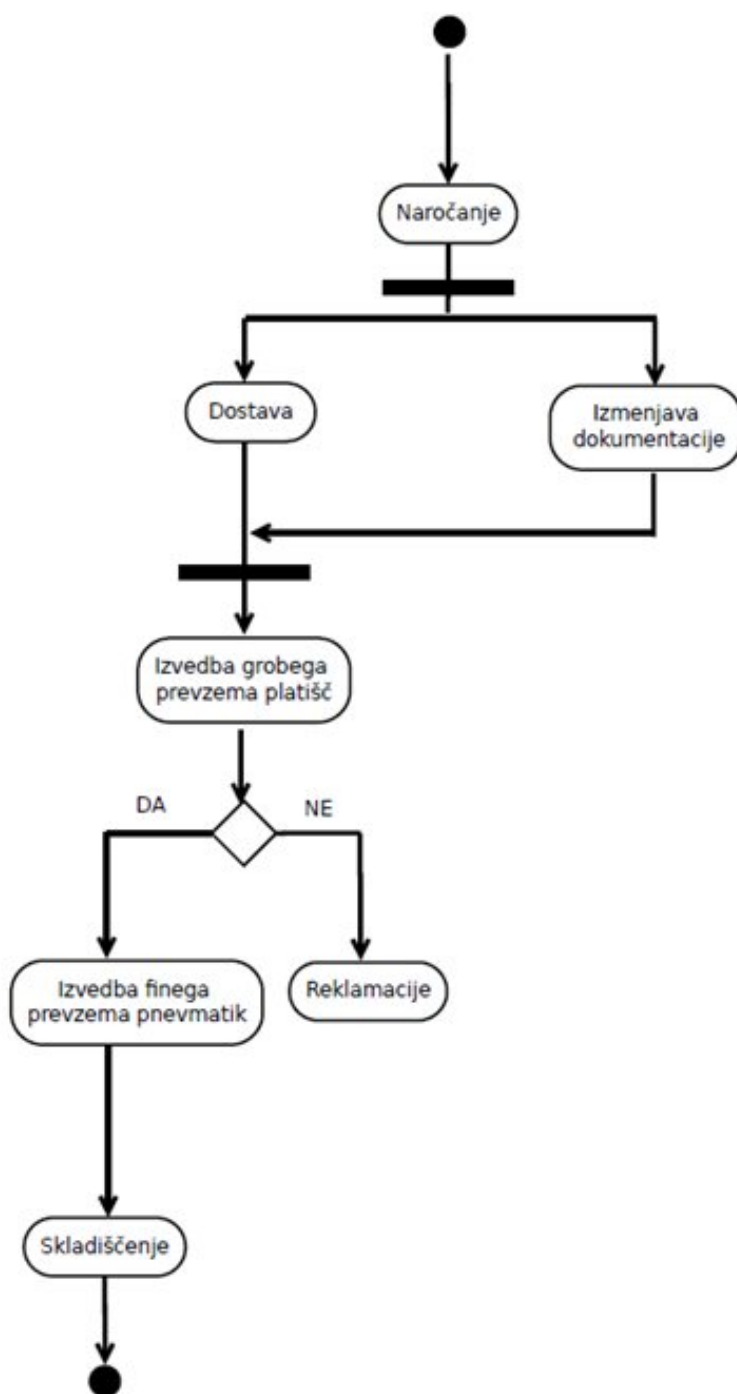
Dia ponuja možnost pogleda v tako imenovano *Drevo diagrama*, ki ga najdemo pod razdelkom *Datoteka*. Drevo diagrama je ena izmed možnosti,

 Razred	 Realizira, implementira določen vmesnik
 Posploševanje, delovanje	 Omejitev (dodaj omejitve na nekaj)
 Majhen paket	 Časovna linija UML
 Predmet	 Zbiralnik
 Razred predlog	 Dogodkovni odtok
 Povezava (dva razreda sta med sabo povezana)	 Ikona stereotipa razreda
 Velik paket	 Razvejaj
 Sporočilo	 Razvejaj/spoji
 Opomba	 Vozlišče
 Agregacija (en razred je del drugega)	 Začetno/končno stanje
 Igralec/izvršilec dejanja (dogodka)	 Vir dogodka
 Komponenta	 Pretvorba
 Odvisnost	 Stanje
 Implementiranje	 Aktivnost
 Primer uporabe	 Faseta

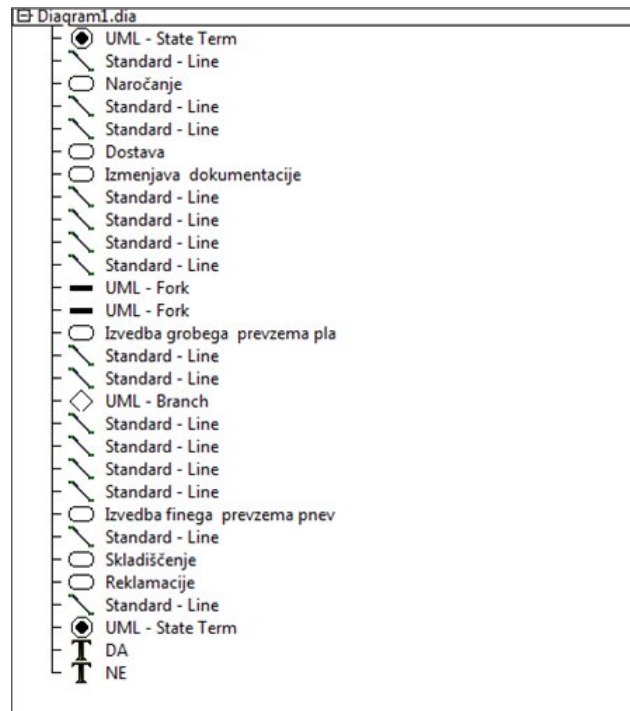
Slika 3.24: Orodja UML



Slika 3.25: Lastnosti diagrama



Slika 3.26: Diagram aktivnosti z UML



Slika 3.27: Drevo diagrama

ki prikazuje, kako je sam diagram sestavljen, kateri predmeti so vključeni in kaj le ti pomenijo (glej Sliko 3.27).

Povzetek

Diagram aktivnosti je ključnega pomena za planiranje in prikaz posameznih elementov ali celotnega dela projekta. S programskim orodjem Dia prikažemo diagram aktivnosti, ki se povsem razlikuje od diagrama, ki ga izdelamo s programskim orodjem Planner. Osredotočimo se na standard UML, na podlagi katerega deluje tudi Dia. Prikažemo proces oskrbe z izbrano komponento v podjetju (proces naročanja, dostave, izmenjave dokumentov, grobega in finega prevzema platišč, reklamacij in skladiščenja).

Z Dia imamo možnost izrisa najrazličnejših oblik tehničnih diagramov. Je uporabno programsko orodje za vsakega študenta, ki se pri študiju in izpopolnjevanju sreča s tovrstno problematiko. Prednost je v delovanju na različnih operacijskih sistemih, neplačljivost, lahka dostopnost, enostavnost učenja in prožnost.

Poglavje 4

ZINT– generator črtnih kod



Tehnologija črtne kode
Sledljivost
Primer: generiranje črtne kode

4.1 Teoretično ozadje

4.1.1 Tehnologija črtne kode

Obvladovanje oskrbne verige in zagotovitev sledljivosti komponent na vhodni strani sistema in izdelkov na izhodni strani sistema, predvsem v procesu transporta in skladiščenja, zahteva poznavanje in uporabo standardov označevanja in identifikacije logističnih enot, kot so surovina, izdelek, paket, paleta, kontejner ipd. Označevanje je temelj učinkovite sledljivosti logističnih enot v celotni oskrbni verigi oz. logistiki.

Sledljivost zagotavlja:

- zmanjšanje stroškov poslovanja;
- nadzor nad tokovi blaga, informacijami in stroški;
- izmenjava informacij v realnem času;
- upravljanje oskrbne verige;
- zmanjšanje tveganja;
- hitro ukrepanje in odločanje;
- učinkovito planiranje procesov.

Za doseg sledljivosti blaga je potrebno upoštevati različne vrste standardov, opredeliti procese dela in zagotoviti informacijsko podporo, tako programsko kot mehansko. Pod pojmom programska informacijska podpora razumemo uporabo različnih programskih rešitev in orodij, kot so npr. poslovni informacijski sistem (ERP oz. PIS) za vodenje celotnega poslovnega sistema (naročila, računi, zaloge itd.), skladiščni informacijski sistem (WMS oz. SIS) za vodenje skladiščnega sistema oz. funkcije podjetja, računalniška izmenjava podatkov (EDI oz. RIP) za izmenjavo podatkov med sistemi, sistemi za upravljanje transportnih sistemov (TMS), programi za vodenje logističnih procesov. Pod pojmom mehanska oprema razumemo uporabo informacijske tehnologije, kot so razni čitalniki črtnih ali RFID kod, tiskalniki, računalniki, strežniki itd. [79].

Potrebno je upoštevati naslednje zakonitosti [99]:

- kako se bodo podatki zajemali;
- kdaj se bodo podatki zajemali;
- kdo jih bo zajemal;

- kdo jih bo koristil;
- sistem sledenja;
- označevanje;
- začetek/konec označevanja.

Velik poslovni interes za sledenje in določanje položaja blaga se je pojavil v zadnjem desetletju. Struktura za razvoj sledenja in določanja položaja je zasnovana v treh plasteh dobavne verige: kodiranje (fizična plast); informacijska arhitektura (informacijska plast); načrtovanje ter kontrola (nadzorna plast). Z znanstvenega vidika takšna struktura zagotavlja usklajeno in sistematično organizacijo sledenja [93].

Notranja sledljivost je vzpostavljena, ko partner pri sledljivosti dobi enega ali več sledljivih predmetov, za katere se izvajajo notranji postopki, preden ta predmet ali več predmetov odda naprej. Notranji postopek obsega eno ali več faz, ki jih izvaja isti subjekt in ne zahtevajo bistvenega sodelovanja drugih trgovskih partnerjev. Notranji postopek mora biti sestavljen iz najmanj ene od štirih faz. To so premik, predelava, hramba in uničenje [19].

Elementi sledenja blaga [99]: blago; tehnologija označevanja; informacijska tehnologija (strojna oprema); informacijski sistem (programska oprema); organizacijska struktura zaposlenih; sistem sledenja in odločanja; pravila in standardi sledenja.

Ključni principi sledljivosti [99]: identifikacija blaga in logističnih enot; beleženje vseh zaporednih povezav med proizvodnimi serijami in logističnimi enotami; beleženje podatkov za sledljivost preko celotne oskrbovalne verige; zagotovitev vseh potrebnih podatkov za sledljivost za naslednjega partnerja.

Sledljiv predmet je fizični predmet, pri katerem lahko obstaja potreba po zajemu informacij o njegovi zgodovini, uporabi ali lokaciji. Stopnja, na kateri je sledljivi predmet določen znotraj embalaže ali logistične hierarhije, je odvisna od panoge in potrebne stopnje nadzora. Povzeto po [63] [18].

Sledljiv predmet od najvišje do najnižje ravni je lahko [19]: pošiljka; logistična enota; artikel.

Vsi sledljivi predmeti morajo biti označeni z globalno edinstveno identifikacijo, ki se mora nahajati neposredno na sledljivem predmetu. Če to ni mogoče, se mora nahajati vsaj na sredstvu, v katerem se nahaja, ali na spremnem dokumentu.

Možnosti označevanja blaga [79]: RFID koda; črna koda; ročno označevanje. Značilnosti črne kode [79]:

- razširjenost uporabe;

- globalnost standardov;
- enostavnost označevanja;
- odpravlja človeške napake;
- omogoča hitrejši in natančnejši zajem podatkov;
- širok izbor strojne opreme.

Uvedba črtne kode oz. označevanja blaga omogoča optimizacijo časa procesov v oskrbni verigi in zmanjševanje tveganja. V današnjem času je mnogo izdelkov že označenih. V kolikor označeni izdelki nastopajo v funkciji surovine, označevanje izdelkov brez integracije partnerjev v oskrbni verigi ne zagotavlja sledljivosti blaga v celotni oskrbni verigi. Tako imamo urejen le vhodni del sistema zajema podatkov, vsekakor pa ne smemo zanemariti izhodni del sistema, kjer odpremljamo končne izdelke. Le te je potrebno prav tako označevati in identificirati, da zagotovimo še nadaljnjo sledljivost blaga.

V našem primeru je pomembno identificiranje in označevanje komisionov (pakiranih logističnih enot), ki so pripravljene na nadaljnjo odpremo v proizvodnjo. Označevanje komisionov in njegova integracija z celotnim sistemom označevanja in zajemanja podatkov omogoča in zagotavlja sledljivost blaga od njegovega izvora do njegovega ponora.

Označevanje komisionov ima največji pomen prav za podjetje, saj zagotavlja notranjo sledljivost komponent v notranjih logističnih procesih. Pri tem je potrebno poznavanje tehnologije črtne kode (vrste, zajem podatkov, uporaba tehnologije). Na svetu obstaja 142 različnih vrst črtne kode, med katerimi so najpogosteje uporabljeni črtni kodi tipa GS1-128 (EAN-128) in GS1-13 (EAN-13). Uporaba vrste črtne kode je odvisna od izdelka oz. enote, ki jo označujemo. Tako na primer črtno kodo tipa Data Matrix uporabljamo v proizvodnji, črtno kodo tipa Code Pharmacy 128 v farmaciji, črtno kodo tipa Code 39 pri označevanju arhiva itd. Poznavanje tehnologije črtne kode omogoča "ureditev" materialnega poslovanja v podjetju in natančnejšo opredelitev poslovnih procesov.

Uvajanje črtne kode v poslovanje zahteva:

- planiranje, izvedbo, kontrolo in uporabo;
- nakup informacijske tehnologije;
- pilotni preizkus;
- vzdrževanje.

Uvedba črtne kode zahteva od podjetja določene investicijske stroške, ki lahko ob nekaterih negativnih dejavnikih krepko narastejo. V izogib je potrebno izvesti pilotne uvedbe oz. preizkusna obdobja, da se prepričamo o smiselnosti uvedbe črtne kode. Za začetek je primerna uporaba prosto dostopne programske opreme oz. odprtokodnih programov, ki ponujajo specifične rešitve pri uvedbi črtne kode.

Odprtokodni program Zint omogoča kodiranje, generiranje in tiskanje 142 različnih vrst črtne kode celotnega sveta. Njegova baza z različnimi vrstami črtnih kod se časovno dopolnjuje in izpopolnjuje. Programsko orodje Zint je generator črtnih kod, ki podjetjem oz. uporabniku omogoča pripravo in uvedbo črtnih kod, katere so berljive z ustrezno informacijsko tehnologijo (čitalniki črtne kode) [68].

Upoštevanje različnih standardov podpira ustreznost in standardiziranost črtnih kod [68]:

- BS EN 797:1996;
- BS EN 798:1996;
- BS ISO/IEC 12323:2005;
- BS ISO/IEC 15417:2007;
- BS ISO/IEC 15438:2006;
- BS ISO/IEC 16022:2006;
- ISO/IEC 24778:2008;
- ANSI/HIBC 2.3-2009;
- GS1.

4.2 O programskem orodju

Programsko orodje Zint je odprtokodni program za generiranje črtne kode. Uporabniku omogoča izdelavo različnih črtnih kod, ki jih lahko vgradimo v dokumente ali HTML strani. Črtne kode je možno vključiti v funkcionalnosti druge programske opreme. Namen programa je zagotoviti rešitev, ki je dovolj prožna za poklicne uporabnike in hkrati omogoča enostaven prevod vhodnih podatkov za izdelavo črtne kode [68].



Slika 4.1: Uvodna spletna stran

Prenos in namestitvev

Programsko orodje Zint enostavno prenesemo s spletnega naslova ponudnika programskega orodja Zint. Deluje na platformah Windows Linux (Ubuntu) in MAC. Prenos programskega orodja Zint se razlikuje glede na uporabniška okolja. Preprosto ga prenesemo s klikom na ikono na levi strani uvodne strani (Slika 4.1). Izberemo lahko tudi spletno verzijo generatorja črtne kode. Po zaključku prenosa programsko orodje še namestimo. Dodatna navodila za pomoč pri prenosu in namestitvi so dostopna na spletni strani.

Problem

V podjetju želimo generirati črtno kodo, ki bo omogočala identifikacijo komisiov na izhodni oz. odpremni strani skladiščnega sistema. Interno označevanje komisiov je potrebno zaradi učinkovitejše sledljivosti v oskrbni verigi, med skladiščem in proizvodnjo. Za izbiro označevanja komisiov oz. pripravljenih paketov za proizvodnjo linijo (paket oz. komisio sestavlja 4 pnevmatike, 4 platišča in 16 vijakov) je najprimernejša standardna koda GS1-128 (EAN-13 ali UPC-A ali ITF-14 ali UCC/EAN-128). Za označevanje izberemo črtno kodo tipa ITF-14, ki je primerna za označevanje paketov z vsebino. Simbologija GTIN-14 uporablja simbologijo črtne kode ITF-14 [94]. Obroba črtne kode omogoča izenačitev pritiska tiskarske plošče (printanje črtne kode) po celotni površini simbola in zagotavlja zanesljivost odčitavanja (zmanjšanje verjetnosti napačne razlage simbologije). Označevanje pripravljenih paketov izvedemo na koncu komisioiranja, ko bodo paketi pripravljeni za odpremo.

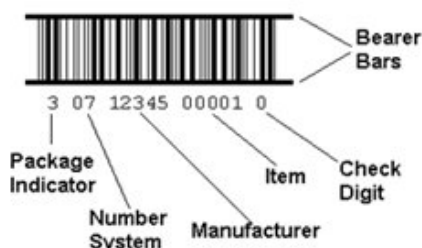
4.3 Uporaba

Z Zint programskim orodjem na preprost način generiramo različne vrste črtnih kod. Začetna stran ponuja:

- izbiro vrste črtne kode, katero želimo generirati (142 različnih vrst);
- prikaz generirane črtne kode;
- dodatne možnosti pri generiranju črtne kode (oblika, velikost itd.);
- vnos teksta;
- resetiranje teksta;
- možnost generiranja zaporedja črtne kode;
- shranitev dela;
- dodatne informacije o programski opreми;
- izhod.

3	Indikator pakiranja (tip paketa)
07	Številka sistema
12345	Proizvajalec
00001	Referenca enote
0	Kontrolna številka

Tabela 4.1: Razlaga simbologije črtne kode ITF-14



Slika 4.2: Simbologija črtne kode ITF-14.

Vir: [32]

Generiranje črtne kode

Z Zint generiranjem črtnih kod lahko že vnaprej pripravimo seznam črtnih kod, ki bodo uporabljene za označevanje. Povezava Zint z namenskim tiskalnikom črtnih kod omogoča tiskanje črtnih kod. Priporočljivo je voditi evidenco že uporabljenih črtnih kod (povezava s skladiščnim in poslovnim informacijskim sistemom), da zagotovimo popolno sledljivost. Vsa dodatna označevanja v procesih oskrbne verige zahtevajo celovito informacijsko-logistično podporo. Izberemo in uporabimo simbologijo črtne kode ITF-14 (Slika 4.2).

Oblikovanje komisijonov se izvrši na podlagi izdajnice oz. pakirnega lista, ki ga pripravijo v projektnem delu za planiranje in vodenje proizvodnje. Vsak komisijon pridobi svojo identifikacijsko številko, ki ga spremlja na poti od skladišča do proizvodnje. Identifikacijska številka vsebuje podatke o vsebini komisijona, številko šarže in proizvajalca (delovno mesto in ime oblikovalca komisijona). Vsako izdajnico oz. navodilo za oblikovanje komisijona spremlja identifikacijska številka, ki jo pretvorimo v del črtne kode. Tako sledimo hkrati dokumentaciji in blagu.

S številko črtne kode identificiramo dokument (številka izdajnice oz.

Sestavljena črtna koda	50712345000014
Indikator pakiranja	5
Številka sistema	07
Proizvajalec ali delovno mesto	12345
Številka šarže ali številka dokumenta	00001
Kontrolna številka	4

Tabela 4.2: Primer črtne kode

Simbol	Učinek
#	Vstavi prosto mesto
\$	Vstavi ničlo
*	Vstavi zvezdico
Ostali simboli	Vstavi, kar želite

Tabela 4.3: Ukazi za generiranje zaporedja črtne kode

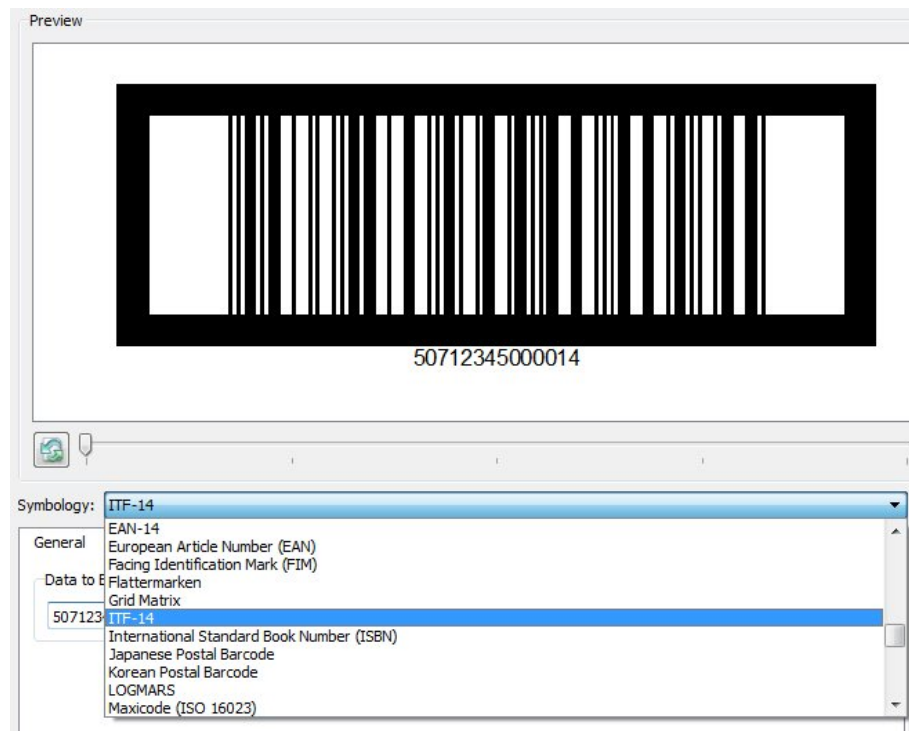
pakirnega lista), delovno mesto (kdo je oblikoval komisijon), šaržo komisijona (številka paketa, vrstni red). Za celovito podporo sledljivosti blaga potrebujemo še preostalo tehnologijo (čitalnik, informacijski sistem). Za potrebe sledenja naših komisijonov v *Symbology* izbiramo med različnimi vrstami črtnih kod. Izberemo črtno kodo ITF-14 (Slika 4.3).

Za potrebe slednja komisijonov zadostuje že črtna koda s številko **50712345000014**. Številka je kreirana na podlagi naših zahtev oz. postavk, ki so že predhodno razložene (Slika 4.4).

V razdelku *Appearance* izbiramo velikost črtne kode, skalo tiskanja, tip obrobe in barvo črtne kode (Slika 4.5 in 4.6). Velikost in barva črtne kode sta zmeraj odvisna od velikosti oz. barve podlage paketa.

Za generiranje številč črtnih kod uporabimo razdelek *Sequence*. Pri generiranju zaporedja števila črtne kode določimo interval (začetno in končno vrednost). Primer: če želimo kreirati 10 različnih črtnih kod, nastavimo začetno vrednost (*Start Value*) na 1 in končno (*End Value*) na 10. Potrebno je določiti še razmerje zaporedja oz. stopnjo povečanja (*Increment By*) in formata izpisa (*Format*). Stopnjo povečanja nastavimo na 1.

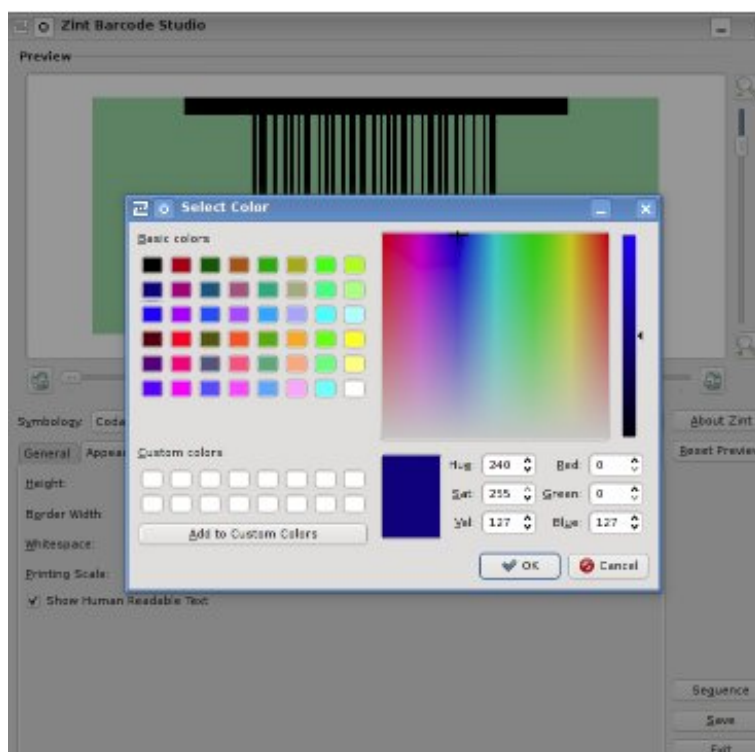
Z ukazom *Create* generiramo zaporedje številč črtnih kod, katere določimo z našimi parametri na Sliki 4.7. Le te s klikom na razdelek *Export* izvozimo v samostojno mapo ali jih pripravimo za tisk. S klikom na *Reset* poenostavimo vse nastavitve generiranja številč črtnih kod na začetno stanje. Na izbiro so različne možnosti shranjevanja datotek. Datoteke (generirane številke črtnih



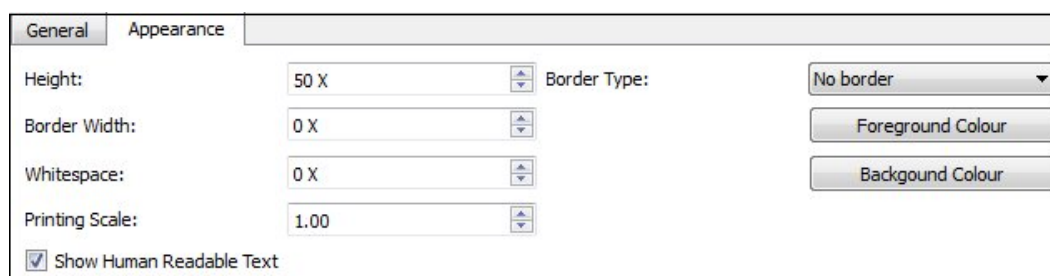
Slika 4.3: Izbira črtne kode ITF-14



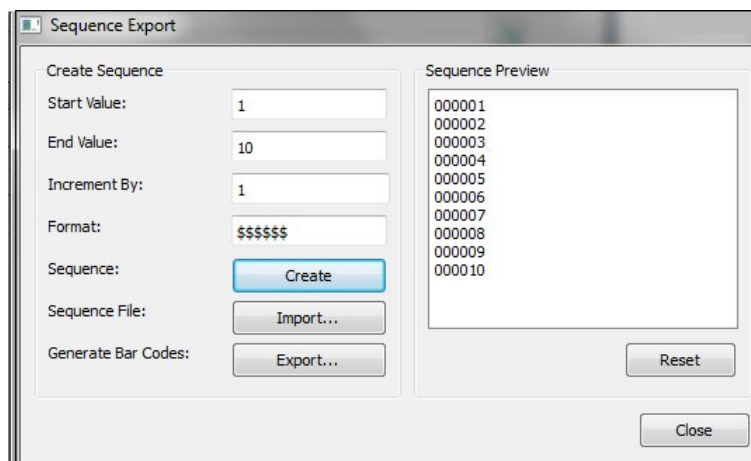
Slika 4.4: Kreirana črna koda



Slika 4.5: Izbira barvne podlage



Slika 4.6: Izbira velikosti črtne kode



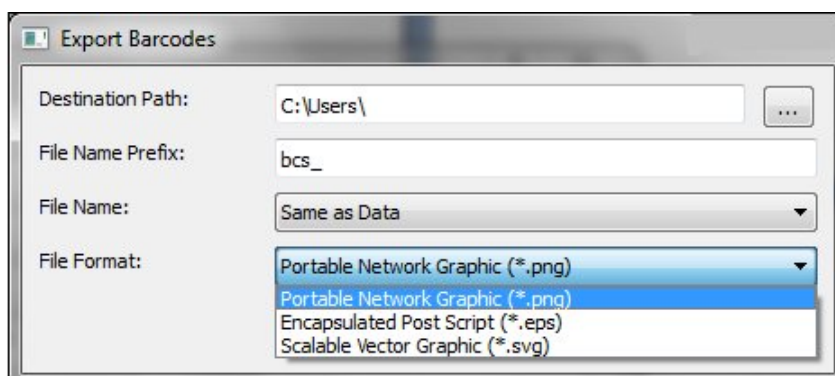
Slika 4.7: Generiranje zaporedja

kod) shranimo (priporočljivo) kot ime datoteke ali serijske številke in določimo tip dokumenta, kjer so razlike v izbiri grafičnih možnosti (.jpeg, .eps, .svg) (Slika 4.8).

Generirane številke črtnih kod izvozimo v novo mapo (Slika 4.9). Kasneje jih z integracijo ustreznega tiskalnika poljubno natisnemo in nalepimo na oblikovane komisione.

Uporabnost programa Zint se izkaže pri implementaciji tehnologije črtne kode v poslovanje podjetja. Uvedba tehnologije črtne kode je dolgotrajen postopek, ki zahteva poznavanje tehnologije in standardov črtne kode. Rezultat uspešne implementacije tehnologije črtne kode v luči učinkovitega sledenja vhodnih in izhodnih tokov v poslovnem procesu dolgoročno povrne vse finančne investicije v obliki zmanjšanja stroškov poslovanja (napake, izgube, tveganja) in optimizacije delovanje celotne oskrbne verige.

Implementacija tehnologije črtne kode zahteva še določena strokovna znanja in veščine, katere deloma ponuja programsko orodje Zint. Programsko orodje za generiranje različnih črtnih kod seznanja uporabnika z vrsto različnih kod, metodologijo generiranja črtne kode, možnostjo tiskanja črtnih kod itd.



Slika 4.8: Izbira oblike dokumenta shranitve generiranega zaporedja

The screenshot shows a file explorer window with the following table of files:

Name	Size	Type	Date Modified
bcs_00000...	537 bytes	PNG image	Mon 07 Jun 2010 09:46:52 AM CEST
bcs_00000...	558 bytes	PNG image	Mon 07 Jun 2010 09:46:52 AM CEST
bcs_00000...	551 bytes	PNG image	Mon 07 Jun 2010 09:46:52 AM CEST
bcs_00000...	561 bytes	PNG image	Mon 07 Jun 2010 09:46:52 AM CEST
bcs_00000...	546 bytes	PNG image	Mon 07 Jun 2010 09:46:52 AM CEST
bcs_00000...	549 bytes	PNG image	Mon 07 Jun 2010 09:46:52 AM CEST
bcs_00000...	563 bytes	PNG image	Mon 07 Jun 2010 09:46:52 AM CEST
bcs_00000...	554 bytes	PNG image	Mon 07 Jun 2010 09:46:52 AM CEST
bcs_00000...	561 bytes	PNG image	Mon 07 Jun 2010 09:46:52 AM CEST
bcs_00001...	573 bytes	PNG image	Mon 07 Jun 2010 09:46:52 AM CEST

Slika 4.9: Slike generiranega zaporedja črtne kode

Povzetek

Z odprtokodnim programom Zint prikažemo generiranje različnih vrst črtnih kod. V danem primeru izberemo črtno kodo simbologije ITF-14, ki je med najprimernejšimi za interno označevanje oblikovanih komisionov. S programom prikažemo:

- različne vrste simbologij črtnih kod;
- uporabnost črtne kode;
- generiranje črtne kode;
- generiranje poljubnega števila črtnih kod (generiranje zaporedja);
- shranitev generiranih črtnih kod (različne oblike dokumenta);
- strokovna razlaga simbologije črtne kode ITF-14.

Programsko orodje Zint je primerno orodje za začetek uvajanja črtne kode v poslovne procese. Nudi široko paleto različnih znanj, ki so potrebna pri izbiri in implementaciji tehnologije črtne kode.

Poglavje 5

ASDN - integracija oskrbne verige



Oblikovanje logističnih mrež
Oskrbna veriga

Primer: potek dobave pnevmatik od dobavitelja do podjetja Open Storage

5.1 Teoretično ozadje

5.1.1 Oskrbna veriga

V preteklosti se je pojem oskrbne verige nanašal na tok materiala od dobaviteljev (virov) do določenega proizvodnega podjetja ter naprej skozi proizvodni proces tega podjetja, nato pa so se končni izdelki dostavljali kupcem. V sedanjem času je poimenovanje oskrbne verige veliko širše. Le ta obsega tok materiala, informacij, plačil in storitev, ki potekajo od dobaviteljev surovin, skozi tovarne in skladišča do končnih uporabnikov. V oskrbno verigo so vključene organizacije in procesi, ki ustvarjajo in dobavljajo proizvode, informacije in storitve končnim uporabnikom. Zajema številne naloge, kot npr. nabava, tok plačil, rokovanje z blagom, načrtovanje in nadzor proizvodnje, logistika, upravljanje z zalogami, distribucija in dostava. Cilj modernega menagementa oskrbne verige je zmanjšanje negotovosti in tveganj v verigi, s čimer pozitivno vpliva na raven zalog, čas proizvodnega cikla, poslovne procese in servisiranje odjemalcev. Vse te koristi prispevajo k izboljšanju konkurenčnosti in profitabilnosti. Organiziranje logističnih procesov v oskrbni verigi je kompleksno, tako imenovani determiniran projekt s posrednim ekonomskim učinkom, ki horizontalno in vertikalno povezuje in vpliva na več podjetij oz. organizacij [88].

Kadar govorimo o oskrbni verigi hkrati zajemamo delovanje in ustvarjanje logističnih mrež in omrežij. Le ta morajo biti na stroškovno dostopni ravni celovito in kvalitetno povezana. Pri tem je pomembno sodelovanje strokovnjakov danega področja, ki s skupnimi močmi ustvarijo ugodno logistično omrežje, za uporabnika na eni in drugi strani. Pogosto se srečamo s pojmom "agility" - opredelimo ga kot zmožnost delovanja v negotovosti, medtem ko se ohranja stabilna raven produktivnosti. Negotovost se nanaša na mešano proizvodnjo in tehnološke spremembe. Do težav dandanes prihaja zaradi številnih dejavnikov, zato je z izdelavo logističnih mrež potrebno predvideti potrebe in povpraševanje ter zmanjšati negotovost. Management zmogljivosti predstavlja način za izboljšanje obdobj in koristi hitrega odziva. S tradicionalnimi računovodskimi metodami ne moremo vedno upravičiti vseh stvari, saj lahko slabša izkoriščenost zmogljivosti pripomore k povečanju stroškov na enoto. Zato se je oblikovalo veliko število najrazličnejših rešitev, ki so oblikovane za industrijska podjetja in se soočajo s problemom hitro premikajočega blaga za potrošnike.

5.2 O programskem orodju

ASDN Logistics Analysis (v nadaljevanju ASDN) je programska oprema za oblikovanje industrijskih logističnih mrež, ki prispevajo k izboljšanju industrijskih omrežij z optimizacijo zalog. Na enostaven način oblikujemo industrijsko logistično mrežo, za uporabnika prijazen način. Je programska oprema za hitro modeliranje oskrbnih verig, ki uporabniku omogoča analizo industrijskih omrežij, skozi različne scenarije in različne zorne kote. Izračun funkcionalnosti sistema uporabnikom omogoča, da opredelijo pravice na ravni zalog za izpolnitev obveznosti do strank ob hkratnem zmanjšanju stroškov. Izračuni so prikazani v naprej določenih sredstvih odvisnosti.

ASDN je prosto dostopno programsko orodje, ki omogoča izgradnjo potrebnih logističnih mrež, izdelavo grafov, izračun velikosti serije modelov ipd. S predhodnim znanjem o programskem orodju, lahko glede na naše potrebe, enostavneje razporejamo in spreminjamo vse modele znotraj logistične mreže. ASDN programsko orodje je bilo razvito na Univerzi v Vaasi na Finskem, prvotno za ABB korporacijo za analizo velikih globalnih omrežij med ponudbo in povpraševanjem. V ASDN modelu vsako vozlišče (kot je dobavitelj, proizvajalec ali distribucijsko skladišče) vsebuje določene lastnosti (zaloge, predhodni čas ...).

Programska oprema ASDN omogoča:

- izgradnjo industrijsko logističnih omrežij;
- ogled grafov zgrajenega omrežja;
- pregled poročil;
- vpis podatkov in izračun rezultatov z različnimi vzorci velikosti serij itd.

Prenos in namestitev

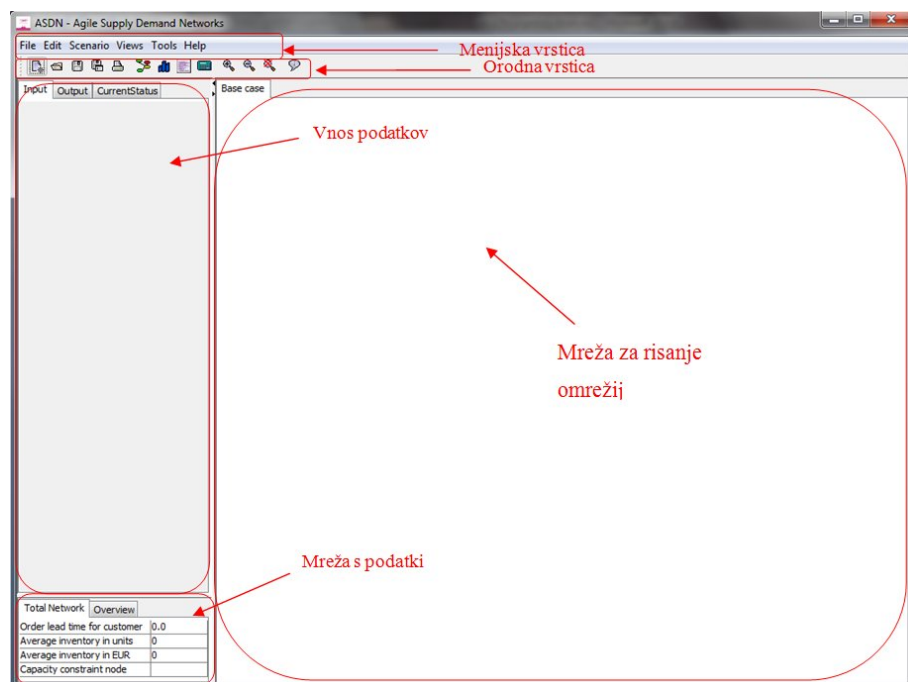
Programsko orodje ASDN enostavno prenesemo s spletne strani ponudnika programskega orodja ASDN [4]. Deluje na vseh platformah (Windows Linux (Ubuntu), MAC itd.).

Prvi korak (pri uporabi ASDN) je ustvariti industrijsko logistično mrežo v glavnem oknu. Na Sliki 5.2 prikažemo osnovno okno. Kot privzeta oblika se na vrhu nahaja naslovna vrstica. Nato sledi osnovni meni in orodna vrstica, s katerima je zagotovljen dostop do posameznih orodij.

Meni *File* omogoča dostop do operacij z oblikovanjem novih datotek (*New*), odpiranjem že shranjenih datotek (*Open*), shranjevanjem (*Save*, *Save as*), tiskanjem (*Print*) itd. V meniju *Edit* je možnost dodajanja vozlišč in puščic.



Slika 5.1: Agile Supply Demand Network



Slika 5.2: Osnovno okno ASDN

Datoteko shranimo na želeni prostor tako, da izberemo *Save as*. V primeru, kadar želimo odpreti že shranjeno datoteko, najprej odpremo program ASDN in nato znotraj odpremo želeno datoteko.

Meni *Scenario* omogoča dodajanje ali spreminjanje scenarijev. ASDN ponuja možnost oblikovanja več različnih scenarijev, na podlagi katerih primerjamo dane rezultate. V meniju *View* si lahko ogledamo mreže, grafe ipd. Sledita menija *Orodja (Tools)* in menija *Pomoč (Help)*, kjer je zapisano nekaj splošnih besed o programu, dodan pa je tudi priročnik za uporabo programa (*Tutorial*).

Osnovne definicije:

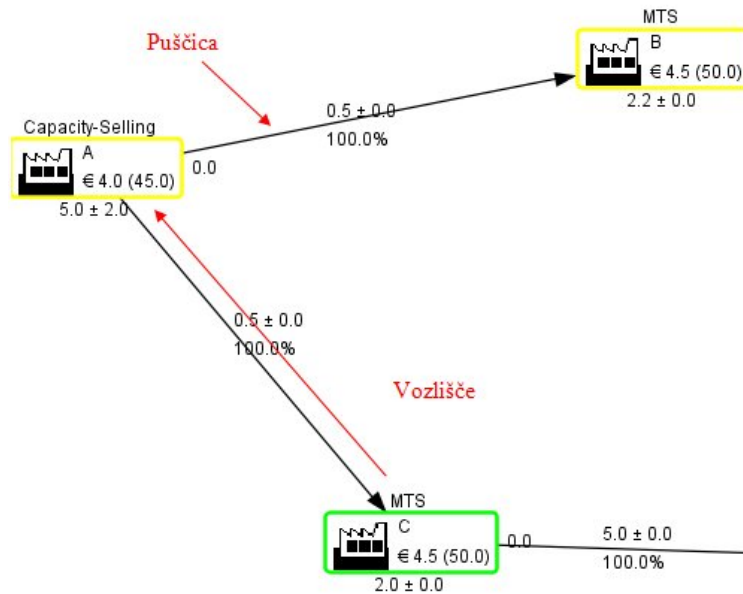
- Vozlišče (*Node*) je osnovna enota omrežja, ki predstavlja družbo;
- Puščica (*Arrow*) predstavlja del omrežja - razmerje prenosa blaga od dobavitelja do naročnika;
- Mreža (*Network*) predstavlja kombinacijo vozlišč in puščic - predstavlja točno določeno razporeditev med dobaviteljem in stranko, torej razmerja med podjetji;
- Scenarij (*Scenario*) omogoča določeno razporeditev med dobaviteljem in stranko ter odnose med podjetji;
- Datoteka (*File*) je dokument, ki lahko vsebuje več scenarijev.

V ASDN moramo zapisati posamezne vrednosti atributov, ki jih delimo na vhodne (*Input*) in izhodne (*Output*). Atributi so lahko razporejeni bodisi v skupinah na strukturiran ali nestrukturiran način. Da lahko določimo vrednosti atributov kliknemo na posamezno vozlišče na mreži. Prikažejo se dani atributi, katere nato poljubno spreminjamo in oblikujemo. Atributi za vozlišča so prikazani v Tabeli 5.1.

Iz Slike 5.4 je razvidno, da točko nevezanosti ETO ("Inženir-naročilo") označimo v primeru, ko se odločimo, da bomo oblikovali komponento, torej proces oblikovanja. MTO ("Naročilo") uporabimo v primeru, ko je že vzpostavljen nadzor pretoka materiala in je potrebno izvesti naročilo, na podlagi pridobljenih materialov oz. surovin. ATO ("Skupinska izvedba") uporabimo, kadar se npr. proizvodnja že izvede. Gre torej za delo na procesu. MTS ("Na zalogo") uporabimo takrat, ko je izvedena končna komponenta in le to ustrezno uskladiščimo do nadaljnjega.

Tabela 5.2 prikazuje attribute za vozlišče, ki jih izberemo za vsako posamezno vozlišče posebej.

S Sliko 5.6 in Tabelo 5.3 prikažemo orodja *Orodne vrstice*.



Slika 5.3: Vozlišča in povezave med njimi

Parameter	Uporaba
ID	Zaporedna oznaka vozlišča, ki je ni mogoče urejati.
Label	Ime vozlišča.
Tip vozlišča	Izbiramo lahko med petimi različnimi vrstami vozlišč: predelovalna dejavnost (izdelava), inženiring, skladišče, h končnim uporabnikom, distributer, trgovina na debelo, trgovina na drobno itd.
Točka nevezanosti (Order Decoupling Point)	Prikažemo jo lahko s pomočjo Slike 5.4. Glede na fazo v kateri se nahajamo izberemo ustrezno možnost v proizvodnem okolju, katere so: ETO – Engineer-to-Order; MTO – Make-to-Order; ATO – Assembly-to-Order; MTS – Make-to-Stock in Capacity – Selling.

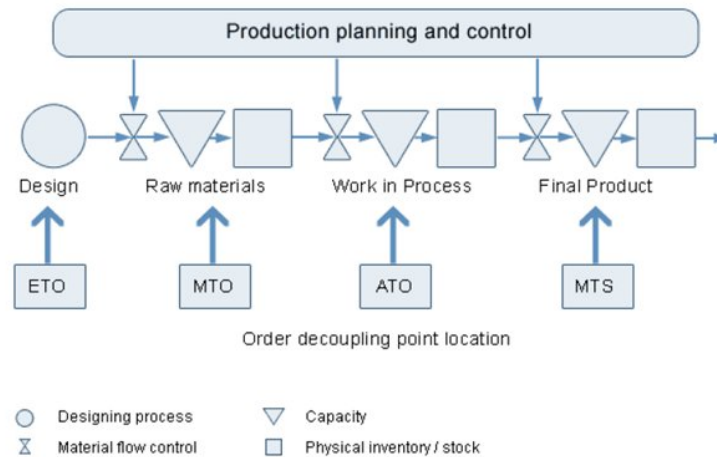
Tabela 5.1: Atributi za vozlišča

Parameter	Uporaba
Prodajna cena izdelka (<i>Price</i>)	Uporablja se za izračun dodane vrednosti (dodana vrednost = cena - stroški).
Proizvodni stroški (<i>Cost</i>)	Stroški izdelave izdelka. Uporabljajo se za izračun dodane vrednosti.
Potreba (<i>Demand</i>)	/
Čas proizvodnje (<i>Production Throughput time - TPT</i>)	Čas, ki se porabi od začetka do konca proizvodnje.
Standardni odklon OLT (<i>St.deviation of OLT</i>)	Odklon OLT obsega naročila, sodne zaostanke, čas inženiringa ipd. V primeru izbrane možnosti <i>Make-to-Stock</i> (MTS) se standardni odklon nanaša na čas proizvodnje (TPT). V primeru <i>Assembly-to-Order</i> (ATO) se standardni odklon nanaša na čas predelave TPT in ladijski čas. V primeru <i>Engineer-to-Order</i> (ETO) se standardni odklon nanaša na čas inženiringa in dobaviteljev za obdelavo TPT in ladijski promet.

Tabela 5.2: Drugi atributi

Številka	Pomen oznake
1	ustvari novo mapo
2	odpri mapo
3	shrani mapo/shrani mapo kot. . .
4	natisni posamezni scenarij ali graf
5	preklopi na mrežni pogled
6	preklopi na Ganttov diagram
7	preklopi v pogled poročila
8	preklopi na velikost serije kalkulatorja
9	povečaj/pomanjšaj ali ponastavi pogled
10	info o programski opremi

Tabela 5.3: Pomen oznak orodne vrstice



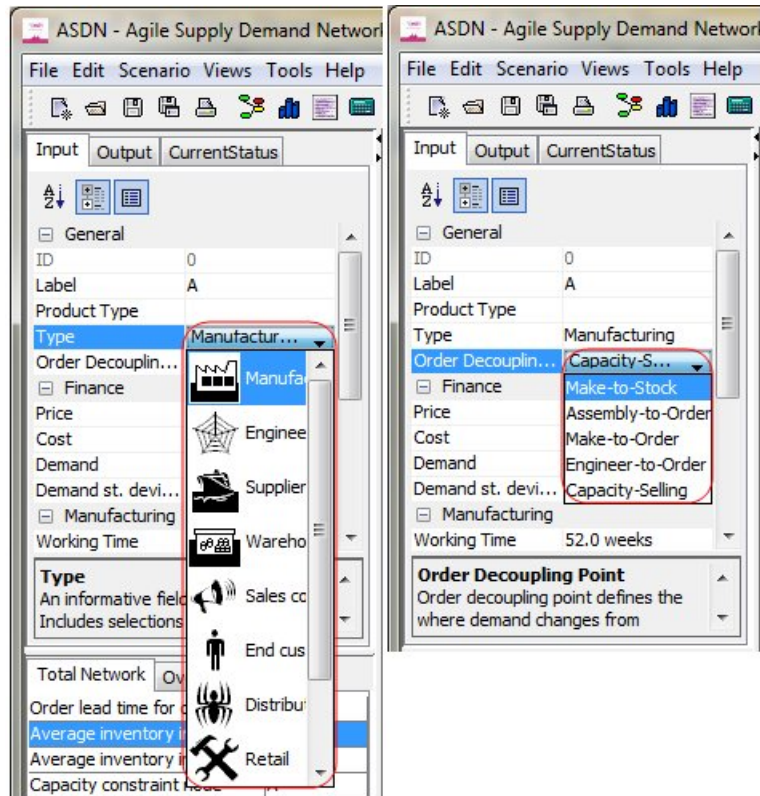
Slika 5.4: Izpolnitev procesa

Vir: [85]

Problem

V podjetju OpenStorage imamo problem dostave komponent s strani dobaviteljev. Če želimo ustvarjati maksimalne dobičke in minimalne izgube je v prvi vrsti potrebno poskrbeti za celovito oskrbno verigo.

S pomočjo programskega orodja ASDN zgradimo avtomobilsko industrijsko omrežje in podrobneje prikažemo potek dobave pnevmatik od dobavitelja v Franciji do našega podjetja v Bohinjski Bistrici. Hkrati poteka logistična mreža od dobavitelja preko distributerja, do podjetja, ki komponente prodaja ne da bi jih pri tem vgrajevali na končni izdelek. Z ASDN programskim orodjem prikažemo potek delovanja oskrbne verige, z vključitvijo predvidenih stroškov, ki v procesu nastanejo. Vsekakor lahko z danim orodjem zastavimo povsem drugačen potek logistične mreže, kar je odvisno od problema, ki ga obravnavamo. Menimo, da je pomembno poznati problematiko dostave komponent tudi iz drugega vidika, torej v obratni smeri, kar omogoča nadzor nad celotnim procesom dostave.



Slika 5.5: Atributi za vozlišča



Slika 5.6: Orodna vrstica

5.3 Uporaba

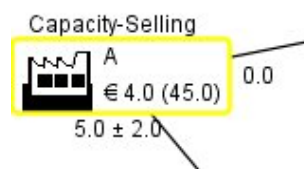
V nadaljevanju na kratko opišemo, kako ustvariti in analizirati posamezne logistične mreže. V danem primeru zgradimo avtomobilsko industrijsko omrežje.

Na osnovno okno dodamo šest ključnih vozlišč, katere v nadaljevanju ustrezno identificiramo. Prvo vozlišče je dobavitelj, ki prihaja iz Francije. Ta odpremi pnevmatike do našega podjetja, katere dostavimo v skladišče, kjer so uskladiščene dokler se ne namestijo na končni izdelek. Hkrati, vzporedno s to aktivnostjo prikažemo še, kako je pnevmatika dobavljena drugemu podjetju, ki se preko distributerjev posreduje do prodajnega podjetja, ne da bi se izvedlo nameščanje na končni izdelek.

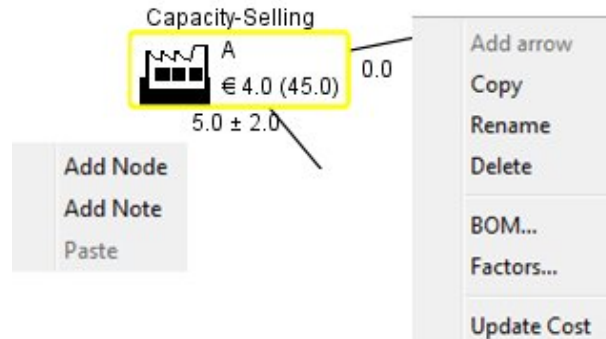
Postopek izdelave logistične mreže pričnemo z odprtjem nove datoteke (*New*), dodamo vozlišča in puščice ter sestavimo mrežo. Za dodajanje vozlišč na mrežo, se z desnim klikom na miško, prikaže možnost *Add Node*, kar pomeni dodajo vozlišča. Pri dodajanju puščic se postavimo na izbrano lokacijo, z desnim klikom na miško iberemo možnost *Add Arove* in povezava je narejena (glej Sliko 5.8).

V nadaljevanju je prikazan del izdelane logistične mreže na konkretnem primeru dostave pnevmatik iz tujine (glej Sliko 5.9).

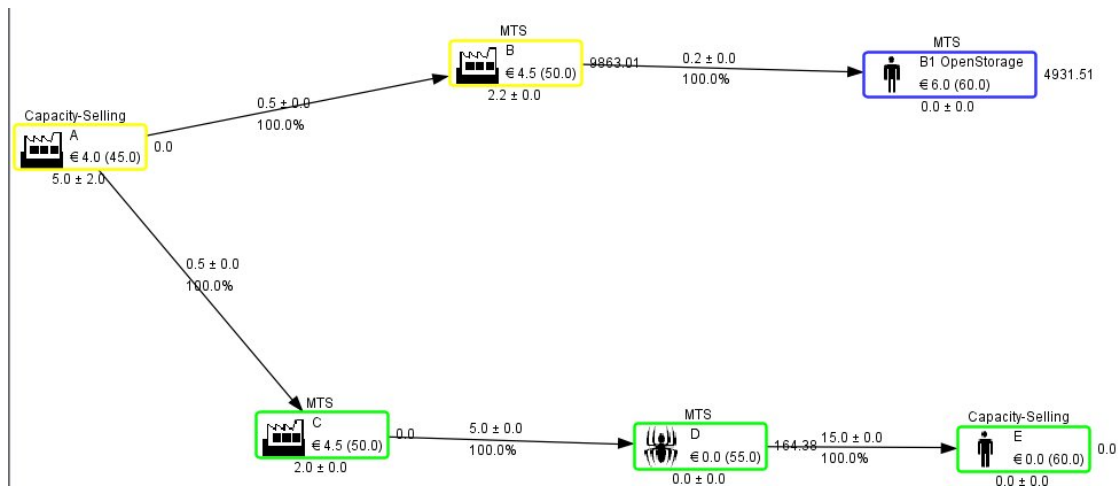
Najprej izpolnimo attribute za kategorijo A (dobavitelj). Enak postopek izvedemo še za določitev ostalih atributov na vozliščih. Potrebno je vstaviti številne podatke, saj le ti omogočajo pravilen izračun stroškov, količin ipd. Določitev vrednosti atributov je po našem mnenju najtežje delo, saj je potrebno natančno izračunati vrednosti za vsako posamezno vozlišče. Za poenostavitev je v ASDN tabela s katere uporabnik prebere širšo definicijo posameznega atributa in razlago določitve vrednosti atributov (glej Sliki 5.10 in 5.11). Pripravi in zapisu atributov je potrebno nameniti največ časa - smiselno je sodelovanje ostalih sodelavcev v podjetju in poslovnih partnerjev. V primeru, ko je logistična mreža sestavljena zgolj iz vozlišč v okviru podjetja, je delo enostavnejše, že zaradi hitre pridobitve podatkov in odziva vseh sodelujočih. Po zapisu atributov, je uveljavljen celovit sistem, ki se že s spremembo ene vrednosti spremeni v vseh



Slika 5.7: Uporaba programa ASDN



Slika 5.8: Dodajanje povezav in pušic



Slika 5.9: Dodajanje povezav in pušic

2.2	Average inventory in units	Average inventory in units. Cycle stock (2.4) + Safety outbound stock (2.6) + Safety inbound stock (2.5)	units
2.3	Average inventory in euros	Average inventory in euros. Inbound inventory value (2.8) + Outbound inventory value (2.9) + Cycle stock (2.4) * Price (1.4) + Holding cost rate (1.13) * Price (1.4)	euros
2.4	Cycle stock	Amount of units required for the reorder time. Manufacturing stage time (1.8) * Demand (1.6) / 2	units
2.5	Safety inbound stock	Safety stock kept for the uncertainty from supply Demand (1.6) * (100 – previous node's Service level (1.12))	units
2.6	Safety outbound stock	Safety stock kept for the uncertainty from demand Manufacturing stage time (1.8) * Demand standard deviation (1.7) * NORMSINV(Service level) NORMSINV (P) returns the inverse of the standard normal cumulative distribution, where P is probability corresponding to the normal distribution. Normal distribution is standard when mean = 0 and standard deviation = 1. NORMSINV(P) returns value X, such that $P = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} dt$	units

Slika 5.10: Izračun zahtevnejših vrednosti – 1. del

ostalnih fazah.

Na primeru prikažemo logistično mrežo in okviren zapis atributov z izmišljenimi vrednostmi (glej Sliki 5.12 in 5.13).

Ko določimo vse attribute vozlišč, enako storimo še za posamezne puščice. Pričnemo pri povezavi med točko A in B, torej od dobavitelja do našega podjetja. S klikom na puščico se odprejo atributi, ki jih ustrezno označimo. Najprej izberemo prevozno sredstvo s katerim bomo transportirali komponento. Na voljo imamo sledeče možnosti: *Ground* (cestni prevoz), *Air* (zračni prevoz), *Rail* (železniški prevoz), *Ship* (ladijski prevoz) in *Other* (druge vrste prevoza) (glej Sliko 5.14).

V nadaljevanju dopolnimo vrednosti atributov z danimi podatki, ki so na voljo, ali pa jih sami izračunamo (cena, stroški, čas transporta...). Za lažje razumevanje in oblikovanje mreže se s klikom na posamezen atribut izpiše njegov pomen (glej Sliko 5.15).

Ko določimo vsa vozlišča in povezave lahko na omrežje dodamo zemljevid. Izbiramo med zemljevidom Evrope, Sveta ali ostalih, ki jih zasledimo na seznam oz. so shranjeni v lastni datoteki (glej Sliki 5.16 in 5.17).

		In our case NORMSINV (Service level) returns number of units in the stock that is up to Service level.	
2.7	Inventory turn rate	Demand per year divided by average inventory in units $\text{Demand (1.6)} * 365 / \text{Average inventory in units (2.2)}$	days
2.8	Inbound inventory value	Inbound inventory value. $\text{Price (1.4)} * \text{Average inventory in units (2.2)}$	euros
2.9	Outbound inventory value	Outbound inventory value. $\text{Price (1.4)} * \text{Safety outbound stock (2.6)}$	euros
<i>TOC (theory of constraints)</i>			
2.10	Capacity utilization	Capacity utilization $\text{Demand (1.6)} / \text{Capacity (1.10)}$	percent
2.11	Throughput Dollar Days	Throughput Dollar Days $(\text{Price (1.4)} - \text{Cost (1.5)}) / \text{Manufacturing stage time (1.8)}$	
2.12	Inventory Dollar Days	Inventory Dollar Days $\text{Average inventory in units (2.2)} / \text{Manufacturing stage time (1.8)}$	

Slika 5.11: Izračun zahtevnejših vrednosti – 2. del

The image displays three screenshots of the ASDN - Agile Supply Demand Network software interface, showing the 'Input' tab for different nodes. Each screenshot shows a list of attributes and their values, along with a summary table at the bottom.

Node 0 (Label A):

General	ID	0
General	Label	A
General	Product Type	
General	Type	Manufacturing
General	Order Decoupling ...	Capacity-Selling
Finance	Price	45.0 €
Finance	Cost	4.0 €
Finance	Demand	735000.0 units / y...
Finance	Demand st. devia...	0.0 units / year
Manufacturing	Working Time	52.0 weeks
Manufacturing	Production Throu...	10.0 days
Manufacturing	St.deviation of OLT	2.0 days
Manufacturing	Order backlog	5.0 days
Manufacturing	Shipping time	12.0 days
Manufacturing	Engineering lead-t...	2.0 days
Manufacturing	Capacity	2770.0 units / day
Manufacturing	Maximum order fu...	50.0 days
Manufacturing	OTD	90.0 %
Inventory	Holding cost rate	20.0 %

Node 1 (Label B):

General	ID	1
General	Label	B
General	Product Type	
General	Type	Manufacturing
General	Order Decoupling ...	Make-to-Stock
Finance	Price	50.0 €
Finance	Cost	4.5 €
Finance	Demand	720000.0 units / y...
Finance	Demand st. devia...	0.0 units / year
Manufacturing	Working Time	52.0 weeks
Manufacturing	Production Throu...	0.0 days
Manufacturing	St.deviation of OLT	0.0 days
Manufacturing	Order backlog	10.0 days
Manufacturing	Shipping time	2.0 days
Manufacturing	Engineering lead-t...	0.0 days
Manufacturing	Capacity	720000.0 units / day
Manufacturing	Maximum order fu...	0.0 days
Manufacturing	OTD	90.0 %
Inventory	Holding cost rate	20.0 %

Node 2 (Label B1 OpenStorage):

General	ID	2
General	Label	B1 OpenStorage
General	Product Type	
General	Type	End customer
General	Order Decoupling ...	Make-to-Stock
Finance	Price	60.0 €
Finance	Cost	6.0 €
Finance	Demand	720000.0 units / y...
Finance	Demand st. devia...	0.0 units / year
Manufacturing	Working Time	52.0 weeks
Manufacturing	Production Throu...	0.0 days
Manufacturing	St.deviation of OLT	0.0 days
Manufacturing	Order backlog	0.0 days
Manufacturing	Shipping time	0.0 days
Manufacturing	Engineering lead-t...	0.0 days
Manufacturing	Capacity	0.0 units / day
Manufacturing	Maximum order fu...	0.0 days
Manufacturing	OTD	90.0 %
Inventory	Holding cost rate	20.0 %

Summary Table (Total Network Overview):

Order lead time for customer	69.0
Average inventory in units	14959
Average inventory in EUR	523870
Capacity constraint node	A

Slika 5.12: Zapis vrednosti atributov – 1. del

The image shows three screenshots of the ASDN - Agile Supply Demand Network software interface, each displaying the configuration for a different node (ID 3, 4, and 5). The interface is divided into several sections: a menu bar (File, Edit, Scenario, Views, Tools, Help), a toolbar, and a main area with tabs for Input, Output, and CurrentStatus. The main area contains a tree view of attributes and a summary table at the bottom.

Node 3 (ID 3, Label C, Product Type Manufacturing):

- Price: 50.0 €
- Cost: 4.5 €
- Demand: 15000.0 units / year
- Working Time: 52.0 weeks
- Production Throughput: 0.0 days
- St.deviation of OLT: 0.0 days
- Order backlog: 0.0 days
- Shipping time: 2.0 days
- Engineering lead-time: 0.0 days
- Capacity: 0.0 units / day
- Maximum order fulfillment: 0.0 days
- OTD: 99.99 %
- Holding cost rate: 20.0 %

Node 4 (ID 4, Label D, Product Type Distributor):

- Price: 55.0 €
- Cost: 0.0 €
- Demand: 15000.0 units / year
- Working Time: 52.0 weeks
- Production Throughput: 1.0 days
- St.deviation of OLT: 0.0 days
- Order backlog: 0.0 days
- Shipping time: 0.0 days
- Engineering lead-time: 0.0 days
- Capacity: 0.0 units / day
- Maximum order fulfillment: 0.0 days
- OTD: 90.0 %
- Holding cost rate: 20.0 %

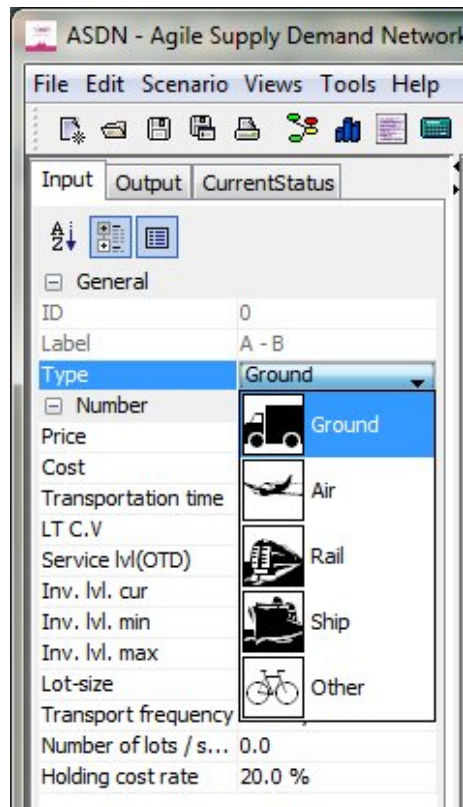
Node 5 (ID 5, Label E, Product Type End customer):

- Price: 60.0 €
- Cost: 0.0 €
- Demand: 15000.0 units / year
- Working Time: 52.0 weeks
- Production Throughput: 1.0 days
- St.deviation of OLT: 0.0 days
- Order backlog: 0.0 days
- Shipping time: 0.0 days
- Engineering lead-time: 0.0 days
- Capacity: 15000.0 units / day
- Maximum order fulfillment: 0.0 days
- OTD: 90.0 %
- Holding cost rate: 20.0 %

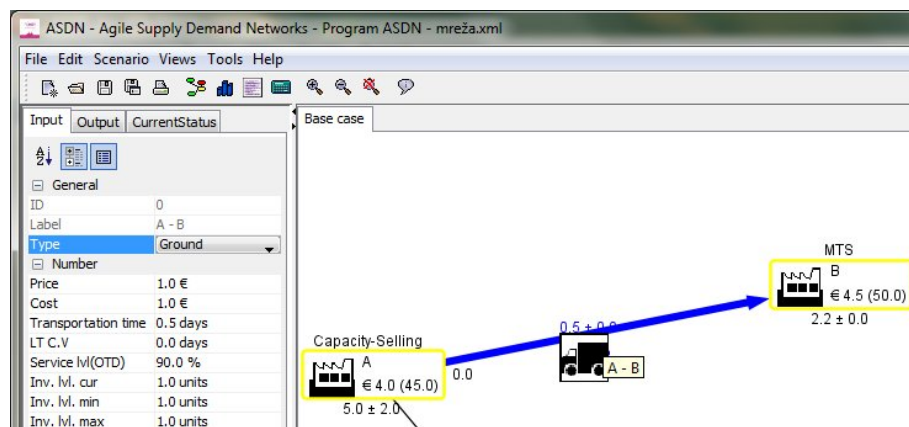
Summary Table (Total Network Overview):

Attribute	Value
Order lead time for customer	69.0
Average inventory in units	14959
Average inventory in EUR	523870
Capacity constraint node	A

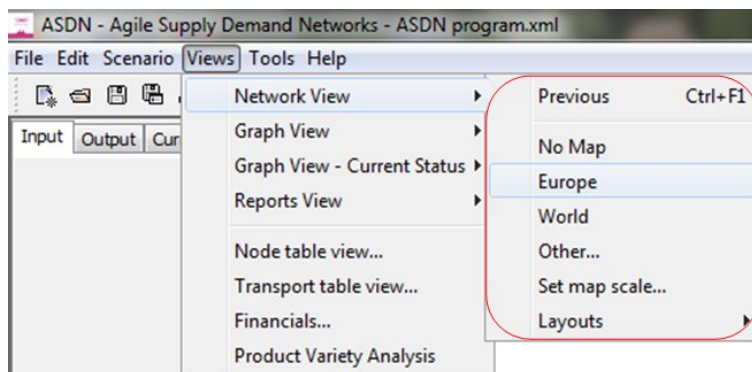
Slika 5.13: Zapis vrednosti atributov - 2. del



Slika 5.14: Izbira prevoza



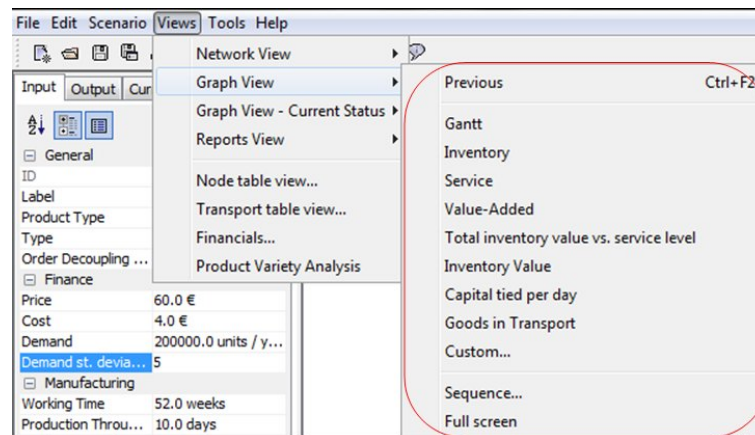
Slika 5.15: Atributi povezav



Slika 5.16: Izbira zemljevida



Slika 5.17: Prikaz na zemljevidu Evrope



Slika 5.18: Grafični prikaz rezultatov

Rezultati

Ko izdelamo logistično mrežo in določimo vse povezave sledi analiza podatkov. V meniju *Graph View* so na voljo možnosti s pomočjo katerih izvedemo celovito analizo našega dela. Izbiramo med izpisom Ganttovega diagrama (*Gantt*), zalogami (*Inventory*), vrednosti zalog (*Inventory Value*), blagom v transportu (*Goods in Transport*) ipd. (glej Sliko 5.18).

ASDN izriše Ganttov diagram (*Gantt*), ki prikazuje časovni potek celotne dobavne verige, glede na določena vozlišča (glej Sliko 5.19).

Slika 5.20 prikazuje graf zalog (*Inventory*) za posamezna vozlišča.

Iz Slike 5.21 je razviden grafičen prikaz vezanega kapitala na dan (*Capital tied per day*) za posamezna vozlišča.

V meniju *Graph View* možnost *Custom* omogoča prikaz grafov po meri. Na voljo so vsi atributi, ki jih uporabimo pri svojem delu ter izris najrazličnejših grafov. Le te lahko prikažemo v klasični ali 3D obliki (glej Sliki 5.22 in 5.23).

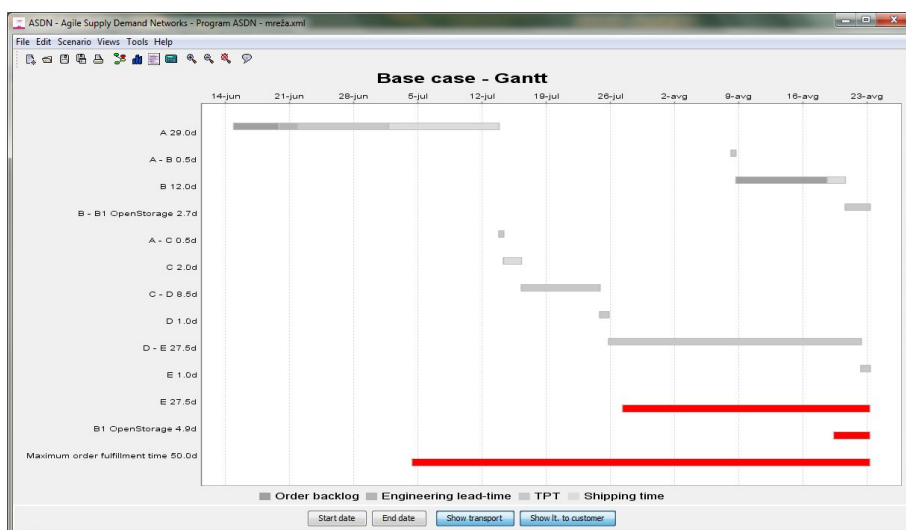
Slika 5.24 prikazuje raven storitev (*Total Inventory Value*).

Prikažemo zgolj nekaj izmed možnih grafičnih prikazov. Po ogledu grafičnih prikazov si v meniju *View* ogledamo skupno tabelo vozlišč (*Node table view*). Na njej so prikazana predhodno določena vozlišča z atributi, ki jih poljubno spreminjamo (glej Sliko 5.25).

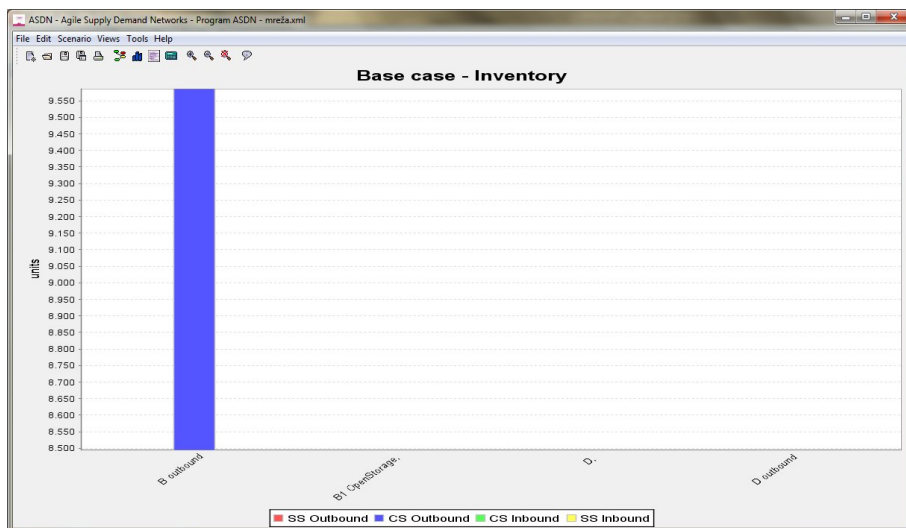
Prav tako si ogledamo še t.i. transportno tabelo (*Transport table view*), ki je kreirana glede na vozlišča in povezave med posameznimi kategorijami (glej Sliko 5.26).

Ogledamo si tudi finančne podatke (*Financials*) (glej Sliko 5.27).

Za preračun stroškov v meniju *Tools* izberemo možnost *Lot size calculator*,



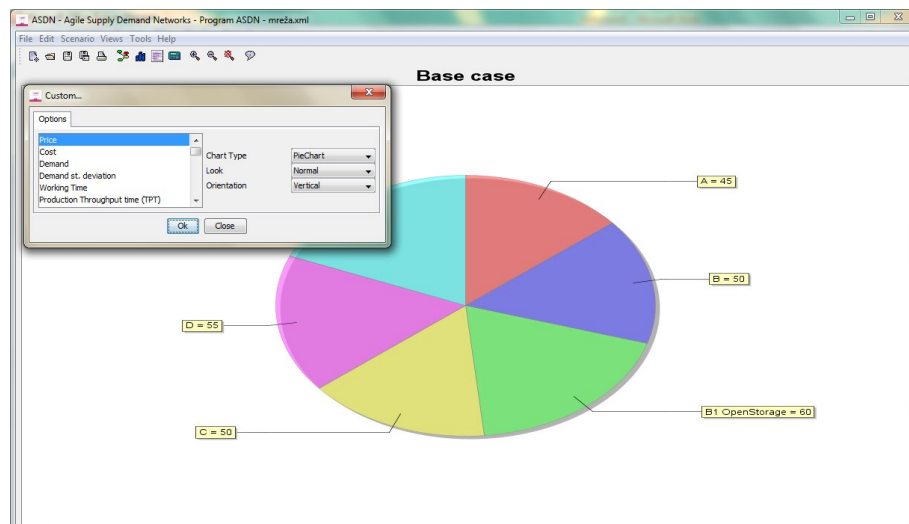
Slika 5.19: Ganttov diagram



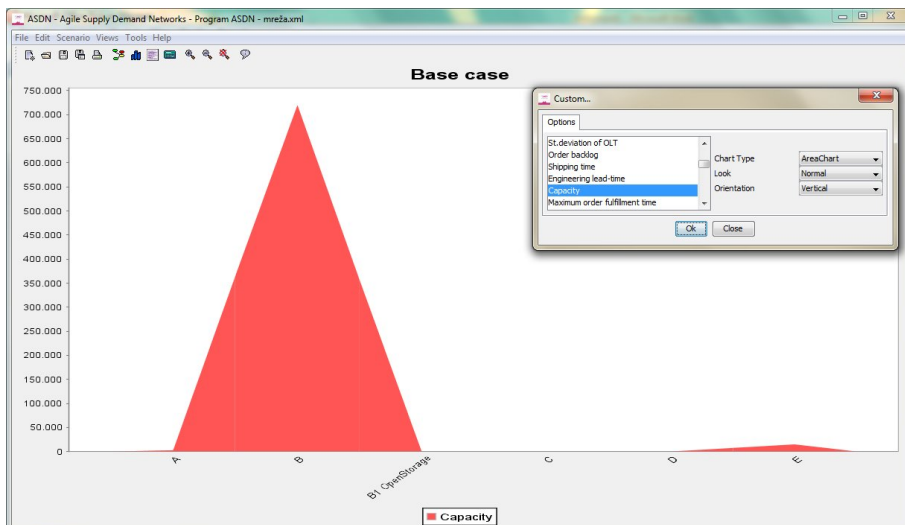
Slika 5.20: Zaloge



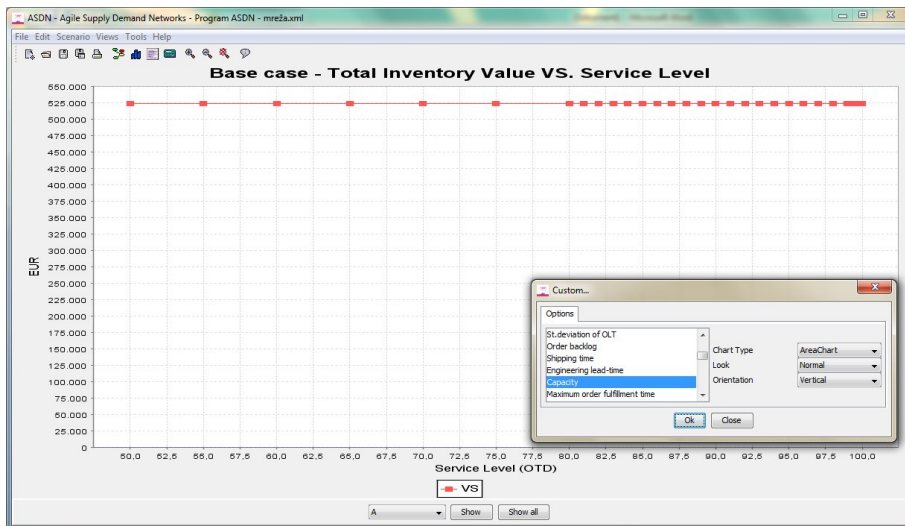
Slika 5.21: Vezani kapital na dan



Slika 5.22: Grafi po meri – 1. del



Slika 5.23: Grafi po meri – 2. del



Slika 5.24: Raven storitev

Node table view...

Input	Output					
	A	B	B1 OpenStorage	C	D	E
ID	0	1	2	3	4	5
Label	A	B	B1 OpenStorage	C	D	E
Product Type						
Type	Manufacturing	Manufacturing	End customer	Manufacturing	Distributor	End customer
Order Decoupling Point	Capacity-Selling	Make-to-Stock	Make-to-Stock	Make-to-Stock	Make-to-Stock	Capacity-Selling
Price	45.0	50.0	60.0	50.0	55.0	60.0
Cost	4.0	4.5	6.0	4.5	0.0	0.0
Demand	735000.0	720000.0	720000.0	15000.0	15000.0	15000.0
Demand st. deviation	0.0	0.0	0.0	0.0	0.0	0.0
Working Time	52.0	52.0	52.0	52.0	52.0	52.0
Production Throughput time (...)	10.0	0.0	0.0	0.0	1.0	1.0
St.deviation of OLT	2.0	0.0	0.0	0.0	0.0	0.0
Order backlog	5.0	10.0	0.0	0.0	0.0	0.0
Shipping time	12.0	2.0	0.0	2.0	0.0	0.0
Engineering lead-time	2.0	0.0	0.0	0.0	0.0	0.0
Capacity	2770.0	720000.0	0.0	0.0	0.0	15000.0
Maximum order fulfillment time	50.0	0.0	0.0	0.0	0.0	0.0
OTD	90.0	90.0	90.0	90.0	90.0	90.0
Holding cost rate	20.0	20.0	20.0	20.0	20.0	20.0

Ok

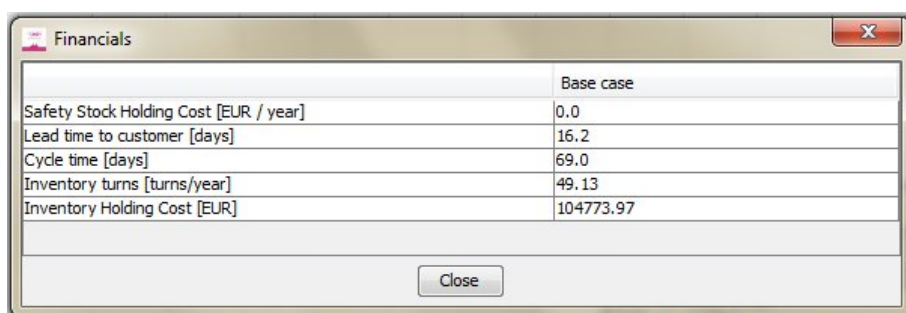
Slika 5.25: Tabela vozlišč

Transport table view...

Input	Output				
	New Edge	New Edge (1)	New Edge (2)	New Edge (3)	New Edge (4)
ID	0	1	2	3	4
Label	A - B	B - B1 OpenStorage	A - C	C - D	D - E
Type	Ground	Ground	Ground	Rail	Ship
Price	1.0	60.0	1.0	50.0	8000.0
Cost	1.0	1800.0	1.0	7200.0	1.0
Transportation time	0.5	0.20000000000000004	0.5	5.0	15.0
LT C.V	0.0	0.0	0.0	0.0	0.0
Service lvl(OTD)	90.0	90.0	90.0	90.0	90.0
Inv. lvl. cur	1.0	1.0	1.0	1.0	1.0
Inv. lvl. min	1.0	1.0	1.0	1.0	1.0
Inv. lvl. max	1.0	1.0	1.0	1.0	1.0
Lot-size	1.0	1.0	1.0	1.0	1.0
Transport frequency	0.0	5.0	0.0	7.0	25.0
Number of lots / shipment	0.0	0.0	0.0	0.0	0.0
Holding cost rate	20.0	20.0	20.0	20.0	20.0

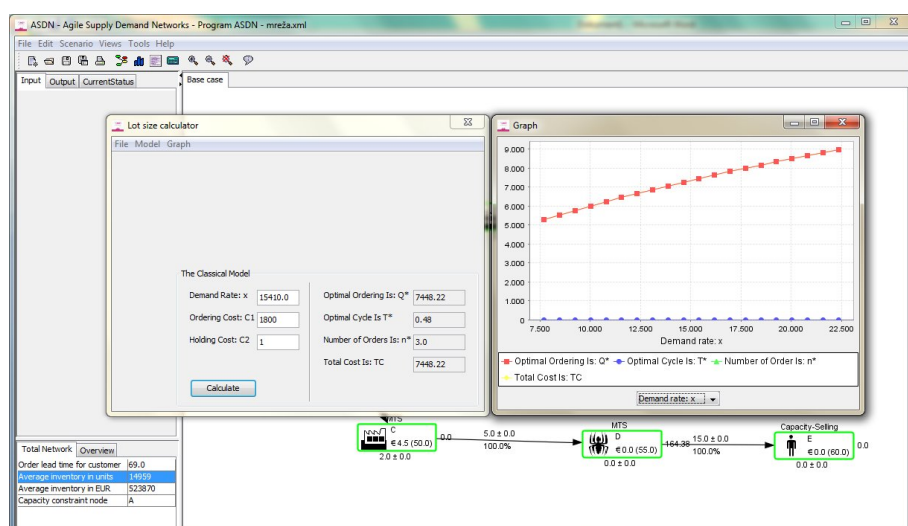
Ok

Slika 5.26: Transportna tabela



Base case	
Safety Stock Holding Cost [EUR / year]	0.0
Lead time to customer [days]	16.2
Cycle time [days]	69.0
Inventory turns [turns/year]	49.13
Inventory Holding Cost [EUR]	104773.97

Slika 5.27: Finančni podatki



Slika 5.28: Preračun

kje izbiramo med šestimi različnimi modeli. Če izberemo klasičen model (*The Classical Model*) vpišemo kakšne so potrebe (*Demand Rate x*), stroški naročanja (*Ordering Cost*) ter stroški skladiščenja (*Holding Cost*), nakar pritisnemo gumb *Calculate*. Na desni strani se izpiše optimalna naročilna količina (*Optimal Ordering ISs Q**), ki jo je smiselno naročiti; čas cikla (*Optimal Cycle Is T**), število ciklov (*Number of Orders Is*) ter izračun celotnih stroškov (*Total Cost Is:TC*). Za vsak izračun je na voljo tudi možnost grafičnega izrisa (glej Sliko 5.28).

V roadni vrstici *Report View*, si ogledamo posnetke našega dela - grafe, tabele ipd. (glej Sliko 5.29).

The screenshot shows a 'Reports View' window with a toolbar at the top. The main content area displays a table with three scenarios. Each scenario has a 'Total' value of 113777. The table is structured as follows:

Base case					
Standard case					
Transport	9003				
Inventory holding cost	104774				
Total :	113777				
Best case					
Transport	9003				
Inventory holding cost	104774				
Total :	113777				
Worst case					
Transport	9003				
Inventory holding cost	104774				
Total :	113777				

Slika 5.29: Pregled opravljenega dela - scenarij

Povzetek

ASDN programsko orodje uporabimo za oblikovanje logističnih mrež, ki prispevajo k izboljšanju industrijskih omrežij. Je programska oprema, ki uporabniku omogoča hitro modeliranje oskrbne verige in analizo industrijskih omrežij, skozi različne scenarije in iz različnih zornih kotov.

S programskim orodjem ASDN prikažemo potek dobave pnevmatik od dobavitelja do našega podjetja, kjer jih skladiščimo. Hkrati poteka logistična mreža od dobavitelja - distributerja, do prodajnega podjetja, ki kupi komponente (npr. pnevmatike), ne da bi jih pri tem vgradilo v druge komponente.

Pri opisu programskega orodja ASDN smo uporabili še dodatne vire in literaturo: [102] [85].

Poglavje 6

GOOGLE ZEMLJA - načrtovanje poti z digitalnim zemljevidom



Iskanje lokacij, računanje razdalj, izris poti
Planiranje in načrtovanje

Primer: načrtovanje poti od poslovnega partnerja v tujini do izbranega podjetja

6.1 Teoretično ozadje

6.1.1 Digitalni zemljevidi

Digitalni zemljevidi (v nadaljevanju zemljevidi) so prikaz sodobne tehnologije, s pomočjo katerih vizualiziramo pogled v najrazličnejša področja z najrazličnejšimi posnetki. V svetu poznamo številne inštitucije, ki se ukvarjajo z izdelavo digitalnih zemljevidov oz. interaktivnih globusov. Izpostavimo Google, Yahoo in Microsoft, ki so veliki konkurenti ne le na področju iskalnikov ampak tudi na področju zemljevidov. Prednosti zemljevidov so predvsem kvalitetni satelitski posnetki, posnetki zemeljskega površja visoke resolucije in dodatne storitve pregleda.

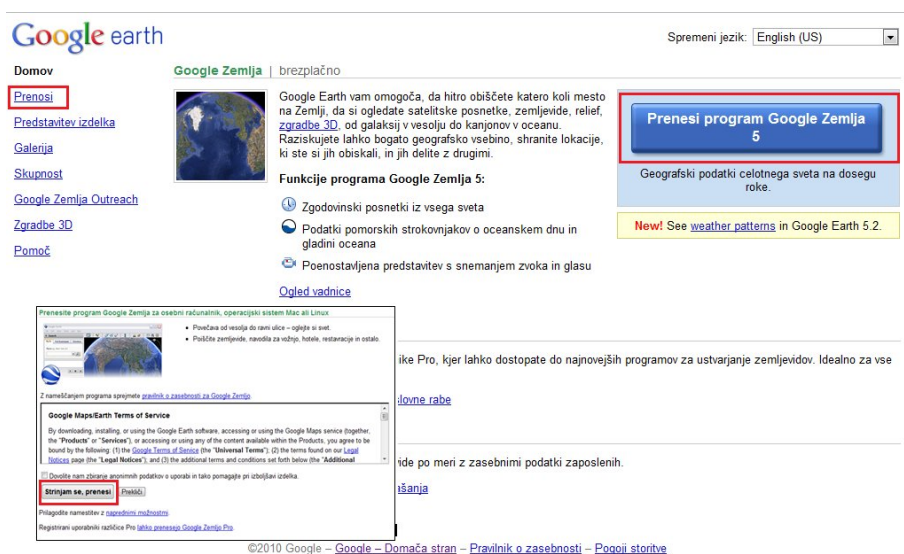
S področja spletne kartografije na preprost način izvemo, kdo se nahaja na določenem naslovu, pridobimo kontaktne podatke in ostale informacije, ki nas zanimajo. Načrtamo pot, po kateri bomo potovali iz ene na drugo lokacijo. Z izbranim orodjem se uporabnik na enostaven način premika po zemljevidu ter povečuje in zmanjšuje merilo prikazanega zemljevida. Eno izmed znanih orodij je programsko orodje Google Zemlja, katerega podrobneje opredelimo v nadaljevanju.

6.2 O programskem orodju

Google Zemlja (*Google Earth*) je programsko orodje s katerim postane naš računalnik okno v svet. Uporabljamo ga kot vizualizacijsko orodje, ki uporabniku omogoča kreiranje in vnos vhodnih podatkov [76]. Je prostodostopno in uporabniku prijazno, saj je priljubljeno na številnih področjih, tudi geografsko-znanstvenem [109].

Od leta 2005 naprej so raziskovalci Google Zemlja poiskali številne aplikacije, ki so jih vključili v že delujoče programsko orodje - npr. klimatske spremembe v prihodnosti, naravne nesreče, okoljski in prometni podatki ipd. Te aplikacije so uporabljale 2D geoprostorske in social-demokratske podatke za ustvarjanje virtualizacije v navideznem svetu. V sedanjem obdobju številna spletna vizualna orodja spreminjajo način prikaza podatkov s tako imenovanimi 3D geoprostorskimi podatki v navideznem okolju. Google Zemlja medsebojno povezuje satelitske, zemeljske, vodne in druge podatke z visoko ločljivostjo fotografij, dvignjenega reliefa, oznak za ceste in ulice, seznamov podjetij in še mnogo več. Uporabniki raziskujejo in izmenjujejo informacije o lokacijah in drugih subjektih, ki so geografsko locirani [69].

Google Zemlja predstavlja interaktivni globus našega planeta, ki ga ogledujemo v trirazsežni tehniki. Z navigacijskimi ikonami opazovano področje



Slika 6.1: Prenos programskega orodja Google Zemlja

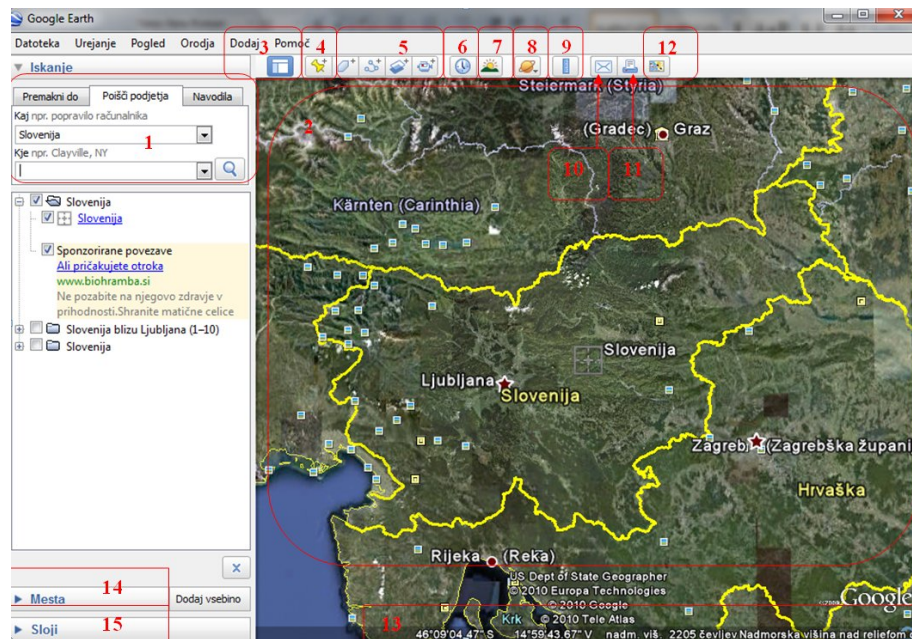
približujemo oz. oddaljujemo, pri podrobnejših posnetkih pa spremenjamo tudi kot ptičje perspektive. Gledanje satelitskih posnetkov zemeljskega površja pod kotom namesto navpično navzdol, daje možnost, da spoznamo konfiguracijo terena, ki ga opazujemo. Pri tovrstnem početju smo precej omejeni - vsako pretiravanje z naklonom popači prikaz, saj so slike posnete pod pravim kotom. Programsko orodje se razvija tudi v smeri realističnega prikazovanja okolja. V programu ustvarjamo tridimenzionalne objekte, ki jih na enostaven način prenesemo v Google Zemljo, kjer jih prilagodimo po velikosti in nastavimo na željeno lokacijo [37].

Prenos in namestitev

Uporabniki operacijskega sistema Windows si programsko orodje Google Zemlja namestijo s spletne strani Google earth [17] s klikom na ikono "Prenesi program Google Zemlja 5" ali s klikom na razdelek *Prenosi*, kjer potrdimo, da se strinjamo s pogoji uporabe (glej Sliko 6.1). Uporabniki operacijskega sistema Ubuntu pa enostavno poiščejo, prenesejo in namestijo programsko orodje preko *Programskega središča Ubuntu* oz. preko *Upravljanja paketov Synaptic*.

Programsko okno

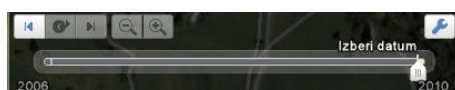
Podrobnejši pogled na programsko okno prikazuje Slika 6.2.



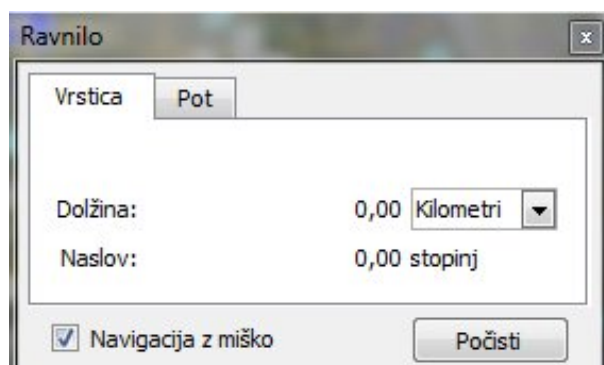
Slika 6.2: Ogljed programskega orodja

Pomen prikazanih orodij in ikon je razložen v nadaljevanju:

- Oznako *Plošča iskanja* uporabimo za iskanje mest in navodil za pot ter za upravljanje rezultatov iskanja.
- Oznako *Pregledni zemljevid* uporabimo za dodatni pogled Zemlje.
- Oznako *Pokaži/skrij stransko vrstico* uporabimo, kadar želimo skriti ali prikazati stransko vrstico.
- Oznako *Položaj* uporabimo kadar želimo označiti določeno lokacijo.
- Oznako *Mnogokotnik* uporabimo, kadar želimo dodati mnogokotnik; oznako pot, če želimo dodati pot.
- Oznako *Prosojnica s sliko*, če želimo na Zemljo dodati prosojnico s sliko in oznako posnemi potovanje, če želimo posneti pot.
- Oznaka *Zgodovinski posnetki* prikaže zgodovinsko lestvico, ki jo poljubno spreminjamo, glede na zemljevid, ki ga imamo odprtega (glej Sliko 6.3).
- Oznaka *Sonce* prikaže sončno svetlobo preko pokrajine, katero poljubno spreminjamo; s časovnim drsnikom nastavimo čas dneva.

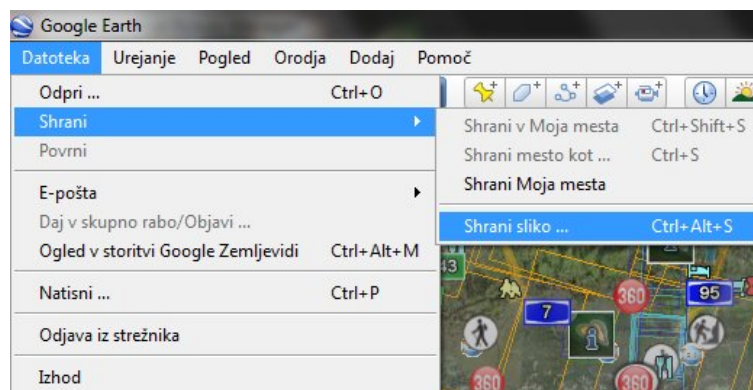


Slika 6.3: Nastavitev zgodovinskih posnetkov



Slika 6.4: Merjenje razdalj

- Oznaka *Nebo* omogoča ogled zvezd, ozvezdja, galaksij, planetov ali Zemljine lune.
- Oznaka *Merjenje* omogoča, da izmerimo razdaljo ali velikost območja (glej Sliko 6.4).
- Oznaka *E-pošta* omogoča pošiljanje pogleda ali slike preko e-pošte.
- Oznaka *Tiskanje* omogoča tiskanje trenutnega pogleda Zemlje.
- Oznaka *Ogled* omogoča pogled v Googlovih Zemljevidih v spletnem brskalniku.
- Oznaka *Vrstica stanja* prikazuje koordinate, nadmorsko višino, datum slike in status pretoka.
- Okno *Mesta* uporabimo za iskanje, shranjevanje, urejanje in ponovni obisk oznak položajev.
- Okno *Sloji* uporabimo za prikaz različnih zanimivih podatkovnih točk, na območju ogleda. Tukaj so vključene zanimivosti, zemljevid, ceste, relief in podatki o zgradbah. Celoten seznam slojev je na voljo na plošči *Sloji*.



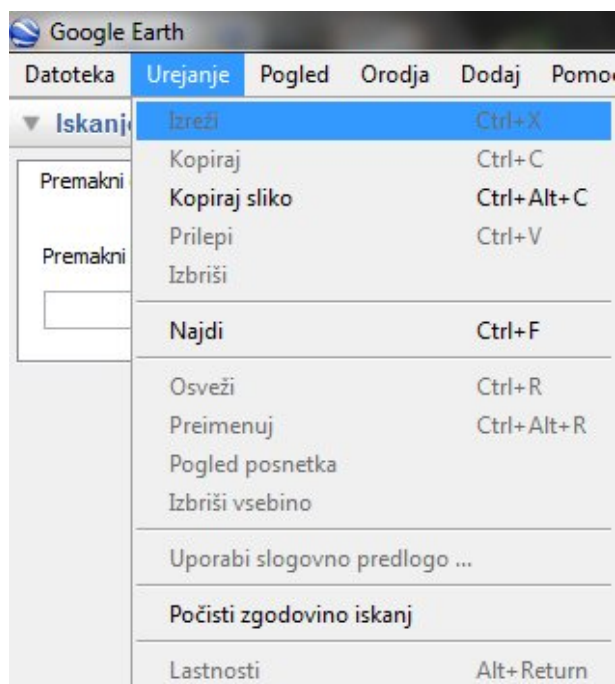
Slika 6.5: Orodna vrstica - Datoteka

Ko zaženemo programsko orodje, je najprej smiselno uporabiti datoteko, v katero shranimo dani dokument (*Shrani v Moja mesta*, *Shrani mesto kot...*, *Shrani Moja mesta*) oz. sliko (*Shrani sliko*). Ponuja možnost pošiljanja datoteke preko e-pošte, ogled v storitvi Google Zemljevidi (*Google Maps*), tiskanja datoteke, odjavo iz strežnika ali izhod iz programa (glej Slika 6.5).

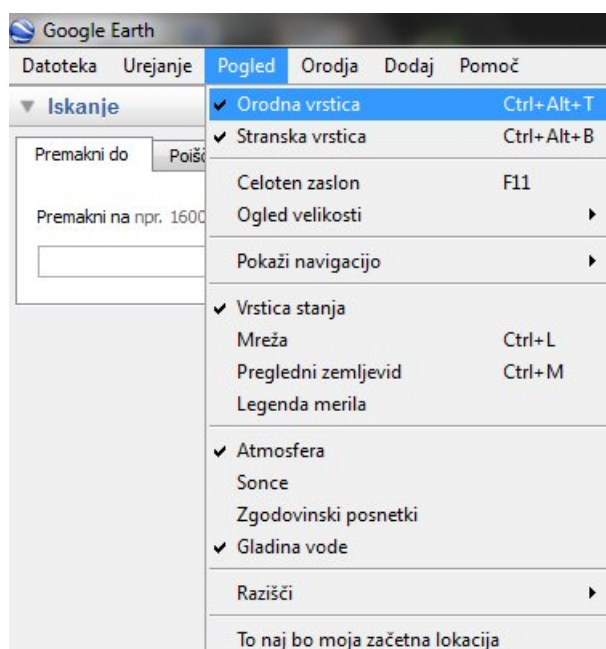
Orodna vrstica *Urejanje* omogoča kopiranje posameznih segmentov, slik, pogleda posnetka ipd. (glej Slika 6.6).

Orodna vrstica *Pogled* omogoča urejanje prikaza orodnih vrstic, stranskih vrstic, celotnega zaslona, mreže, preglednega zemljevida ipd. (glej Slika 6.7).

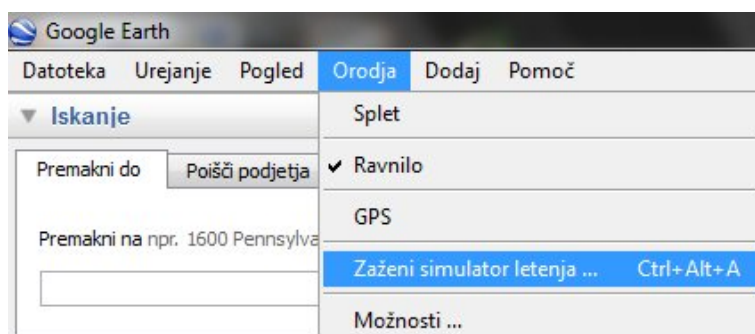
Orodna vrstica *Orodja* omogoča izbiro med različnimi možnostmi. Ena izmed možnosti je iskanje lokacij s pomočjo simulatorja letenja. Simulirano letalo upravljamo z miško ali s kakšnim drugim krmilnikom. Če želimo vzleteti, najprej pritisnimo tipko za premik po strani navzgor (*Page Up*), da povečamo potisk in premaknemo letalo naprej po vzletni stezi. Ko se letalo začne premikati, premaknemo miško ali kontrolno ročico nekoliko nazaj oz. navzdol. Ko dosežemo zadostno hitrost, vzletimo. Če želimo spremeniti smer ali nagniti letalo, počasi premikamo miško oz. kontrolno ročico. Pristajanje z letalom je nekoliko bolj zapleteno in zahteva nekaj vaje. Gre za eno izmed orodij s katerim lahko pogledamo svet v majhnem zaslonu. Simulator letenja vklopimo z izbiro možnosti *Zaženi simulator letenja* v razdelku *Orodja* (glej Sliki 6.8 in 6.9).



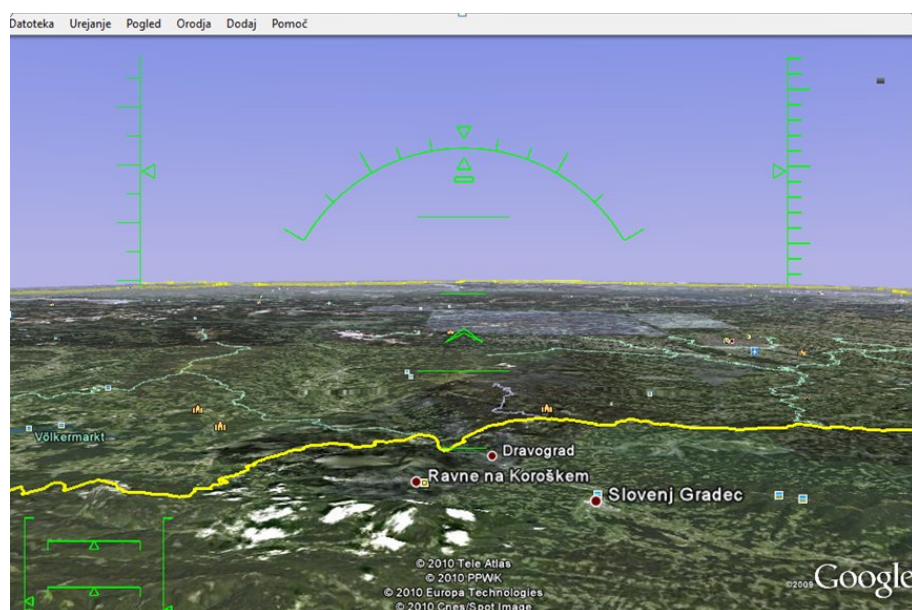
Slika 6.6: Orodna vrstica - Urejanje



Slika 6.7: Orodna vrstica - Pogled



Slika 6.8: Zagon simulatorja letenja



Slika 6.9: Simulator letenja

Problem

V podjetju OpenStorage se venomer srečujemo s številnimi poslovnimi potovanji in obiski parterjev, dobaviteljev. Pred vsakim pomembnim poslovnim potovanjem se je potrebno pripraviti na pot, kar omogočajo številna spletna orodja. V izbranem primeru se osredotočimo na problem iskanja lokacije našega podjetja, ki se nahaja v Bohinjski Bistrici na Triglavski cesti, v občini Bohinj. Tuji poslovni partnerji, ki sodelujejo z nami, naročajo in dobavljajo komponente, vendar po večini ne vedo, kje se nahajamo.

Da jim olajšamo pot, bomo s pomočjo programa Google Zemlja natančno preučili, kako nas bodo najhitreje našli. S praktičnim primerom prikazemo kaj programske orodje nudi, kako pridobiti natančna navodila za vožnjo iz enega mesta v drugo, hkrati pa pregledamo še druge zanimivosti bližnje lokacije in funkcij, ki so jih ustvarili razvijalci programa Google Zemlja.

6.3 Uporaba

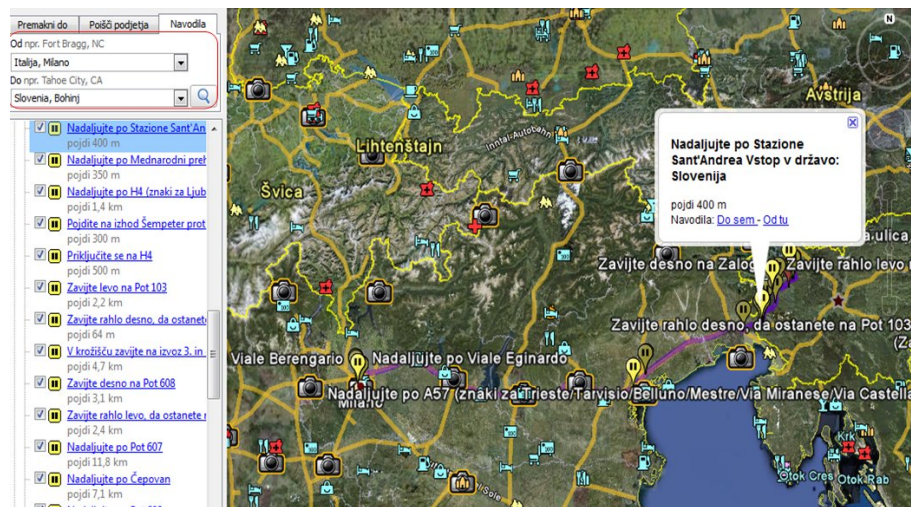
Načrtovanje poti

V sklopu poslovnih srečanj je v našem podjetju organiziran sestanek s poslovnimi partnerji iz številnih držav. Povabljenim gostom moramo na preprost način prikazati kako nas bodo najhitreje našli, kar storimo s pomočjo programskega orodja Google Zemlja, katerega lahko upravljajo tudi naši poslovni partnerji, kjerkoli na zemeljski obli. Načrtovanje poti je povsem enostavno in lahko dostopno. V meniju *Navodila* vpišemo lokacijo odhoda (od: Italija, Milano) in lokacijo prihoda (do: Slovenija, Bohinj) (glej Sliko 6.10).

Google Zemlja opiše celotno pot potovanja. Na zemljevidu in v samem besedilu so natančno opisane smeri, poti in razdalje do zelene točke. Potrebno je slediti navodilom, ki se izpišejo in nas popeljejo do končne relacije. V primeru, če želimo pogledati posamezne odseke na zemljevidu, kliknemo nanje. Prav tako se s krogcem na miški približamo ali oddaljimo od določene lokacije oz. posameznega manjšega segmenta. Na koncu se izpiše rezultat predvidenega časa in skupnih kilometrov (skupno 486 km, trajanje potovanja približno 5 h in 15 min).

Značilnosti iskane lokacije

Podjetje OpenStorage se nahaja v občini Bohinj, natančneje na Triglavski cesti X v Bohinjski Bistrici. V orodni vrstici *Poišči podjetje* v okence *Kje* zapišemo: Bohinj, Triglavska cesta. V primeru, ko iskalec ne ve točnega naslova lahko ime kraja napiše v okence *Kraj* in se nato premika po samem zemljevidu, da poišče



Slika 6.10: Načrtovanje poti

pravo ulico. V okence *Kaj* vpišemo le besedo *podjetje*, saj je delo enostavnejše in prikaz rezultatov hitrejši. Izbrana možnost v okencu na levi strani izpiše imena in naslove vseh podjetij, ki se nahajajo v občini Bohinj (glej Sliko 6.11).

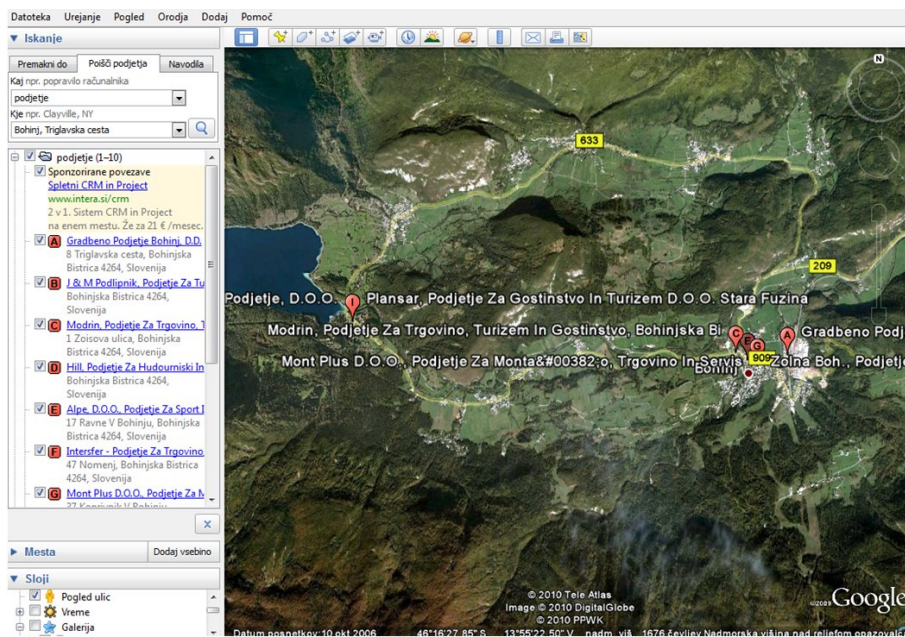
Navigacijski gumb

Na voljo je tudi navigacijski gumb, katerega izberemo v orodni vrstici *Pogled*. Izberemo na kakšen način naj se prikaže navigacija - *Pokaži navigacijo*. Izberemo možnost *Samodejno*. Navigacijske gumbe uporabljamo za povečevanje ali pomanjševanje, nagibanje in vrtenje pogleda (glej Sliko 6.12).

Z uporabo navigacijskih gumbov spremenimo pogled, kar je še posebej uporabno, kadar opazujemo kraje na višjih reliefnih oblikah (glej Sliko 6.13).

Dodajanje slojev

V polju *Pogled* imamo možnost izbire med vsemi razpoložljivimi sloji (*Vsi sloji*), ključnimi sloji (*Jedro*) ali samo tistimi, ki so trenutno prikazani (*Omogočeni zdaj*). Funkcija *Sloji* v programu Google Zemlja zagotavlja veliko različnih podatkovnih točk, ki jih prikažemo nad svojim območjem ogleda. Vsebujejo zanimivosti in zemljevid, ceste, relief ter celo podatke o zgradbah. V meniju *Sloji* določimo predmete, ki naj bodo razvidni na sliki. Primer: Če nas zanima zgolj en segment, npr. pogled ulic, kliknemo na ta sloj, izberemo možnost *Začni iskanje* in zemljevid se posodobi. Nekaj izmed slojev je prikazanih na Sliki 6.14.



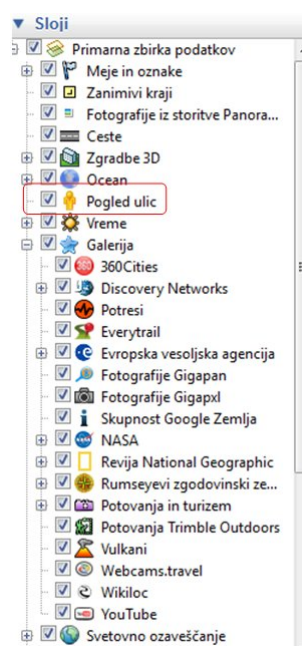
Slika 6.11: Iskanje podjetja



Slika 6.12: Navigacijski gumb



Slika 6.13: Sprememba reliefnega pogleda



Slika 6.14: Dodajanje slojev

Programsko orodje na dani lokaciji, glede na izbrane možnosti izriše določene zanimivosti. Na Sliki 6.15 je prikazano večje število slojev na relaciji v bližini občine Bohinj, kjer se nahaja namišljeno podjetje. Poslovni partnerji si lahko s pomočjo programskega orodja ogledajo dodane fotografije, video, geografske značilnosti kraja ipd.

Po ogledu osnovnih značilnosti občine Bohinj se odpravimo poiskati namišljeno podjetje na Triglavsko cesto. Pomembno je, da v slojih označimo izpis ulic in ostalih značilnosti, ki so v danem trenutku pomembne (predvsem izpostavimo ceste, ulice in zgradbe). Z vrtečim se miškinim gumbom se še bolj približamo zgradbi, ki nas zanima in cilj je dosežen (glej Sliko 6.16).

S klikom na ikono *Geografske značilnosti* si lahko ogledamo značilnosti, ki se nahajajo v bližini določene lokacije.

Pod oznako ~~xxx~~ Tube se nahaja video

Medvedov Vrh
11:10 četrtek (GMT+02:00)

Sponzorirane povezave

Shujkaj 2.27Kg na teden
V samo 1 mesecu se pripravi na poletje s popolno postavol wulong onlos.si/hujsanje

Fotografije

Panorama

Dog on kayak

0:03 / 0:19

Ustvaril: [ChokoSan](#) | Št. ocen: 1 ★★★★★

Bohinjsko Jezero

Medvedov Vrh

A Vie

209

Bohinjsko Jezero

Dr. basic
Mapa: [Mapa](#)
Komentarji
Panoramo
[Naloži svoj photo](#)

Sponzorirane Povezave
[Spartan Triathlon](#)
Povezave v Bohinju - Rezervacije in
Kontaktirajte nas
[www.spartan-triathlon.eu](#)

Pod modrim kvadratom se nahajajo fotografije

Slika 6.15: Dodani sloji



Slika 6.16: Triglavska cesta X

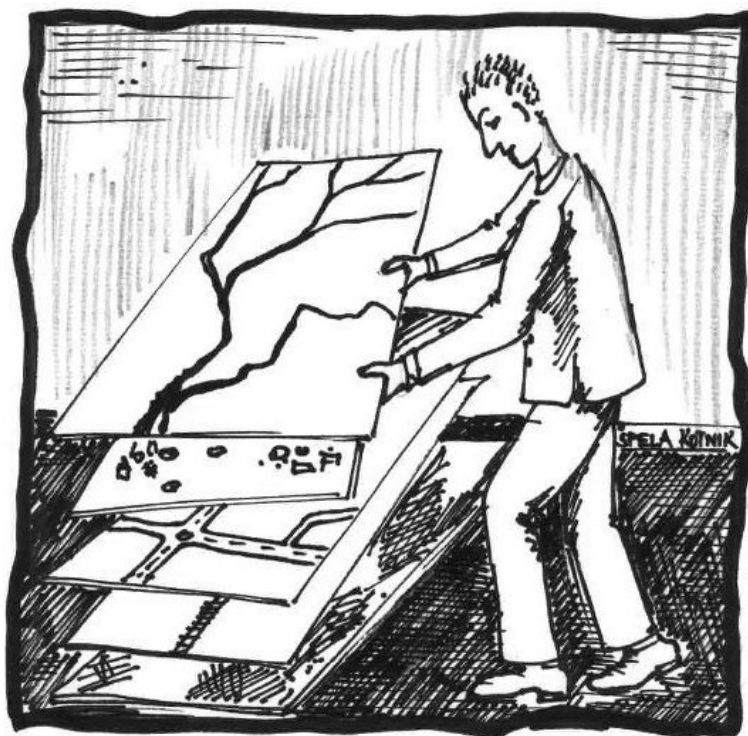
Povzetek

V vsakdanjem življenju se pogosto srečujemo s problemom mobilnosti in pomanjkanjem časa. Pogosto se zgodi, da se nenadoma odločimo za poslovno potovanje v tujino, čeprav niti ne vemo, kako bomo tja prispeli. V takšnih primerih je v pomoč eden izmed prosto dostopnih programskih orodij Google Zemlja, s katerim načrtamo poti s t.i. digitalnimi zemljevidi. Na preprost način poiščemo kraje, poslovne subjekte, izračunavamo razdalje in izrisujemo poti na zemeljskem površju. Iz ene določene lokacije na zemljevidu lahko v trenutku poletimo na drug konec sveta. Polet prek zemeljske oble spremljamo iz ptičje perspektive. S pomočjo programskega orodja dodajamo in shranjujemo zaznamke ter opise za poljubno točko na zemljevidu, na katero se lahko kasneje vrnemo.

Izmed številnih segmentov, ki jih ponuja programsko orodje, se osredotočimo na načrtovanje poti od poslovnega partnerja v Italiji do našega podjetja. Nato mu izdelano pot posredujemo, kar mu omogoči enostavnejši prihod v namišljeno podjetje. Smiselno je, da v primeru poslovnega potovanja v tujino oz. pričakovanega obiska poslovnih partnerjev, predhodno načrtamo potek poti. Bralca seznanimo z osnovnimi orodji, ki jih potrebuje, da z nekaj kliki obide svet v nekaj sekundah.

Poglavje 7

QUANTUM GIS - geografsko informacijsko planiranje



Geografski informacijski sistem

Vektorski, rastrski podatki

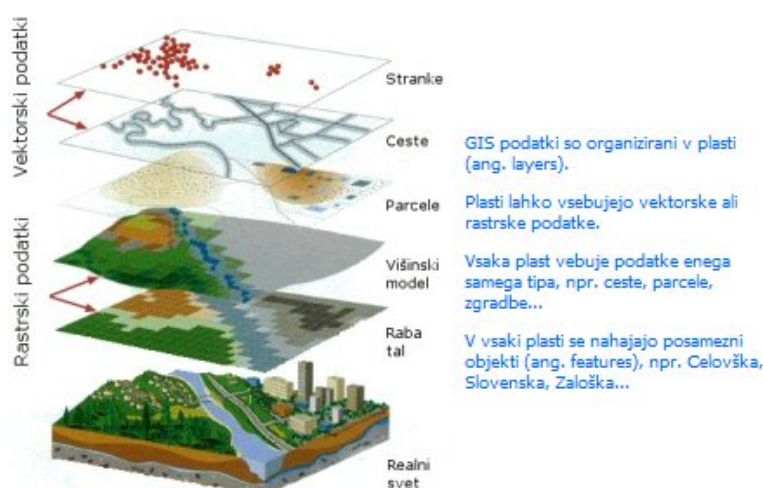
Primer: izbira lokacije za izgradnjo skladišča z upoštevanjem tveganj

7.1 Teoretično ozadje

7.1.1 Geografsko informacijski sistem

Geografski informacijski sistem (GIS, *Geographic Information System*), je sistem za urejanje in upravljanje prostorskih podatkov. Gre za skupek strojne in programske opreme, ki omogoča urejanje, upravljanje, analiziranje, modeliranje, predstavitev in prikaz geografsko referenciranih podatkov z namenom reševanja kompleksnih problemov planiranja in upravljanja virov.

GIS bi lahko poimenovali kot "pametno karto", ki omogoča pridobivanje odgovorov na najrazličnejša vprašanja - katera področja imajo primerno sestavo tal za postavitev skladišča, gostota naselitve na določenem območju ipd. GIS ne odgovarja zgolj na enostavna vprašanja, ki se tičejo pozicije, pač pa kombinira najrazličnejše podatke, tako prostorske kot neprostorske (tematske), zato so geografski podatki postali osnova za kvalitetno odločanje.



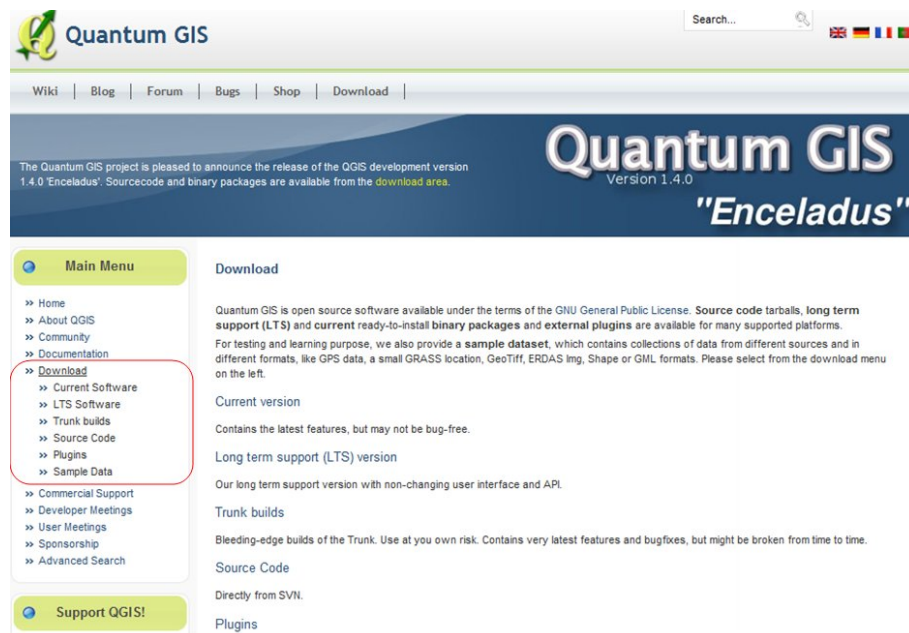
Slika 7.1: Rastrski in vektorski podatki

Vir: [9]

7.2 O programskem orodju

Quantum GIS (QGIS) je uporabniku prijazen odprtokodni geografski informacijski sistem (GIS) z licenco GNU General Public License. QGIS je uradni projekt odprtokodne geoprostorske organizacije (OSGeo). Deluje na različnih platformah - Linux, Unix, Mac OSX in Windows ter podpira številne vektorje, raste, podatkovne baze, formate in funkcionalnosti. Podpira tudi številne formate, kot so vektorski (knjižnica OGR – *shapefiles* ESRI, MapInfo SDTS in GML), rasterski formati, ki jih podpira GDAL, GRASS lokacije, spletni podatki, skladni z WMS ali WFS. Programsko orodje omogoča kreiranje kart in interaktivno raziskovanje prostorskih podatkov z uporabniku prijaznim grafičnim vmesnikom. GUI (*Graphic User Interface*) zajema veliko koristnih orodij: urejanje atributov, prostorski zaznamki, funkcija označevanja, spremembe vektorskih in rastrskih simbiologij, prepoznavanje funkcij, shranjevanje in obnavljanje projektov itd. Kreiranje, urejanje in izvoz prostorskih podatkov poteka z uporabo digitalizacije orodij za GRASS in *shapefile* formatov, GPS orodja za uvoz in izvoz GPX formata in pretvorbo drugih oblik.

GIS je zbirka programske opreme, ki uporabniku omogoča ustvarjanje, vizualizacijo, poizvedbe in analizo geoprostorskih podatkov. Program je uradno začel delovati maja 2002. Idejo so zasnovali le nekaj mesecev prej, ko je Gary Sherman pričel z iskanjem pregledovalnika GIS za Linux, ki bi bil hiter in bi



Slika 7.2: Namestitev programa – 1. del

podpiral široko paleto shranjenih podatkih.

Prenos in namestitev

Programsko orodje Quantum GIS prenesemo s spletne strani ponudnika programskega orodja QuantumGIS [44]; pod razdelkom *Download* na levi strani. Po odprtju *Download* se na desni strani izpiše več možnosti, izmed katerih izberemo *Current version*. Uporabniki operacijskega sistema Ubuntu ga enostavno poiščejo, prenesejo in namestijo preko programskega središča Ubuntu oz. preko Upravljanja paketov Synaptic.

Ob izbiri dane možnosti se pojavi naslov *LTS Software*, kjer so na voljo posamezne verzije programskih orodij, ki jih lahko namestimo. Izberemo verzijo, primerno za operacijski sistem, ki ga uporabljamo (Windows, Ubuntu...) (glej Sliko 7.3).

Po uspešni namestitvi programskega orodja, se na namizju izrišejo številne ikone. Za zagon kliknemo zeleno ikono Q.

The screenshot shows the Quantum GIS website interface. At the top, there is a search bar and flags for various countries. Below the navigation menu (Wiki, Blog, Forum, Bugs, Shop, Download), a banner for Quantum GIS Version 1.4.0 "Enceladus" is displayed. The main content area is titled "LTS packages" and contains a paragraph explaining the LTS (Long Term Support) 1.0.x release series. Below this, there is a table listing available packages for different platforms and operating systems.

Quantum GIS
Version 1.4.0
"Enceladus"

The Quantum GIS project starts to publish short reports about QGIS usage on a regular basis. The first **QGIS Case Study** was contributed in May 2009 by Dr. Horst Düster from the Canton of Solothurn in Switzerland.

Main Menu

- » Home
- » About QGIS
- » Community
- » Documentation
- » Download
 - » Current Software
 - » **LTS Software**
 - » Trunk builds
 - » Source Code
 - » Plugins
 - » Sample Data
- » Commercial Support
- » Developer Meetings
- » User Meetings
- » Sponsorship
- » Advanced Search

Support QGIS!

[Make A Donation](#)

more information

Upcoming Events

FOSS4G 2010

LTS packages

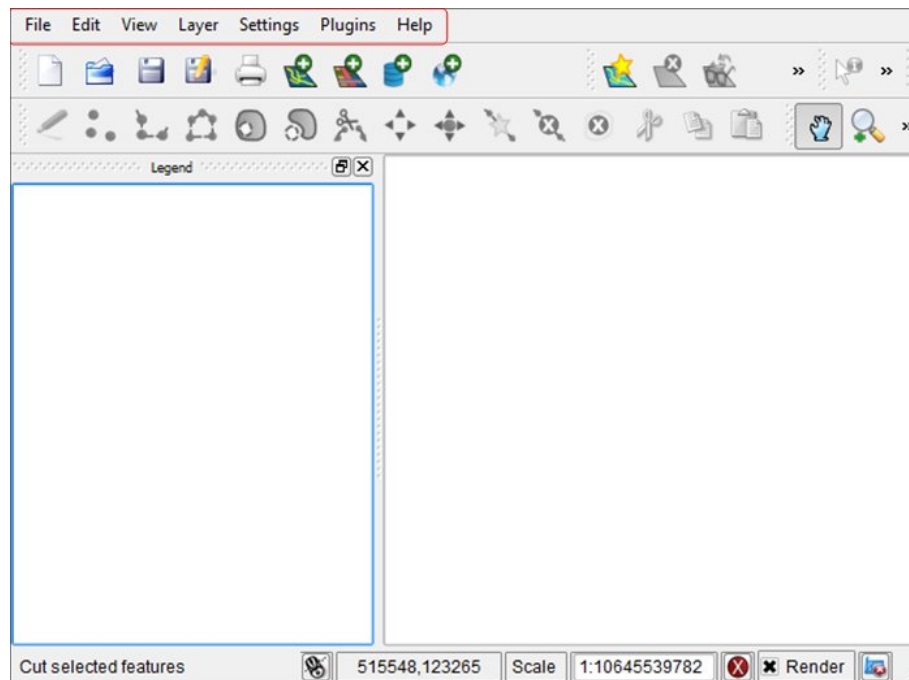
The **LTS (Long Term Support) 1.0.x release series** are part of our effort to provide a stable, unchanging, long term supported environment. Each minor release in the LTS series contains only bug fixes and no new features. The current official LTS version is **QGIS 1.0.2 "Kore"** and was released on 13 May 2009. For a list of bugs that were closed in the 1.0.2 release, please see the 1.0.2 release milestone. We offer Ready-to-Run software packages (.exe, .dmg, .deb, .rpm) for many platforms listed in the table below. If your OS is missing from the list below, please check again soon - we add new packages as they become available.

Show entries

Search:

Version	Platform	Package & Comment
1.0.2 LTS	Windows - Standalone	<p>Download: http://download.osgeo.org/qgis/win32/QGIS-1.0.2-0-Setup.exe</p> <p>This is an MSVC based build of QGIS. All dependencies are supplied in a standalone installer, including python, gdal (including MrSid Wavelet compressed raster support) etc. This is most suitable for people who have slow / no internet connections and who want to be able to share and pass around copies of the installer - the installer can be run offline since it contains everything needed to install QGIS and its dependencies. You can check the integrity of this installer against this checksum. Note: You may need to install the Microsoft C Runtime Libraries too. If QGIS does not start, get the MSVC libs from here and install them before trying to run QGIS again.</p> <p>Note: This does NOT include GRASS</p>
1.0.2 LTS	Windows - OSGeo4W	<p>Download: http://trac.osgeo.org/osgeo4w/</p> <p>This is also an MSVC based build of QGIS. All dependencies are supplied via the OSGeo4W installer, including python, gdal (including MrSid and ECW Wavelet compressed raster support) etc. This is most suitable for people who want to be able to update dependencies from the OSGeo4W repository easily. The installer is able to install from internet or just download all needed packages beforehand. In both cases the downloaded files are kept in a local directory for future installations.</p> <p>Note: This build is an online installer.</p>
1.0.2 LTS	Mac OS X	<p>Download: http://www.kyngchaos.com/software/qgis</p> <p>This Mac OS X version requires separate installation of dependency frameworks (available also from www.kyngchaos.com).</p>
1.0.2 LTS	openSUSE 11.0, 11.1 and Factory	<p>Download: http://download.opensuse.org/repositories/Application/Geo/</p> <p>Add the repository to your installation manager. All packages include GRASS and Python support.</p>

Slika 7.3: Namestitev programa – 2. del



Slika 7.4: Programsko okno

Programsko okno

Programsko okno Quantum GIS v menijski vrstici ponuja sedem razdelkov (glej Sliko 7.4).

Razdelek *File* omogoča odpiranje novega dokumenta (*New Project*), na voljo je tudi možnost odpiranja že shranjenih projektov oz. dokumentov (*Open Project*). Dokument lahko shranimo na želeno mesto (*Save Project*, *Save Project As*), ga natisnemo (*Print Composer*) ali zapremo (*Exit*).

Razdelek *Edit* podrobneje opišemo pri konkretnem delu na projektu. Razdelek *View* omogoča povečanje slike na projektu in ostale možnosti, ki so na voljo v orodni vrstici. Razdelek *Layer* omogoča dodajanje posameznih vektorskih, rasterskih, POSTGIS in WMS slojev (*Add Vector Layer*, *Add Raster Layer*, *Add PostGIS Layer*, *Add WMS Layer*). V tem razdelku imamo še možnost odpreti atributno tabelo (*Open Attribute Table*), pregled lastnosti (*Properties*) itd. (glej Sliko 7.5).

File	Edit	View	Layer	Settings	View	Layer	Settings	Plugins	Layer	Settings	Plugins	Help
New Project				Ctrl+N	Pan Map				New Vector Layer...			N
Open Project...				Ctrl+O	Zoom In			Ctrl++	Add Vector Layer...			V
Open Recent Projects					Zoom Out			Ctrl+-	Add Raster Layer...			R
Save Project				Ctrl+S	Select Features				Add PostGIS Layer...			D
Save Project As...				Ctrl+Shift+	Identify Features			I	Add WMS Layer...			W
Save as Image...				Ctrl+I	Measure Line			M	Open Attribute Table			
Print Composer				Ctrl+P	Measure Area			J	Toggle editing			
Exit				Ctrl+Q	Zoom Full			F	Save as Shapefile...			
					Zoom to Layer				Save Selection as Shapefile...			
					Zoom to Selection			Ctrl+J	Remove Layer			Ctrl+D
					Zoom Last				Properties...			
					Zoom Actual Size				Add to Overview			O
					Map Tips				Add All to Overview			+
					New Bookmark...			Ctrl+B	Remove All From Overview			-
					Show Bookmarks			B	Hide All Layers			H
					Refresh			Ctrl+R	Show All Layers			S
					Panels							
					Toolbars							
					Toggle Full Screen Mode							

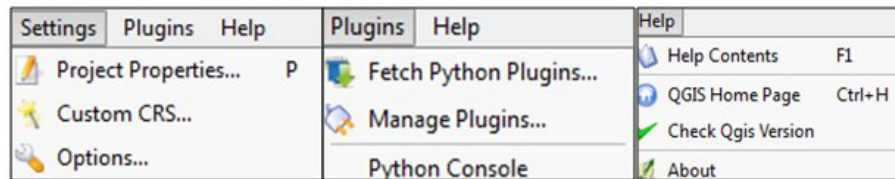
Slika 7.5: Menijska vrstica – 1. del

Rastrski podatki

Ena izmed vrst geoprostorskih podatkov so *rastrski podatki* ali preprosto "raster". Najbolj enostaven način prepoznanih oblik rastrskih podatkov so digitalni satelitski posnetki oz. zračni posnetki in digitalni modeli nadmorskih višin. Raster je mreža, ki jo sestavljajo celice oz. posnetki in slikovne pike. Vsaka celica ima številčno vrednost in določene geografske velikosti (npr. 30x30 m). Prekrivanje različnih rastrov se uporablja za predstavitev slik z več kot eno vrednostjo barve (npr. en raster za vsak sklop je rdeče barve, nato zelene, modre itd.). Vse skupaj omogoča barvno sliko. Rastrji z manjšimi celicami so podrobneje opredeljeni, vendar zasedejo bistveno več prostora. Trik je najti pravo ravnovesje med velikostjo celic za skladiščenje in velikostjo celic za analitične in druge namene.

Vektorski podatki

Tudi vektorski podatki se uporabljajo v geoprostorski aplikaciji. V najpreprostejšem smislu, so vektorji način opisovanja lokacij z uporabo niza koordinat. Vsako usklajevanje se nanaša na opis lokacij z uporabo sistema x in y vrednosti. Obstajajo različni načini, ki predstavljajo dane geografske koordinate, kar je odvisno od namena dela.



Slika 7.6: Menijska vrstica – 2. del



Slika 7.7: Orodna vrstica

Vektorski podatki se pojavljajo v treh oblikah:

- točke – usklajevanje le teh (xy) predstavlja diskretne geografske lokacije;
- linije – več koordinat nanizanih skupaj (x1y1, x2y2, x3y4...xnyn) v določenem vrstnem redu, imajo določeno dolžino in število vrstic z določeno smerjo - tehnično gledano je vrstica en par med seboj povezanih koordinat;
- mnogokotnik – vrstice so nanizane skupaj z več kot dvema točkama, zadnja točka je na enaki lokaciji kot prva, ključna lastnost je, da so znotraj njih določena območja.

Sledijo meniji Nastavitve (*Settings*), Priključki (*Plugins*) in Pomoč (*Help*). V meniju *Help* obstaja možnost podrobnejšega pregleda programa (*Help Contents*), do katerega dostopamo tudi preko tipke F1. Odpremo lahko QGIS spletno stran (*QGIS Home Page*) in pogledamo katere verzije programa so na voljo (*Check Qgis Version*) (glej Sliko 7.6).

Orodna vrstica vsebuje ukaze s pomočjo katerih ustvarjamo projekt. Na voljo imamo možnost odpiranja novih in že shranjenih dokumentov, ostalih dokumentov, tiskanje, dodajanje in ustvarjanje novih slojev, premikanje slojev ter izbiro ukazov, ki prikazujejo atributno tabelo (glej Sliko 7.7).

Na voljo je še orodna vrstica s pomočjo katere oblikujemo posamezne scenarije. Možnost delovanja vrstice je pogojena s tem, da je delovanje omogočeno šele, ko s klikom na prvo ikono preklopimo na urejanje (*Toggle editing*) (glej Sliko 7.8).



Slika 7.8: Preklopi na urejanje



Slika 7.9: Spletna stran ARSO

Pridobitev slojev

Za pričetek dela potrebujemo t.i. sloje, ki jih uvozimo v Quantum GIS. V namišljenem podjetju potrebujemo podatke o naravnih značilnosti, ki so prisotna na območju, katerega želimo izbrati za izgradnjo skladišča. S spletnega mesta Agencije RS za okolje - ARSO [1], izberemo možnost GIS spletne storitve (glej Sliko 7.9).

V oknu izberemo možnost *Vstop v WFS ARSO* (glej Sliko 7.10).

Klient deluje preko aplikacije Java, katero je potrebno predhodno namestiti. Ko je namestitev dokončana pričnemo z zagonom programa za pridobitev baze podatkov s strani ARSO. Prikaže se okno prikazano na Sliki 7.11.

V danem oknu izberemo možnost *Enostavni izvoz*, nakar za ažuriranje podatkov pritisnemo tipko *Osveži*. Pogledamo v meni *Sloj*, kjer so na voljo sloji, ki jih enostavno izvozimo glede na potrebe našega problema (glej Sliko 7.12).

V izbranem primeru najprej izvozimo sloj imenovan *Raziskave onesnaženosti tal Slovenije* (Slika 7.13). Iz okenca na levi strani (*Vsi stolpci*) izberemo tiste, ki jih želimo prenesti. V primeru, ko želimo prenesti vse, v sredinskem stolpcu

ARSO Metapodatkovni portal omogoča iskanje in brskanje po metapodatkovnih opisih, ki jih vodi Agencija za okolje.

- [Vstop v metapodatkovni portal](#)
- [Uporabniška navodila](#)

Centralna evidenca prostorskih podatkov(CEPP) prikazuje seznam obstoječih digitalnih prostorskih podatkov v Sloveniji. V njej si lahko ogledate metapodatke, ki jih vodi Geodetska uprava Republike Slovenije.

- [CEPP baza](#)

Pridobivanje podatkov

Pri uporabi podatkov, ki so bili pridobljeni s pomočjo spletnih storitev, je potrebno navesti vir podatka: Agencija RS za okolje !

WFS ARSO

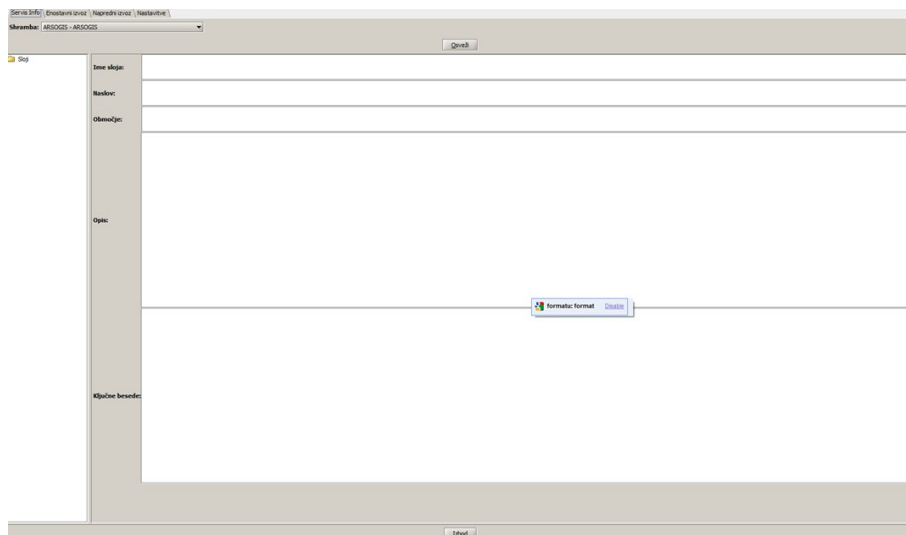
WFS ARSO omogoča izvoz prostorskih podatkov v .shp ali .gml formatu (verzija 2.0). Za delovanje aplikacije morate imeti naloženo JAVA programsko opremo!

- [Vstop v WFS ARSO](#)
- [Uporabniška navodila](#)

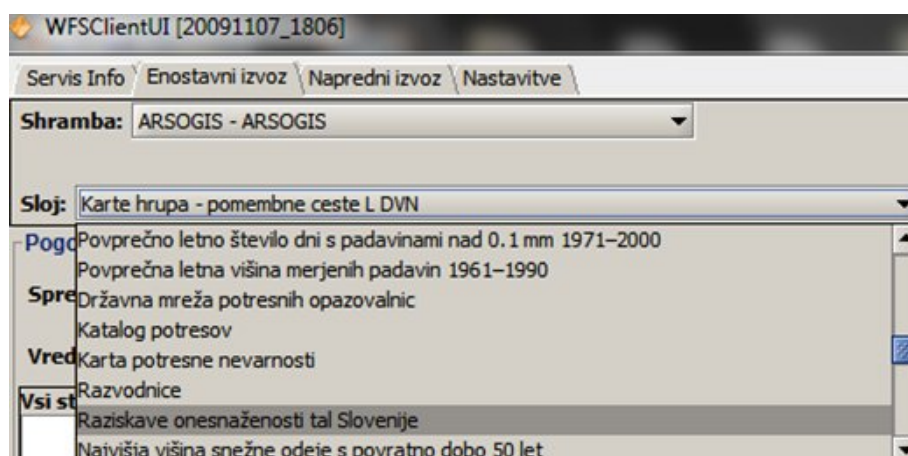
WFS vstopna točka

(prekopirate bližnjico <http://gis.arso.gov.si/wfs?STORE=ARSOGIS> v aplikacijo, kjer želite odpreti WFS) Spletna objektna storitev Web Feature Service (WFS) omogoča na podlagi zahtevka preko spleta, pridobitev geografskih podatkov. Za izmenjavo podatkov uporablja na XML-ju temelječ GML format.

Slika 7.10: Pridobivanje podatkov na WFS ARSO



Slika 7.11: Agencija Republike Slovenije za okolje



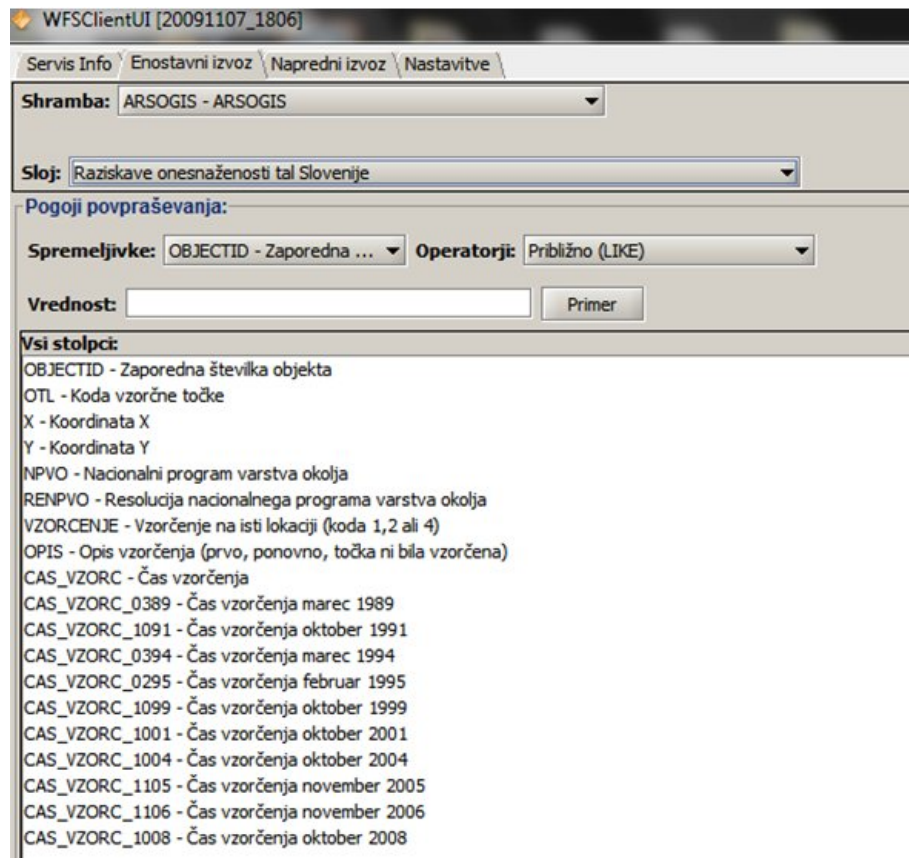
Slika 7.12: Enostavni izvoz WFSCient UI

izberemo tretjo možnost, ki je označena s tremi puščicami. V primeru, ko želimo prenesti zgolj eno ali dve kategoriji iz posameznega področja, pa le te izberemo posamezno, s klikom na enojno puščico. Če ugotovimo, da danega sloja ne potrebujemo, ga z nasprotno puščico iz desnega dela prenesemo nazaj v levo okence.

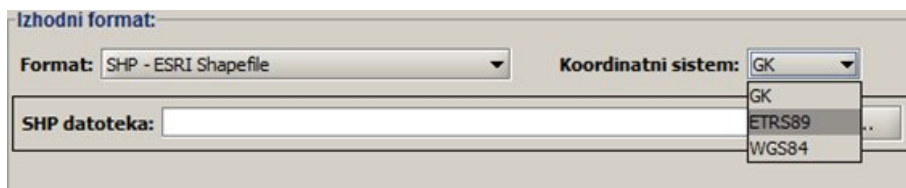
Spodaj levo se nahaja okence *Izhodni format*, kjer določimo format in koordinatni sistem. V meniju *Format* izberemo možnost *SHP-ESRI Shapefile*. Pri izbiri koordinatnega sistema so na voljo tri možnosti – GK, ETRS89 in WGS84.

GK koordinatni sistem je uzakonjen koordinatni sistem z uradnim imenom D-48. Geodetska uprava Republike Slovenije vodi državni koordinatni sistem Republike Slovenije, ki ga predstavljajo točke temeljne horizontalne mreže. Temeljne horizontalne točke vsebujejo več vrst geodetskih točk glede na natančnost njihovih koordinat in način njihove določitve - trigonometrične točke od I. do IV. reda, poligonometrične točke in navezovalne točke. Geodetske točke temeljne horizontalne mreže imajo podane koordinate v 5. meridianski coni Gauss-Krügerjeve ravninske projekcije, preslikane z elipsoida Bessel 1841, ki je orientiran glede na telo Zemlje v fundamentalni točki Hermannskogel z orientacijo na Hundesheimer Berg, ter višine v sistemu normalnih ortometričnih višin z izhodiščem v Trstu preračunano na raven fundamentalnega reperja Ruše. GPS sprejemniki merijo položaj v WGS84 (*World Geodetic System 1984*), lahko pa jih nastavimo na Gauss-Krügerjev, to je Bessel 1841.

ETRS89 koordinatni sistem (*European Terrestrial Reference System 1989*) je sodoben koordinatni sistem, ki je zamenjal obstoječega, kateri izhaja še iz časa Avstroogrške. Menjava sistema je bila potrebna, saj Gauss-Krügerjev koordinatni sistem ni kompatibilen z GPS elipsoidom WGS84, zaradi česar je



Slika 7.13: Izbira sloja



Slika 7.14: Izhodni format

posledično prihajalo do napak in deformacij pri pretvarjanju vrednosti med sistemi. Tako bodo v bodoče podatki, ki jih posreduje GPS naprava, uporabni tudi na bodočih koordinatnih kartah.

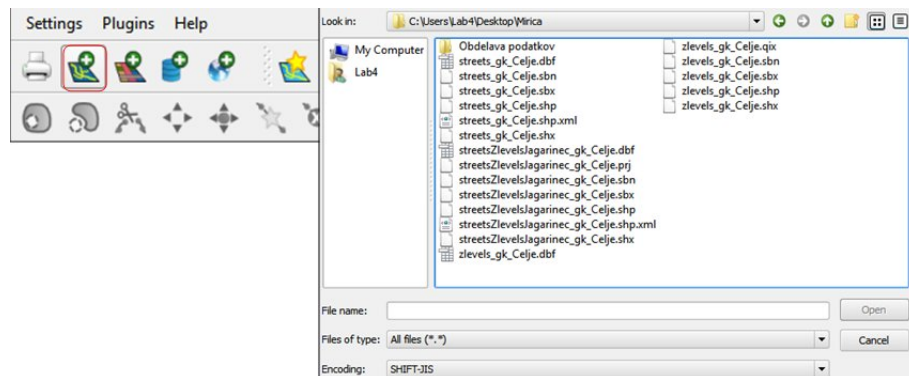
WGS84 koordinatni sistem je različica koordinatnega sistema (ang. *World Geodetic System*, slo. Svetovni geodetski sistem) iz leta 1984. Določa globalni referenčni okvir za Zemljo, uporaben je za geodezijo in navigacijo.

Pomembno je, da vemo v katerem koordinatnem sistemu imamo podane sloje. V primeru, ko potrebujemo zgolj sloje, ki so dostopni na spletnem strežniku ARSO, izberemo kateri koordinatni sistem želimo. V primeru, če s strani Mestne občine Celje (v nadaljevanju MOC) pridobimo sloje, ki so zapisani v GK koordinatnem sistemu, za pridobitev ostalih slojev iz ARSO izberemo GK koordinatni sistem. Pogost problem je nezmožnost spreminjanja koordinatnega sistema. V tem primeru izvedemo transformacijo podatkov iz danega v želeni koordinatni sistem. GK koordinatni sistem ni kompatibilen z GPS elipsoidom WGS84, zato so uvedli nov koordinatni sistem, imenovan ETRS89.

Pri delu ugotavljamo kompatibilnost podatkov s strani treh različnih inštitucij - Navteq podatki, podatki MOC in podatki ARSO. Navteq podatki so shranjeni v WGS84 koordinatnem sistemu; podatki pridobljeni od MOC so shranjeni v GK koordinatnem sistemu, podatki iz ARSO pa omogočajo izbiro med tremi možnostmi koordinatnih sistemov (vključen tudi koordinatni sistem ETRS89). Podatki iz ARSO in MOC so med sabo kompatibilni. Na isti sloj ni možno uvoziti Navteq podatkov, saj jih je potrebno predhodno transformirati in uskladiti z izbranim koordinatnim sistemom.

Posameznik, ki se ukvarja z dano problematiko lahko za pridobitev slojev zaprosi tudi posamezne inštitucije:

- DRSC – Direkcija Republike Slovenije za Ceste (povprečni letni dnevni promet, prometni tokovi ...);
- GURS – Geodetska uprava Republike Slovenije (sloji hišnih števil, sloji naselij ...);
- ARSO – Agencija Republike Slovenije za okolje (raziskave onesnaženosti



Slika 7.15: Dodajanje slojev

tal, hrup, poplave, potresi);

- SURS – Statistični urad Republike Slovenije (statistični podatki);
- Občine ipd.

Dodajanje slojev

Ko pridobimo zelene sloje pričnemo z delom. V orodni vrstici izberemo možnost *Add Layers*, ki jo predhodno uvozimo v eno izmed datotek. V meniju izberemo datoteko z danimi podatki. Odpremo datoteko, v kateri imamo shranjene podatke v GK koordinatnem sistemu, navezujoč na MOC. Na voljo je več tipov datotek: *dbf.*, *sbx.*, *shp.*, *xml.* in *shx.* Za dani primer je pomembna datoteka *.dbf*, katero odpremo z dvojnimi klikom na dani dokument. Dokument lahko odpremo tudi izven programa, kjer se prikaže preglednica z danimi podatki (glej Sliko 7.15).

Po odprtju dane datoteko se na polju pojavijo ulice v MOC. Na levi strani odpremo legendo, ki jo ustvarimo pod razdelkom *View – Panels - Legend*. Ko želimo nadalje upravljati s podatki kliknemo legendo na levi strani in izberemo možnost Lastnosti (*Properties*) (glej Sliko 7.16).

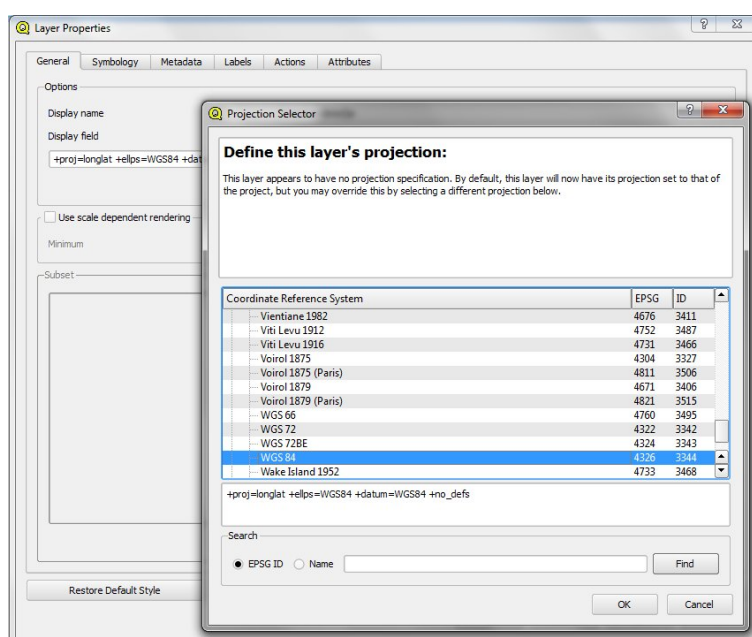
Znotraj dane možnosti je na voljo šest menijev. V meniju Splošno (*General*) so zapisani osnovni podatki dane datoteke ter drugi podatki navezujoči na dani sloj (legenda sloja, sličice sloja ipd.) (glej Sliko 7.17).

Meni Simbiologija (*Symbology*) omogoča poljubno spreminjanje opcije posameznega sloja, stil linije (*Outline style*), barvo (*Outline color*), poudarjenost linij (*Outline width*) itd. (glej Sliko 7.18).

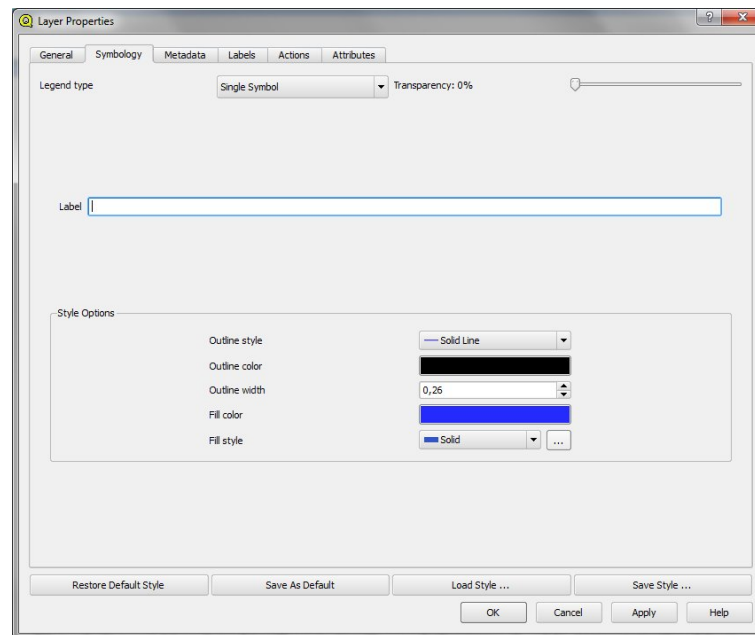
Meni Metadata prikazuje številne podatke o rastrih, vključno s statistiko. Funkcija *Display Labels* v meniju *Labels* omogoča dodajanje podatkov danega



Slika 7.16: Prikaz lastnosti slojev



Slika 7.17: Properties - General



Slika 7.18: Properties - Symbologyl

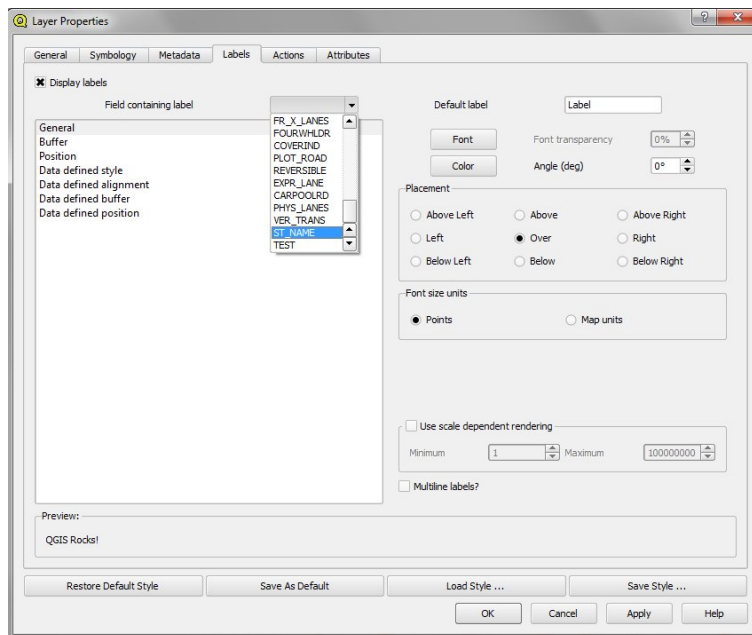
sloja. V okencu *Field containing label* izberemo zeleno področje za prikazan na določenem sloju, npr. ST-NAME pomeni, da se bodo na osnovnem oknu pojavila imena ulic (glej Sliko 7.19).

V meniju *Attributes* je zapisana tabela vseh podatkov določenega sloja (glej Sliko 7.20).

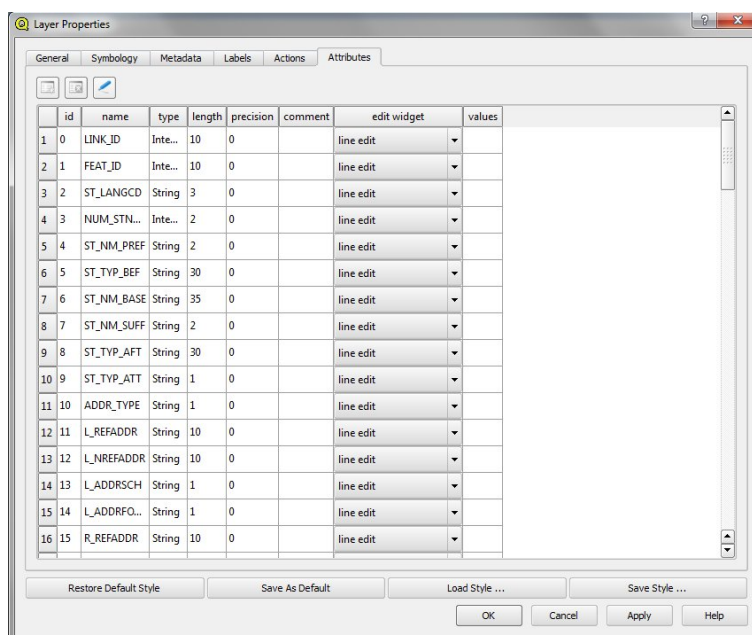
Problem

S programskim orodjem Quantum GIS predstavimo tveganja, ki nastanejo pri izbiri skladiščne lokacije na območju MOC. Preučimo podatke pridobljene s strani ARSO, MOC in SURS-a. Na slikoven način prikažemo območje izgradnje skladišča z vsemi omejitvami in tveganji na danem območju.

Na voljo so še številne baze podatkov - komunalne čistilne naprave, ekološko pomembna območja, hidrografska območja, industrijska odlagališča odpadkov, kategorizacija vodotokov, povprečna letna višina korigiranih padavin, povprečna letna temperatura zraka, karte hrupa, opozorilna karta poplav, vodna dovoljenja, natura, hrup ipd. Za področje logistike je predvsem pomembna pridobitev podatkov, navezujočih na cestno omrežje, naselja, hišne številke ipd.



Slika 7.19: Properties - Labels



Slika 7.20: Properties - Attributes

7.3 Uporaba

Na podlagi pridobljenih podatkov celotne Slovenije pregledamo podatke o značilnostih občine, ki nas zanima. Na Sliki 7.21 je prikazana atributna tabela in karta števila prebivalcev celotne Slovenije. S klikom na posamezno lokacijo se obarva posamezna občina, z dodatkom *labele JAN2010* pa se izpiše še število prebivalcev.

Na sloju MOC grafično prikažemo tamkajšnja naselja. V orodni vrstici izberemo možnost *Add Vector Layer* ter poiščemo bazo, kjer so le ta shranjena. Odpremo datoteko s končnico *.dbf*, nakar se izriše karta. Ob pogledu na atributno tabelo vidimo, da lahko posamezna naselja tudi ustrezno poimenujemo. To storimo tako, da v meniju danega sloja, kliknemo na levi miškin gumb, izberemo možnost *Properties* ter nato *label NA_UIIME*. Izberemo tisto naselje, ki nas v danem trenutku zanima. Izriše se karta, ki je prikazana na Sliki 7.22.

Če v nadaljevanju sloja naselij ne potrebujemo več, izberemo možnost *Remove*, ki omogoča, izbris sloja iz karte. Za nadaljnje delo uvozimo sloj cest MOC. V atributni tabeli izberemo podatke katere želimo, da se izpišejo na danem sloju. V danem primeru izberemo *label ST_NAME*, kar na karti izpiše imena ulic (glej Sliko 7.23).

V nadaljevanju podrobneje pogledamo cestno omrežje v MOC, saj le ta omogoča boljšo orientacijo pri pregledovanju karte (glej Sliko 7.24).

Na podlagi pridobljenih slojev s strani MOC dodamo še sloje hišnih števil. Pomembno je da vemo, kje se nahaja največja kapaciteta ljudi, saj na območju goste naselitve ne bomo gradili skladišča. Poiskati moramo prostor, kjer nas to vrstni dejavnik ne ovira (glej Sliko 7.25).

Če bi namesto "krogcev" želeli imeti prikazane druge simbole, v meniju *Properties* izberemo *Symbology* (glej Sliko 7.26).

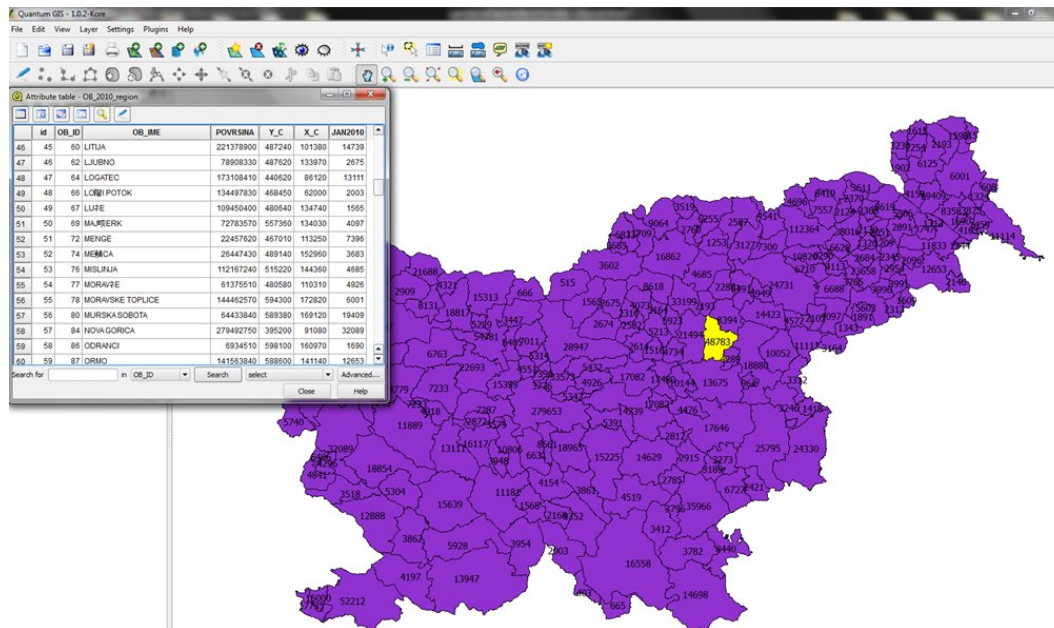
Namesto krogcev se izrišejo "hišice". Izbiramo lahko tudi samo barvo in velikost ikon (glej Sliko 7.27).

Ob pregledu celotnega zemljevida MOC pri sloju cest izberemo možnost, ki izpisuje imena cest, ulic in mest, saj takrat karta postane neberljiva. Slika 7.28 prikazuje stanovanjska naselja, posredno pa gostoto naselitve.

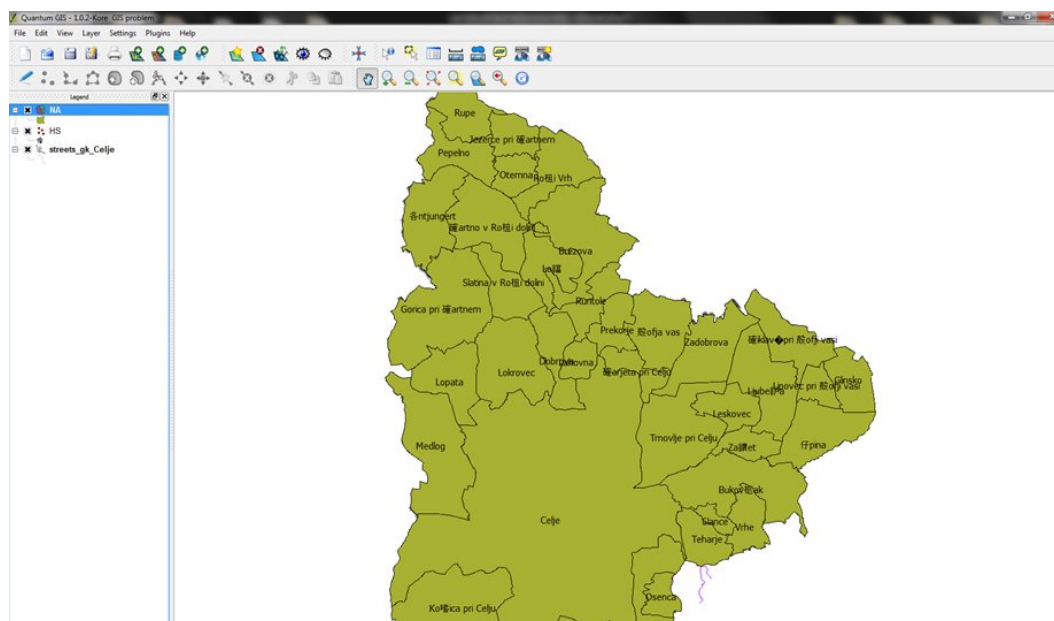
V nadaljevanju se osredotočimo na območje Teharja, saj je na danem območju gostota naselitve bistveno manjša, kot je na območju samega mestnega jedra (glej Sliko 7.29).

Podrobneje preučimo še tveganja, ki so prisotna na območju, kjer želimo postaviti skladišče. Slika 7.30 prikazuje zavarovana naravna območja (označena z rjavo barvo). Razvidno je, da se v okolici Teharja ne nahajajo tovrstna območja, torej ne obstajajo tveganja posega v zavarovana naravna območja.

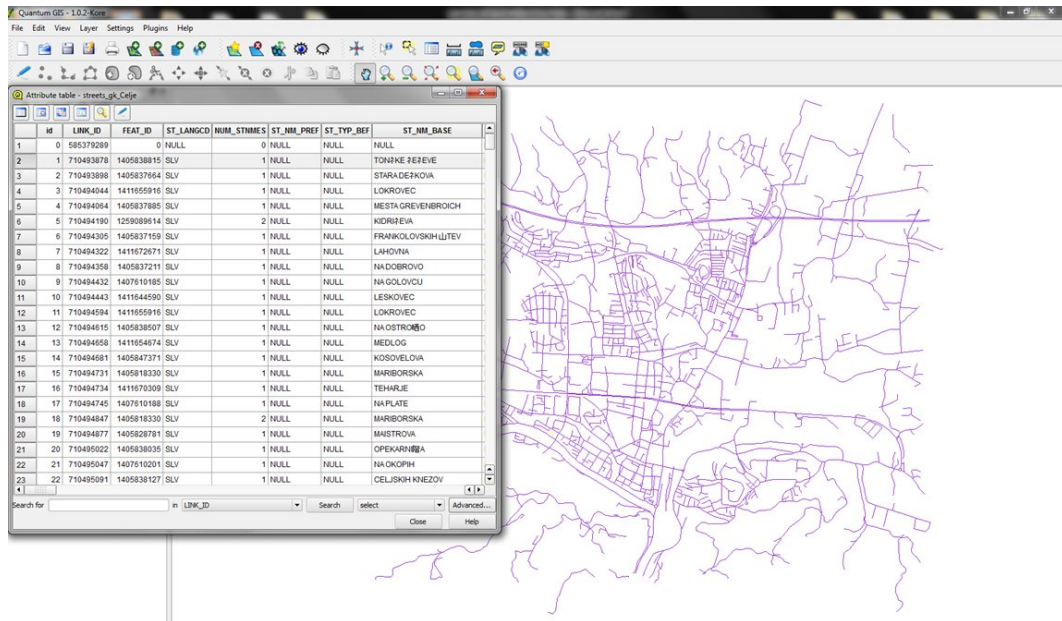
Z namestitvijo sloja iz spletne strani ARSO za celotno Slovenijo prikažemo



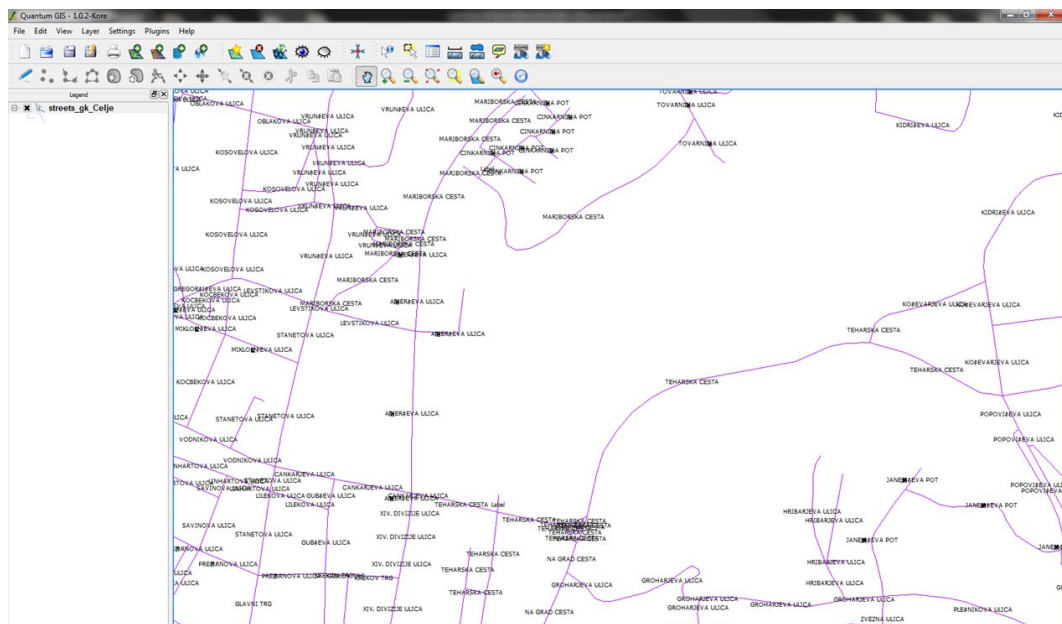
Slika 7.21: Število prebivalcev v MOC - januar 2010



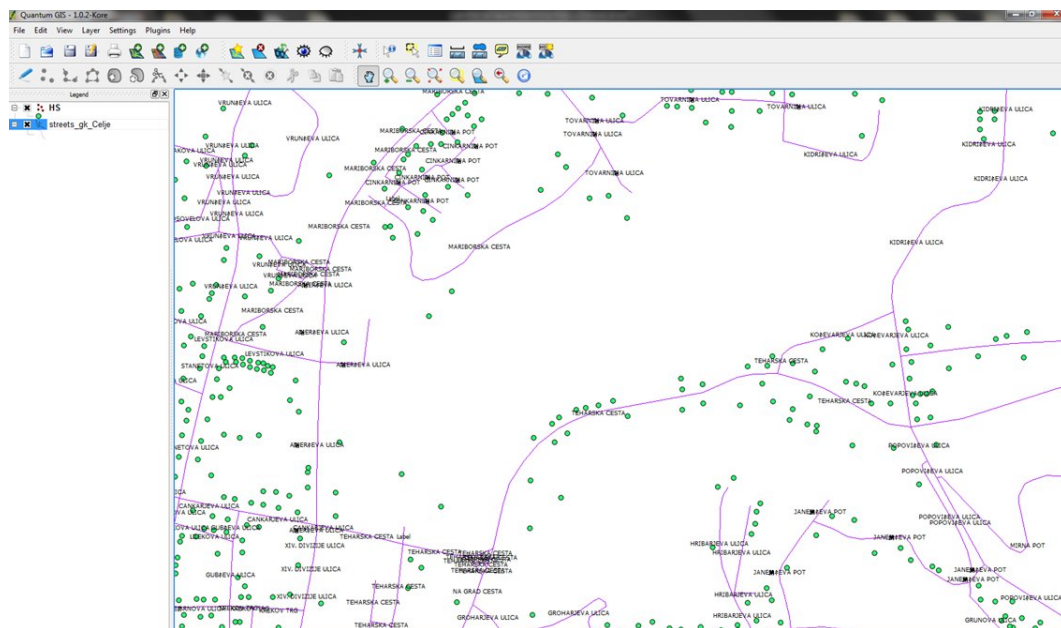
Slika 7.22: Naselja v MOC



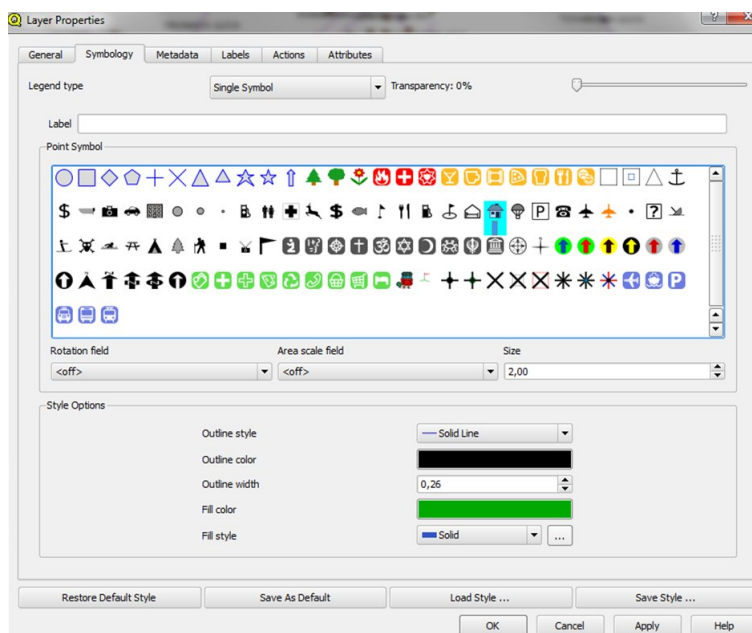
Slika 7.23: Pogled v atributno tabelo cest v MOC



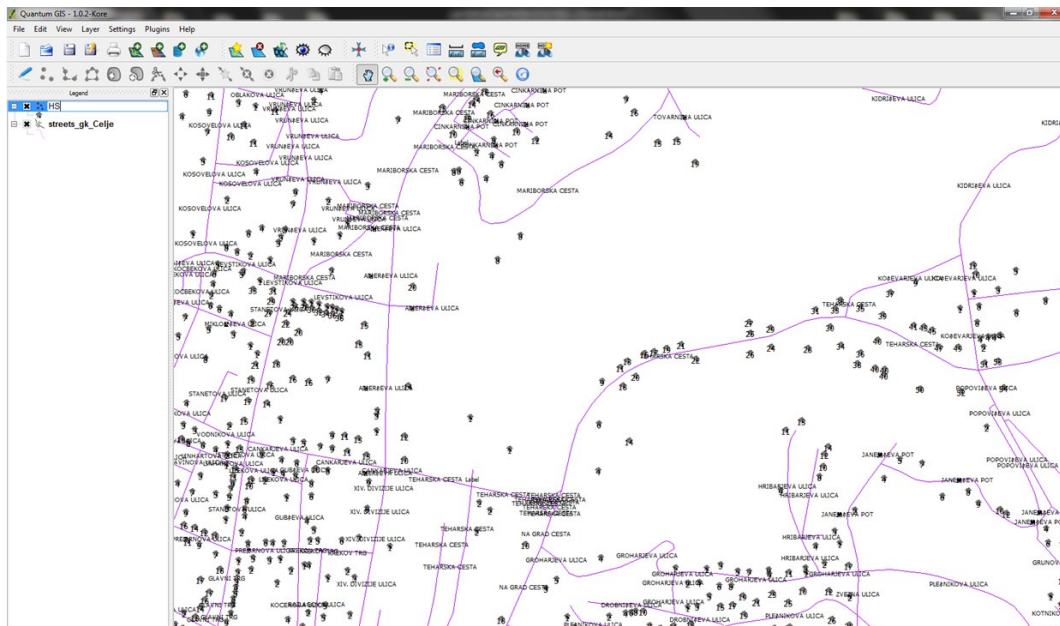
Slika 7.24: Dodajanja imena cest, ulic in poti



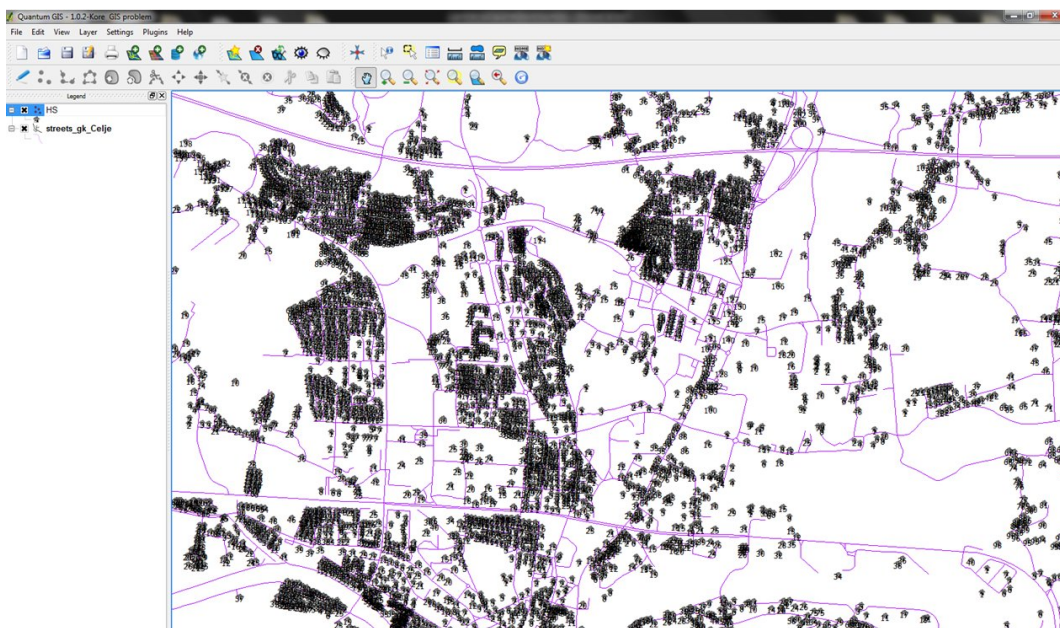
Slika 7.25: Dodajanja slojev hišnih številk



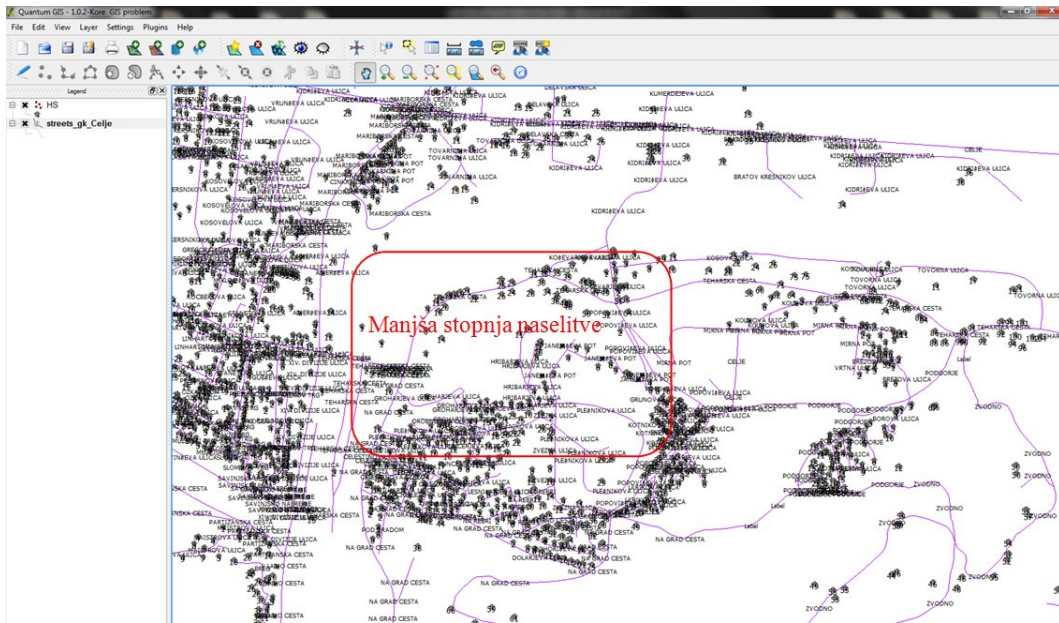
Slika 7.26: Spreminjanje oznak



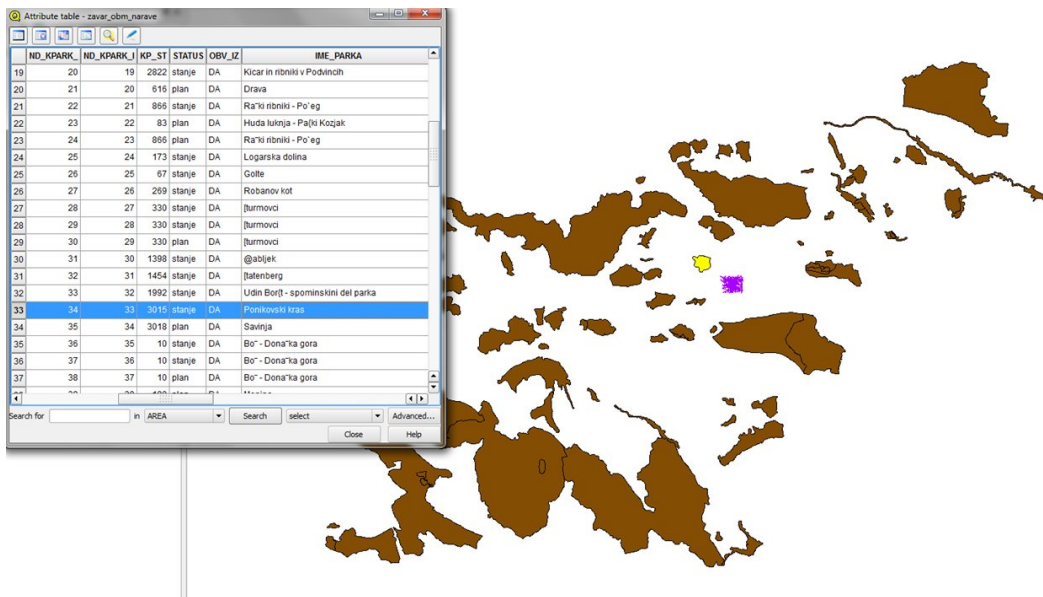
Slika 7.27: Dodajanje številk stanovanj



Slika 7.28: Gostota naselitve stanovanj



Slika 7.29: Stopnja naselitve v bližini Teharja



Slika 7.30: Zavarovana območja narave

ekološko pomembna območja (jame, doline, zatoke, jezera, doline, gore, potoke ipd.). V karto umestimo še pridobljene sloje iz MOC. S klikom na karto se približamo danemu območju in tako preverimo katera ekološko pomembna območja se nahajajo v bližini Teharja, kjer želimo zgraditi skladišče. Iz Slike 7.31 je razvidno, da se v celotni okolici MOC, ne nahajajo ekološko pomembna območja.

Eno izmed pomembnih tveganj so tudi hidrografska območja v okolici MOC oz. natančneje v bližini Teharja. V atributni tabeli se nahaja hidrografska območje Celjska Voglajna. S klikom nanjo se na karti obarva v rumeno barvo. V bližini izgradnje skladišča se nahajajo hidrografska območja, stopnjo tveganja pa preverimo še na podlagi drugih podatkov (glej Sliko 7.32).

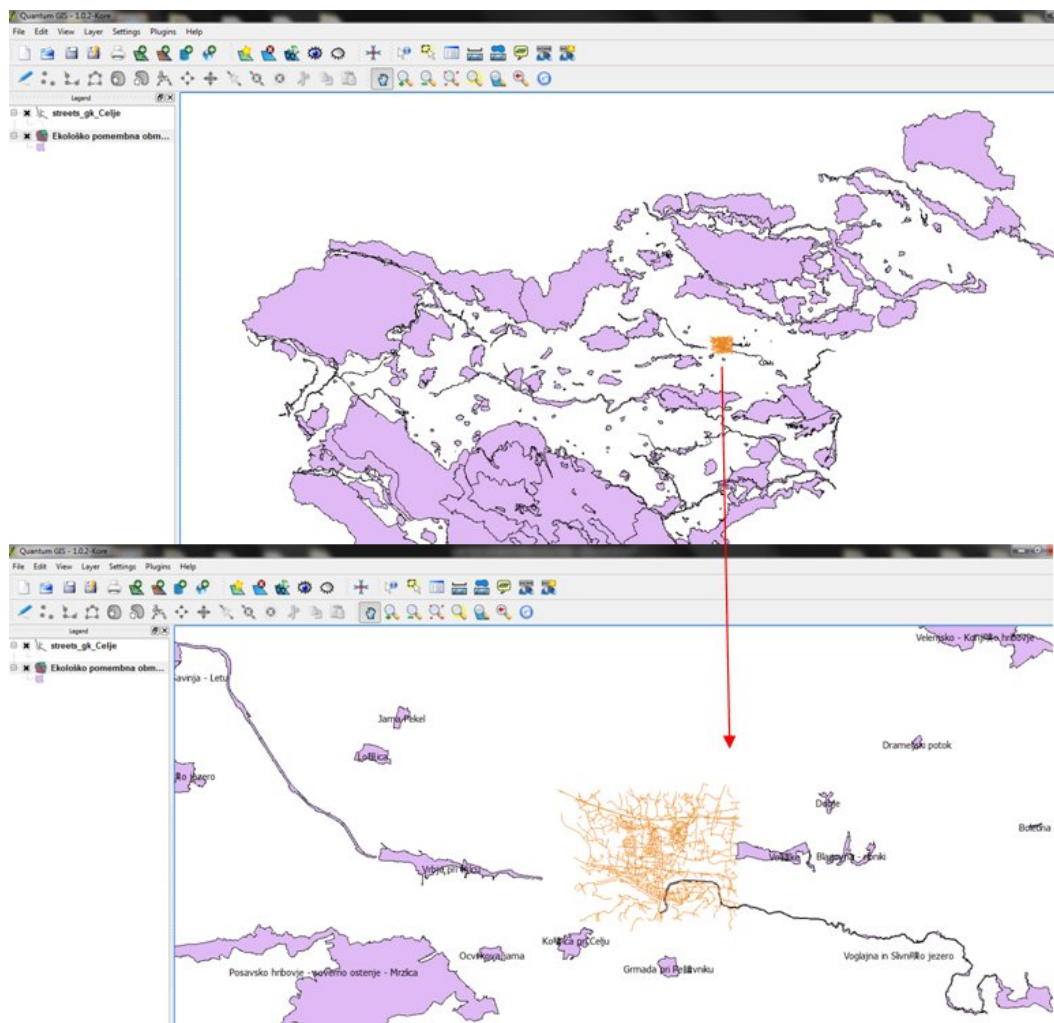
Karta potresne nevarnosti Slovenije je karta projektnega pospeška tal za povratno dobo 475 let, ki je izdelana v skladu z zahtevami evropskega predstandarda Eurocode 8 (EC8). Uporablja se skupaj s slovenskim predstandardom Eurocode 8 - Projektiranje potresno odpornih konstrukcij, ki ga je izdal Slovenski inštitut za standardizacijo (2001). Ozemlje Slovenije je razdeljeno na območja, v katerih se potresna nevarnost v skladu z EC8 ne spreminja. Vrednosti projektnega pospeška tal so zato razvrščene v razrede, zgornja vrednost vsakega razreda pa je pripisana ustreznemu območju. Območje MOC zavzema vrednost projektnega pospeška, ki znaša 0,15. Glede na legendo se nahajajo vrednosti od 0 do 0,25 [g], kar pomeni, da MOC ni izpostavljena velikim potresnim nevarnostim (glej Sliko 7.33).

Pregledamo še karto potresnih nevarnosti skupaj s številom potresov v MOC. Na karto vstavimo podatke potresov (*Katalog potresov*), kjer imamo možnost izbirati med posameznimi izpisi podatkov. Iz karte razberemo, da so se na območju MOC v okolici Medloga, leta 1982 stresla tla. Tla so se tresla še leta 1833 in 1906 na območju Lisc ter leta 1898 na Gradnikovi ulici. Stopnja tveganja ni visoka (glej Sliko 7.34).

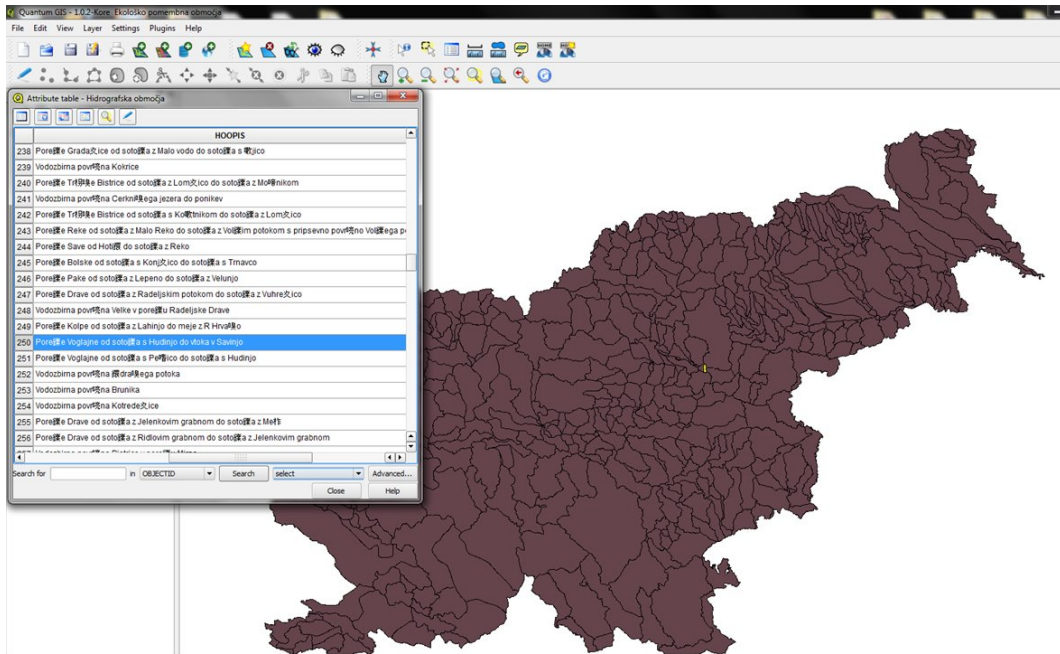
Pomemben element, ki ga je prav tako potrebno upoštevati pri postavitvi skladišča je izpostavitvev tveganju, ki ga povzroča delovanje vetra. Zanima nas torej, ali je območje Teharja izpostavljeno veliki stopnji tveganja tovrstnih dejavnikov. Na karto dodamo sloj ARSO *Projektne hitrosti vetra*. Atributna tabela prikazuje, da se območje Celja nahaja v tako imenovani Coni 1, kjer je hitrost vetra v povprečju od 20 do 30 m/s (glej Sliko 7.35).

Za podrobnejši pregled količine padavin dodamo sloj *Povprečne letne višine korigiranih padavin 1997-2000*. Na območju MOC je v povprečju 1300 mm padavin na leto (glej Sliko 7.36).

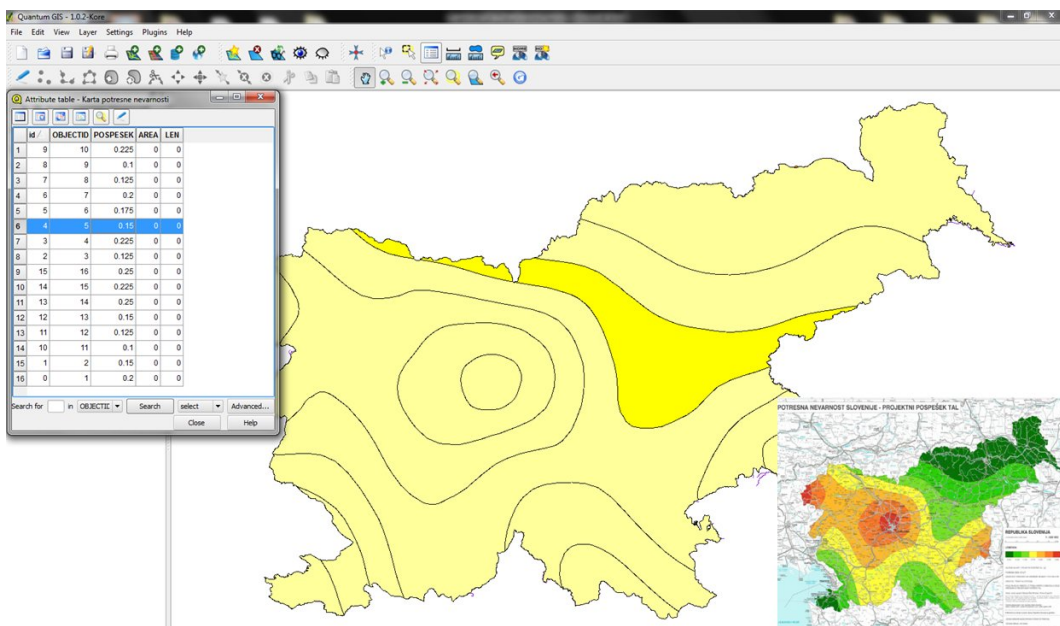
Karta *Onesnaženosti zunanjega zraka* prikazuje območja onesnaženost zraka v Sloveniji. Območje MOC se nahaja v coni SI 2, kar pomeni, da je delež SO₂ med zgornjim pragom za ocenjevanje in mejno vrednostjo. Na karto dodamo še



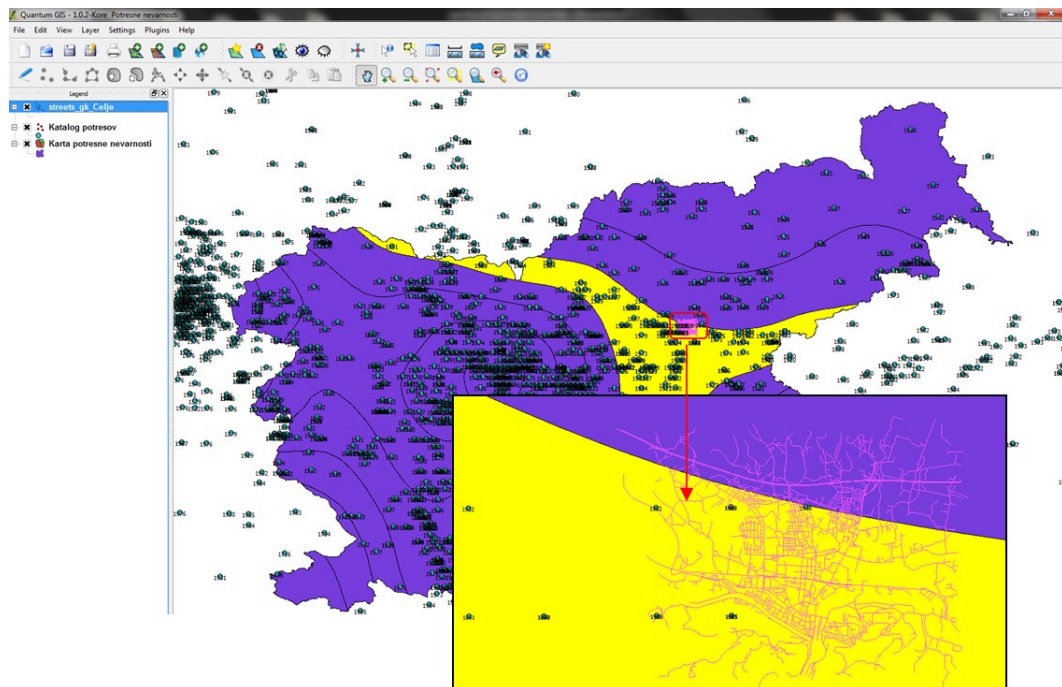
Slika 7.31: Ekološko pomembna območja



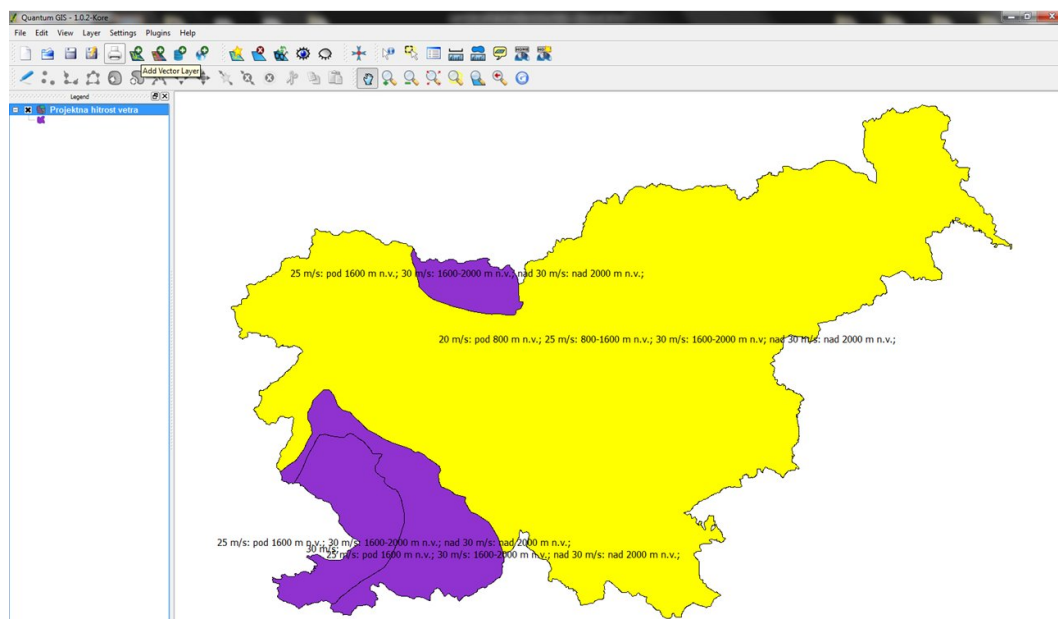
Slika 7.32: Hidrografska območja



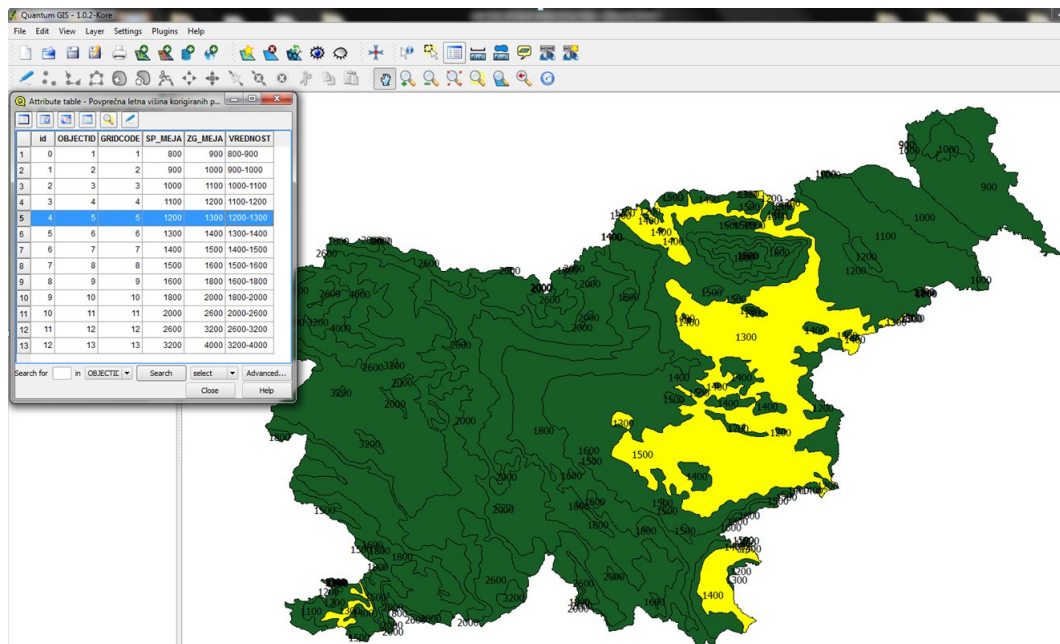
Slika 7.33: Karta potresne nevarnosti - projektni pospešek



Slika 7.34: Karta potresne nevarnosti - splošno



Slika 7.35: Hitrost vetra

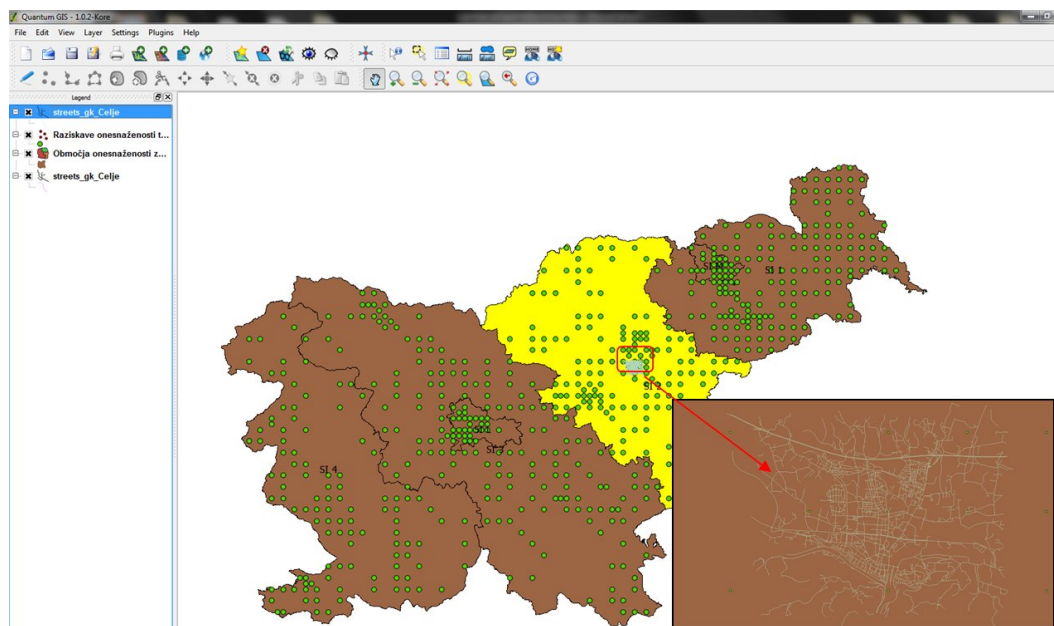


Slika 7.36: Povprečna letna višina korigiranih padavin

sloj izvedenih raziskav onesnaženosti tal Slovenije. Iz Slike 7.37 je razvidno, da so bile meritve opravljene tudi na območju MOC. Za podrobnejše informacije odpremo atributno tabelo z vsemi pripadajočimi podatki (Glej Sliko 7.38).

V nadaljevanju predstavimo tveganja, ki se nanašajo na poplavna območja v RS. S strani ARSO pridobimo podatke o katastrofalnih, pogostih in redkih poplavnih območjih. Na karto dodamo še sloje vodotokov v RS. Iz Slike 7.39 je razvidno, da se v bližini bodočega namišljenega skladišča na Teharju, nahaja pas redkega poplavnega območja (svetlo zelena barva). Tveganju pogostih (svetlo modra barva) in katastrofalnih (temno zelena barva) poplav na danem območju nismo izpostavljeni.

Za zaključek prikažemo še območje hrupa. S spletne strani ARSO pridobimo podatke o hrupu, ki ga povzroča železnica (temno zelena barva). Ugotovimo, da se območje Teharja nahaja na hrupnem območju - hrup povzroča železniški promet (glej Sliko 7.40).



Slika 7.37: Območje onesnaženosti zunanjega zraka

Quantum GIS - 1.0.2-Kore

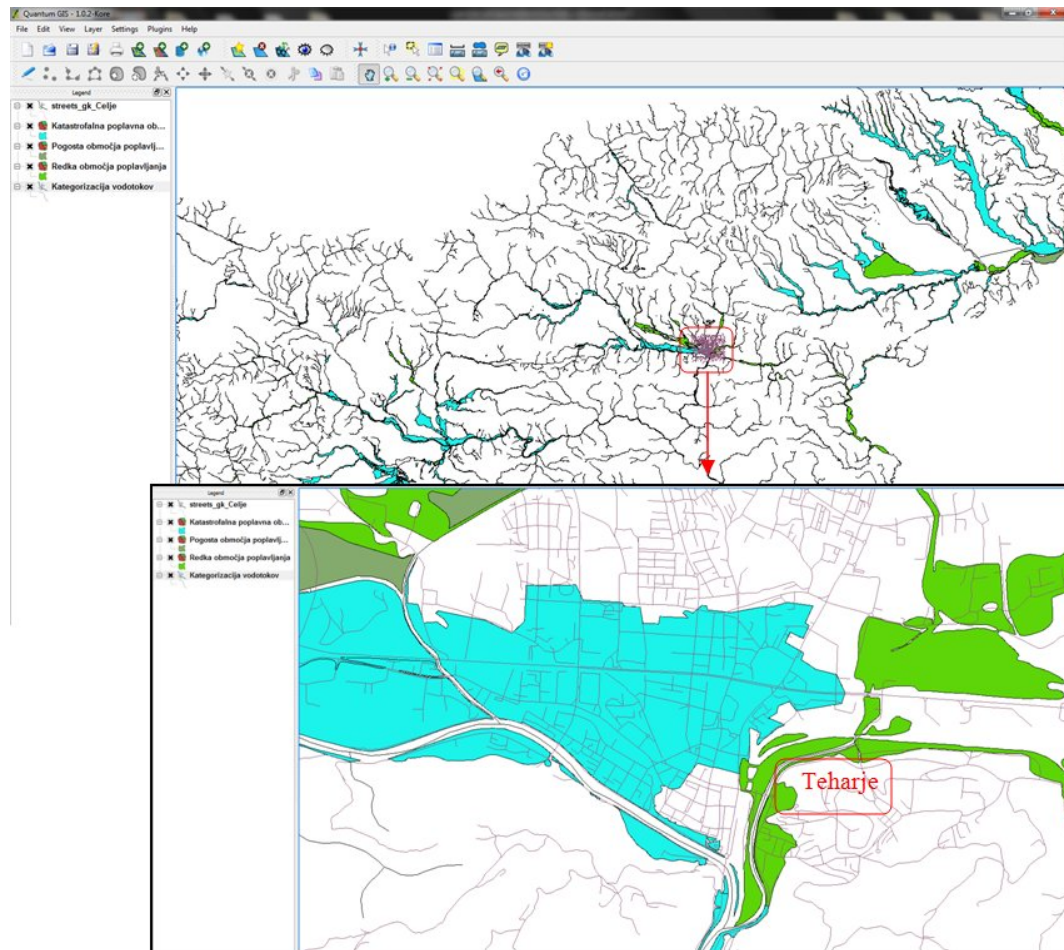
File Edit View Layer Settings Plugins Help

Attribute table - Območja onesnaženosti zunanjega zraka

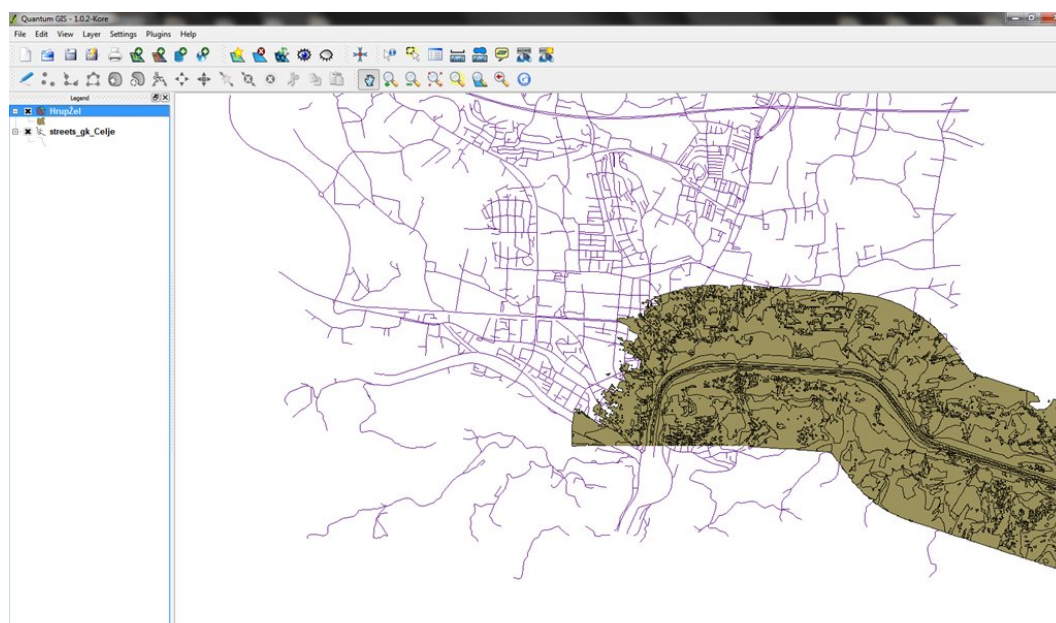
id	OBJECTID	CONA	IME_CONE	SO2
1	0	SI 1	Panonsko območje	pod spodnjim pragom ocenjevanja
2	1	SI 2	Alpsko in panonsko območje	med zgornjim pragom za ocenjevanje in mejno vrednostjo
3	2	SI 3	Alpsko-Dinarsko območje	pod spodnjim pragom ocenjevanja
4	3	SI 4	Sredozemsko območje	pod spodnjim pragom ocenjevanja
5	4	SI L	Aglomeracija Ljubljana	pod spodnjim pragom ocenjevanja
6	5	SI M	Aglomeracija Maribor	pod spodnjim pragom ocenjevanja

Search for in OBJECTID Search select Advanced... Close Help

Slika 7.38: Atributna tabela za območje onesnaženosti zunanjega zraka



Slika 7.39: Poplavna območja



Slika 7.40: Hrup, ki ga povzroča železniški promet

Povzetek

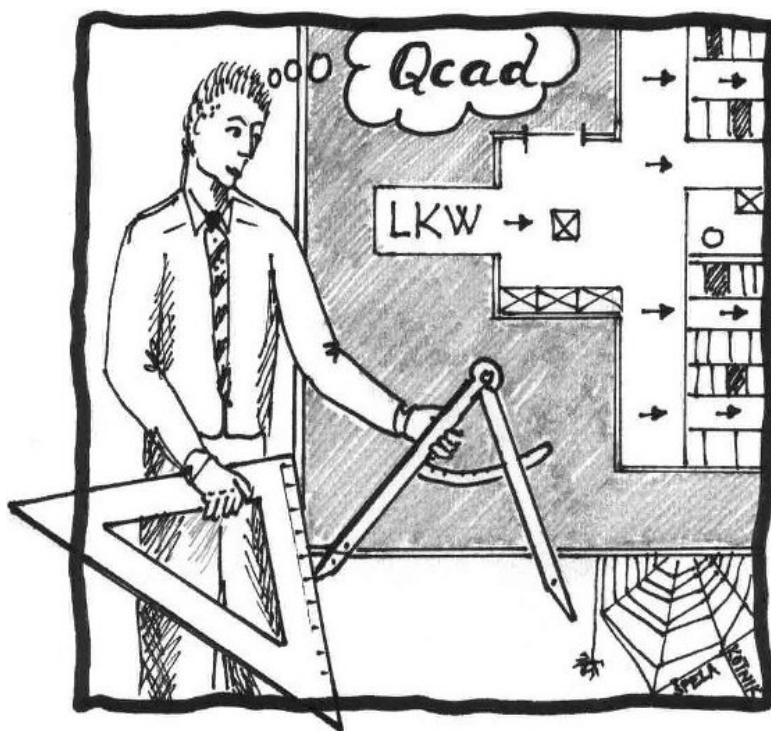
Quantum GIS (QGIS) je uporabniku prijazen odprtokodni geografski informacijski sistem (GIS) pod licenco GNU General Public License. Deluje na različnih platformah (Linux, Unix, Mac OSX in Windows) ter podpira številne vektorje, raste, podatkovne baze, formate in funkcionalnosti.

Z uporabo programskega orodja Quantum GIS predstavimo izbiro skladiščne lokacije v MOC na območju Teharja, z upoštevanjem tveganj (hrup, onesnaženost, poplave, potresi ipd.). Podatke pridobimo s strani različnih inštitucij - ARSO, MOC in SURS. Rezultati raziskave pokažejo, da bi bila lokacija na Teharski cesti primerna za izgradnjo skladišča, saj je stopnja tveganja nizka.

Pri opisu programskega orodja Quantum GIS smo uporabili še dodatne vire in literaturo: [2] [107] [54].

Poglavje 8

QCAD - 2D prostorsko načrtovanje



CAD orodja
Računalniško načrtovanje
Primer: 2D načrtovanje skladiščnega prostora

8.1 Teoretično ozadje

Na trgu obstaja veliko programov za tehnično risanje skic. Mnoga so plačljiva in zaradi visokih cen nedostopna za podjetja. Pomembno je, da znamo s pomočjo brezplačnih programskih orodij narisati skice, ki omogočajo prikaz zamišljenih stvari. Tako lahko v podjetju prikažemo zamisli za nadaljnji razvoj, ki jih s pomočjo tovrstnih programskih orodij enostavno in brez dodatnih stroškov narišemo.

8.1.1 Računalniško podprto načrtovanje CAD

CAD je kratica za Computer-aided design, kar pomeni računalniško podprto načrtovanje. Uporablja se za računalniško načrtovanje realnih ali navideznih objektov. CAD omogoča načrtovanje linij in oblik v 2D in 3D prostoru. Njegova uporaba je razširjena predvsem na tehničnih področjih, kjer je kot orodje nepogrešljiv, saj bistveno pripomore k izboljšanju kakovosti dela oz. končnih izdelkov. Dandanes si je težko predstavljati proizvodnjo in oblikovanje izdelkov brez pomoči programskih orodij CAD. Najpogosteje ga uporabljajo razni oblikovalci, arhitekti, projektanti [58].

Logistika kot dejavnost se ukvarja z upravljanjem različnih tokov v oskrbni verigi ali znotraj podjetja. Pri tem zajema vse procese (nabava, načrtovanje, napovedovanje, skladiščenje, transport), ki se izvajajo. Pri obvladovanju le teh potrebuje različna znanja. Mednje umestimo tudi poznavanje in uporabo CAD orodij, katera pripomorejo k vizualizaciji problema, prostorov ali procesov.

CAD orodja logistu omogočajo:

- načrtovanje procesov (skladišče, transport, proizvodnja);
- načrtovanje logističnih objektov in sredstev;
- 2D in 3D modeliranje;
- optimizacijo transportnih poti v objektih (integracija z drugimi programskimi orodji);
- integracija procesov.

CAD standardi

Standardi CAD so ustvarjeni za izboljšanje produktov in izmenjavo CAD datotek med različnimi CAD orodji. Zajemajo različna področja načrtovanja, kot so [57]:

- CAD sloji (različni standardi: BS 1192, ISO 13567, A/E/C CADD Standard, A-B374-E, AIA);
- debelina črte (npr. za sivo barvo 0,13 mm, za rdečo barvo 0,25 mm);
- dimenzija in stil besedila (npr. višina in debelina besedila, stil oblikovanja besedila);
- merila (npr. 1:1, 1:10 itd.);
- imenovanje datotek (različni standardi: BS 1192, AIA, AEC itd.);
- geometrija, dimenzije in toleranca (različni standardi: ASME Y14.41-2003, ISO 1101, VDA 4955).

Osnovni koncepti CAD orodij

Vsa CAD orodja vsebujejo osnovne koncepte, in sicer: subjekti, atributi, plasti, bloki, priprava v CAD, koordinatni sistem, urejanje, pogled v CAD.

V CAD orodjih so subjekti grafični objekti. Tipični subjekti so točka, črta, krog, posebni pa besedila, mere. Vsak subjekt ima svoje attribute, barvo, tip črte, debelino črte. Poleg tega je vsak načrtovani subjekt na svojem sloju, vsak sloj pa lahko ima več subjektov. Vsak sloj ima tudi svoje attribute, katere prevzamejo subjekti na sloju. Blok sestavlja skupina subjektov, le ta pa je lahko večkrat vstavljen na različne lokacije z različnimi atributi. Takšne bloke imenujemo *Inserts*.

CAD načrtovanje je podobno tradicionalnem načrtovanju brez uporabe računalniške tehnologije. Primer: ko rišemo na papir vsekakor uporabimo ravnilo ali katero drugo orodje za načrtovanje; tako je tudi pri uporabi CAD orodji, ki ponujajo številna orodja za doseg istega cilja. Prednost CAD orodij je, da izdelek popravljamo in dopolnjujemo, kar je na papirju težje izvedljivo. Za enostavnejšo uporabo CAD orodij je priporočljivo poznavanje koordinatnega sistema. CAD orodja uporabljajo kartezijski koordinatni sistem (X in Y koordinatna os) in polarni koordinatni sistem (razdalja in kot).

Urejanje v CAD orodjih je enostavno. Subjekte dodajamo z različnimi orodji ali dupliciranjem. Narisati subjekt pomeni določiti vse točke v koordinatnem sistemu. Na enostaven način odstranimo ali modificiramo subjekt. Pogled v CAD orodjih je odvisen od izbranega merila, npr. merilo 1:10 pomeni, da je en centimeter na risbi enak 10 centimetrom v realnosti. Izbira merila je predvsem pomembna pri tiskanju načrtov.

8.2 O programskem orodju

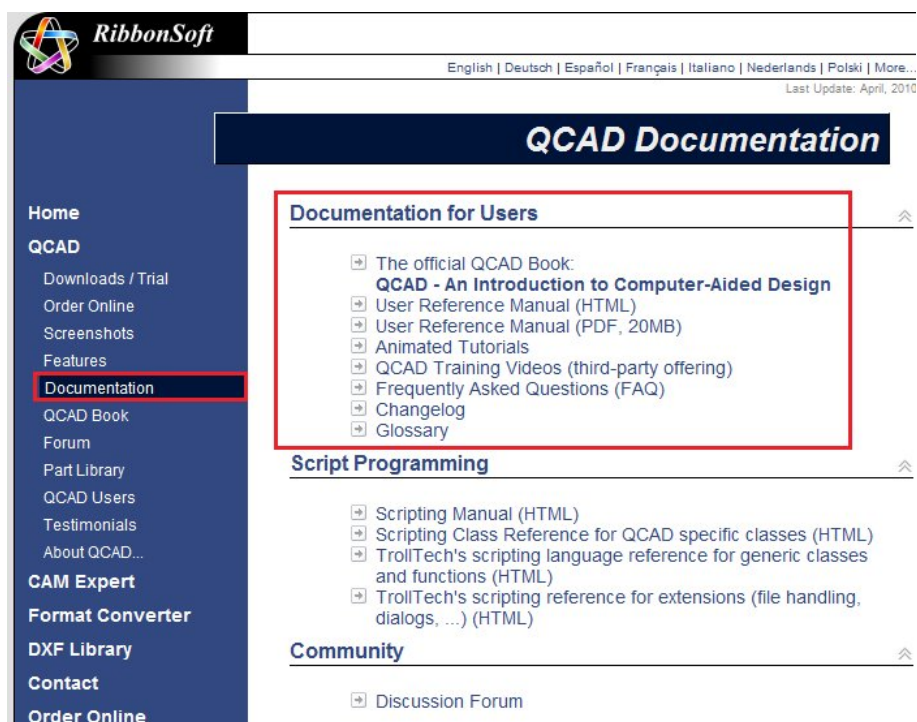
Qcad je računalniško podprto orodje za 2D načrtovanje in risanje. Začetki razvoja segajo v leto 1999, ko je programsko orodje nastalo kot rezultat spinoff projekta izdelave CAD sistema. Z njim izdelamo tehnične risbe (načrti zgradb, njihovih notranjosti, mehanski deli, sheme, diagrami ipd.). Uporaben je na različnih tehniških področjih: strojništvo, lesarstvo, gradbeništvo, arhitektura, geodezija in elektrotehnika. Programsko orodje Qcad deluje na vseh operacijski sistemih (Windows, MAC OS, Linux (Ubuntu, Fedora . . .), Unix) [45].

Qcad je zasnovan z željo po modularnosti, razširljivosti in prenosljivosti, vendar je za uporabnika najpomembnejši njegov uporabniški vmesnik. Je preprosto 2D orodje pripravljeno za takojšnjo uporabo, saj ne potrebujemo nikakršnih dodatnih CAD znanj in izkušenj.

Prenos in namestitev

Qcad namestimo iz uradne spletne strani RibbonSoft Qcad [43], organizacije, ki razvija in distribuira izdelke kot so Qcad, CAM Expert, vec2web itd. Obstajajo različne verzije Qcad, edina odprtokodna in prosto dostopna verzija je *Qcad cummunity edition*. Ostale verzije so plačljive, brezplačne so le demo verzije s časovno omejitvijo uporabe. Programsko orodje je dostopno v številnih svetovnih jezikih. Izvorna koda verzije Qcad cummunity edition je izdana pod GNU GPL licenco (odprtokodna rešitev). To verzijo je možno uporabljati kot uporabnik Linux, Unix ali MAC OS operacijskega sistema. Uporabniki operacijskega sistema Ubuntu lahko preko *Synaptic* enostavno poiščejo, prenesejo in namestijo programsko orodje Qcad.

Na forumu spletne strani programskega orodja Qcad je možno poiskati pomoč in priročnik za njegovo uporabo. Za prikaz in enostavno uporabo zadostuje že pomoč, ki jo najdemo v menijski vrstici programa (Slika 8.1).



Slika 8.1: Spletna pomoč za uporabo programskega orodja Qcad

Problem

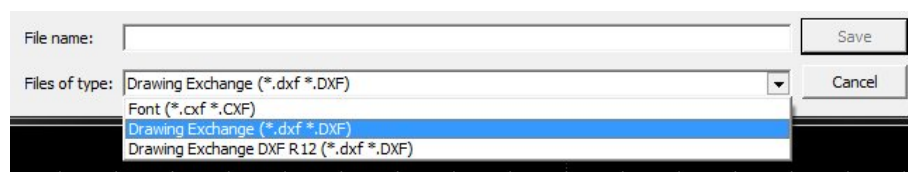
S pomočjo Qcad narišemo skico skladišča, v katerem so uskladiščene pnevmatike, platišča in matice za pritrjevanje platišč. V programskem orodju narišemo skico skladišča, na kateri prikažemo vse potrebne lokacije od sprejema materiala v skladišče do odpreme materiala v proizvodnjo. Pri samem risanju se osredotočimo predvsem na logistični vidik, lokacijo za raztovarjanje in odpremljanje materiala, lokacijo za transportna vozila, transportne poti znotraj skladišča, način uskladiščenja materiala, lokacijo postavitve regalov, lokacijo postavitve pisarne in polnilne postaje. Način uskladiščenja materiala je FIFO (*first in, first out*) sistem, kar pomeni, da se material, uskladišči po točno določenem vrstnem redu. Material, ki se prvi uskladišči, se tudi prvi odpremi v proizvodnjo. Pri skladiščenju zalog je potrebna podpora informacijske tehnologije. Skladiščenje je pomembno področje v podjetju, kateremu je potrebno nameniti veliko pozornosti in hkrati upoštevati vse faktorje, na podlagi katerih določimo optimalen način skladiščenja. V našem primeru izpostavimo in prikažemo zgolj dva faktorja (lokacija in način uskladiščenja materiala).

8.3 Uporaba

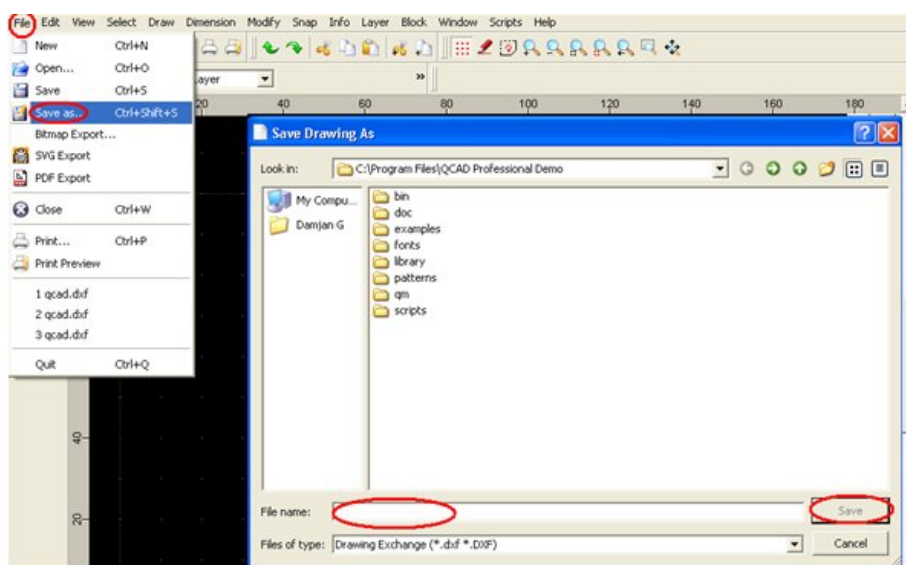
V nadaljevanju prikažemo in opišemo uporabo orodij, ki jih uporabimo za izdelavo skice skladišča. Poudariti je potrebno, da ne uporabimo vseh možnih orodij, ki jih ponuja program Qcad, ampak le tiste, ki jih potrebujemo za izdelavo skice skladišča.

Po zagonu programskega orodja smiselno ustvarimo datoteko kamor shranimo narisano skico. V menijski vrstici kliknemo *File*, nato poiščemo in kliknemo na razdelek *Save As*. Odpre se novo okno v katerega zapišemo ime datoteke in jo nato shranimo na želeno mesto. Datoteko lahko shranimo kot *Font* (.cfx), *Drawing Exchange* (.dxf) in *Drawing Exchange DXF 12* (.dxf) (Slika 8.2 in 8.3).

Skico lahko kadarkoli poiščemo in naložimo v programsko orodje. V orodni



Slika 8.2: Izbira oblike shranitve datoteke



Slika 8.3: Shranjevanje datoteke s skico

vrstici poiščemo in kliknemo *Open* in odpre se novo okno. Poiščemo shranjeno skico, ki jo z ukazom *Open* odpremo (Slika 8.4).

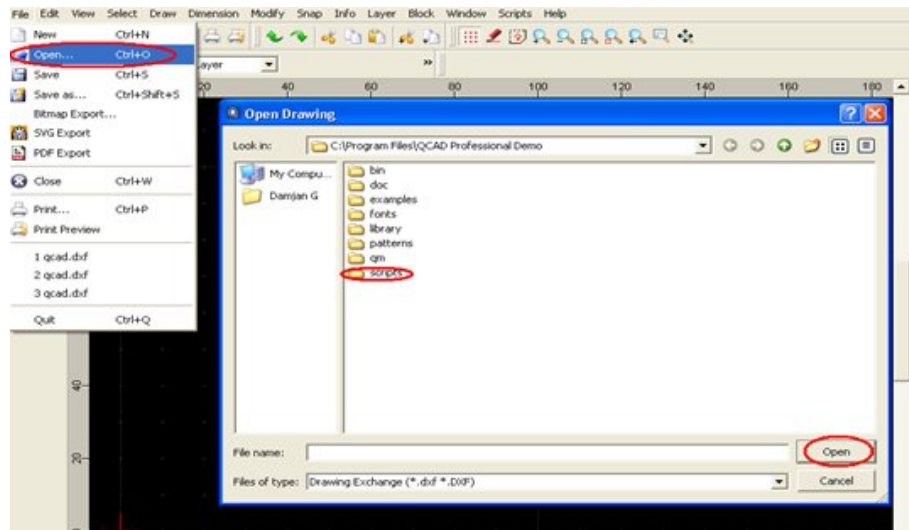
Zaključene projekte oz. tehnične risbe s klikom na meni *File* izvozimo v različnih oblikah. Na voljo so možnosti: *.bmp*, *.svg* in *.pdf* (Slika 8.5).

Preden začnemo z risanjem skic, naredimo seznam blokov, katerim poljubno določimo barvo, debelino in način prikaza posamezne črte, ki jo kasneje uporabimo za risanje skic. V desnem kotu na seznamu *Layer List* med orodji kliknemo znak + pod katerim se odpre okno, kjer določimo prej omenjene attribute (Slika 8.6).

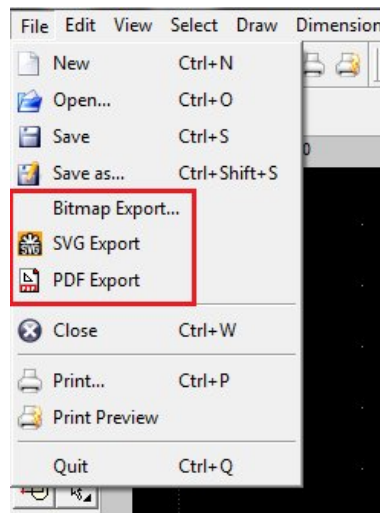
Programsko orodje omogoča povečavo, pomanjšavo in premik posameznih delov skice. Ta funkcija omogoča, da smo pri risanju skice natančni, saj se lahko osredotočimo na posamezni del. Vklopimo jo z izbiro zelenega ukaza v orodni vrstici, na Sliki 8.7 je označen z rdečo barvo.

Enak postopek lahko izvedemo še hitreje in sicer z uporabo miške, kot je prikazano na Sliki 8.8, z vrtenjem miškega kolesčka. Z vrtenjem naprej povečujemo, z vrtenjem nazaj pa pomanjšujemo skico. S pritiskom na kolesček pomikamo skico v zeleno smer.

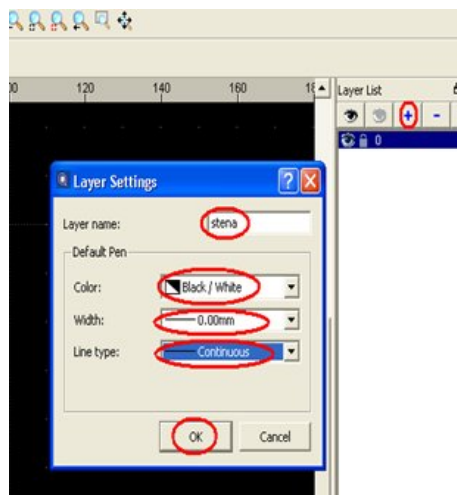
Posamezne linije oz. črte je možno zakleniti, tako da jih kasneje ne moremo spreminjati oz. zbrisati iz skice. To storimo tako, da v orodni vrstici kliknemo na *Ključavnico*. V primeru, ko imamo skupaj narisanih več črt, s klikom na *Oko* v orodni vrstici vidimo, kje se nahaja določena črta na skici. V primeru, da je ne želimo videti jo lahko izbrišemo. Zbrisane črte nato prikažemo s ponovnim



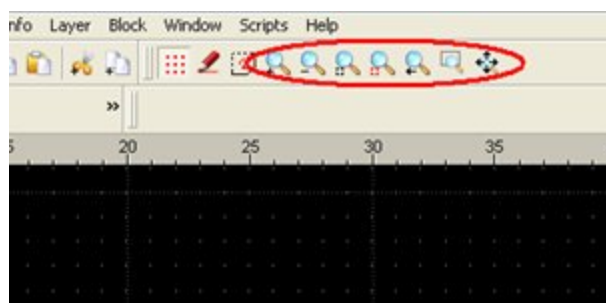
Slika 8.4: Odpiranje datoteke s skico



Slika 8.5: Izvažanje datotek



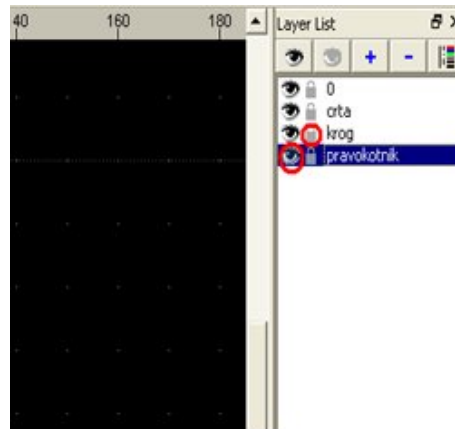
Slika 8.6: Dodajanje novih blokov



Slika 8.7: Možnost povečave



Slika 8.8: Povečava slike z uporabe miške



Slika 8.9: Zaklepanje seznama

klikom na *Oko* (Slika 8.9).

Qcad omogoča risanje različnih oblik. V orodni vrstici označeni z rdečo barvo, izberemo poljuben način za risanje črte. Izbiramo med ravnimi črtami, krogi, elipsami, krivuljami in pikicami, odvisno od tega, kakšen predmet želimo narisati. Znotraj posameznega orodja so dostopni še drugi načini za risanje črt, ki se navezujejo na osnovni način oblikovanja črte (Slika 8.10).

Skica je lahko narisano natančno ali nenatančno. Ko rišemo nenatančno izberemo v orodni vrstici + in poljubno rišemo črte, kot je prikazano na Sliki 8.11.

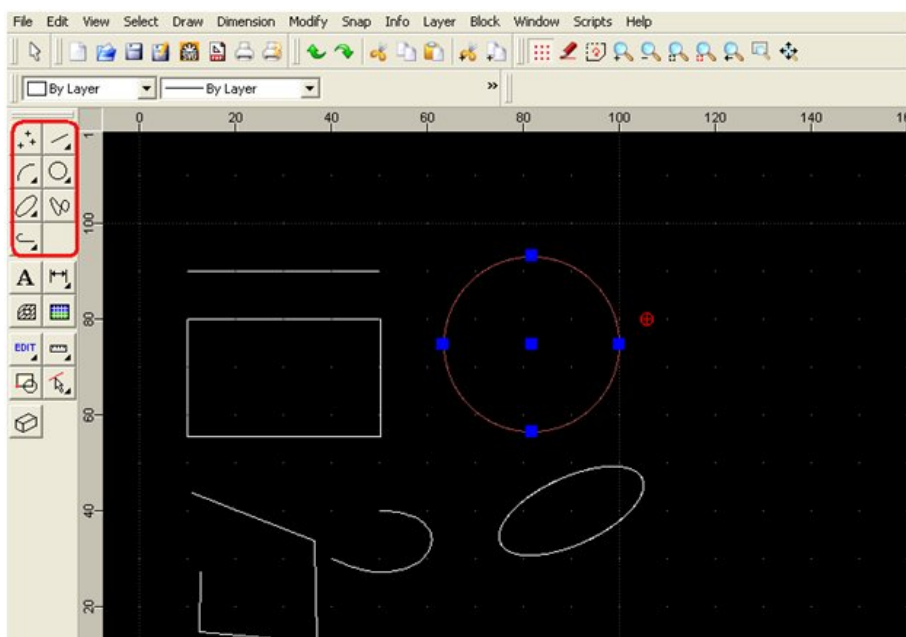
Qcad omogoča risanje skic z visoko natančnostjo. Programsko orodje pri risanju skice pomaga določiti koordinate poteka črt (Slika 8.12).

Razdalje med posameznimi črtami oz. dolžine posameznih črt izmerimo v poljubni merski enoti. Kliknemo na *Ravnilo* kot je prikazano na Sliki 8.13 in izberemo način merjenja razdalje. Na skici se izpiše določena razdalja oz. dolžina.

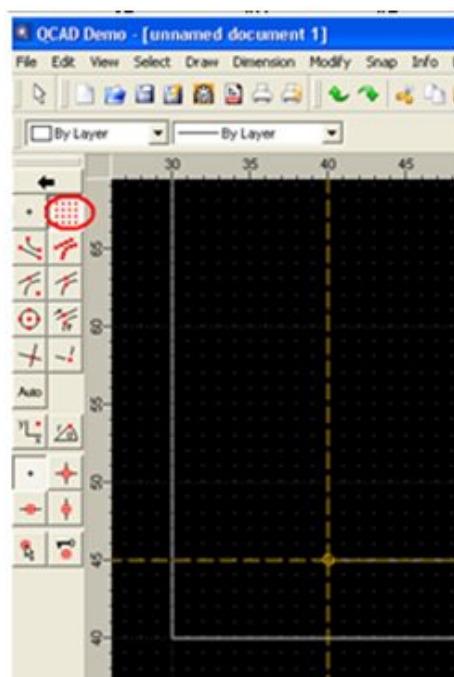
Na skico je možno dodati tudi tekst, ki ga poljubno oblikujemo. V orodni vrstici izberemo *A*, odpre se novo okno v katerem določimo pisavo, velikost pisave, postavitev besed ter še nekaj drugih funkcij. Enostavno vpišemo tekst in ga dodamo na skico (Slika 8.14).

V primeru, ko se pri risanju skice zmotimo, lahko zberemo posamezne črte. Črto, ki jo želimo zbrisati, označimo in kliknemo *Edit* in poiščemo *Radirko* (Slika 8.15).

Obstaja tudi možnost vrnitve v predhodno stanje. V primeru napake se s klikom na *Zeleno puščico* vrnemo na predhodno stanje, prav tako pa se lahko vrnemo tudi nazaj v trenutno stanje. Postopek prikazuje Slika 8.16.



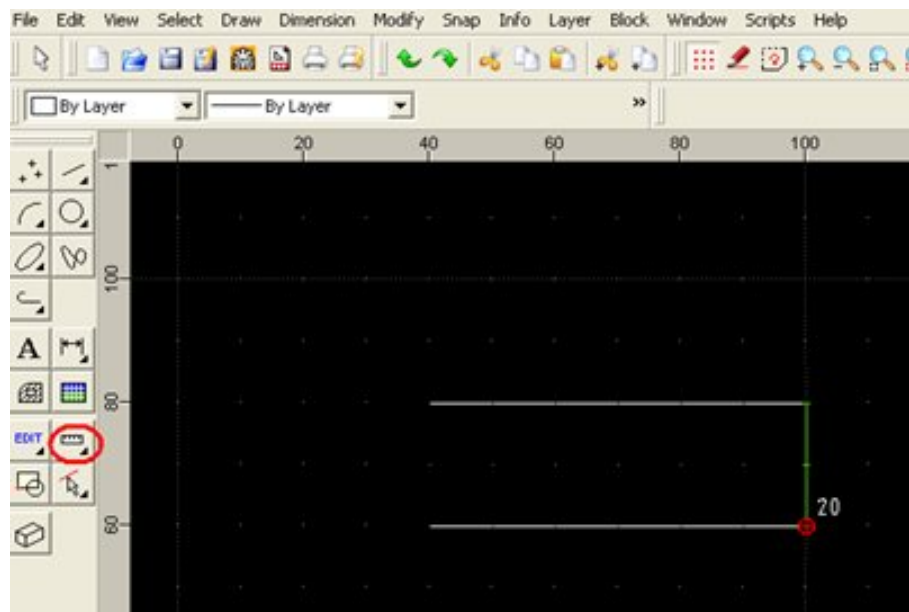
Slika 8.10: Izbira oblike črte



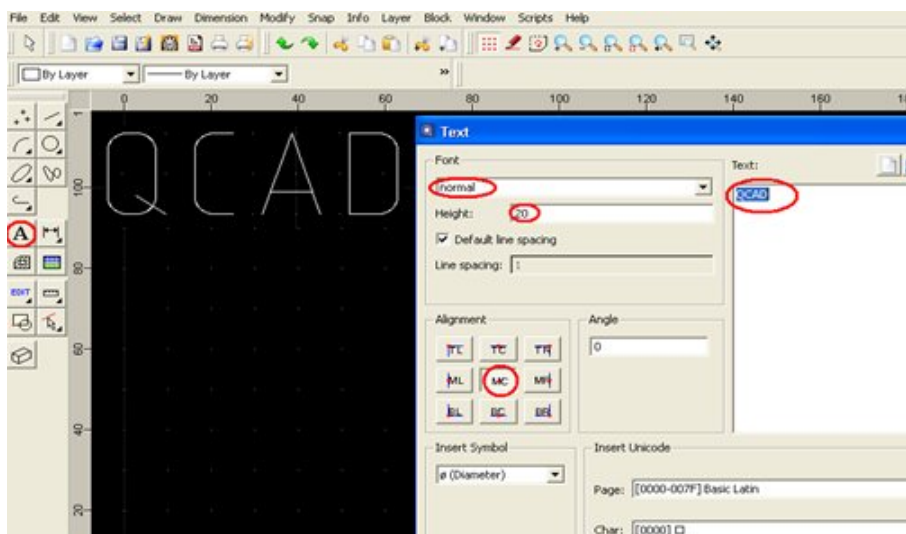
Slika 8.11: Natančnost risanja



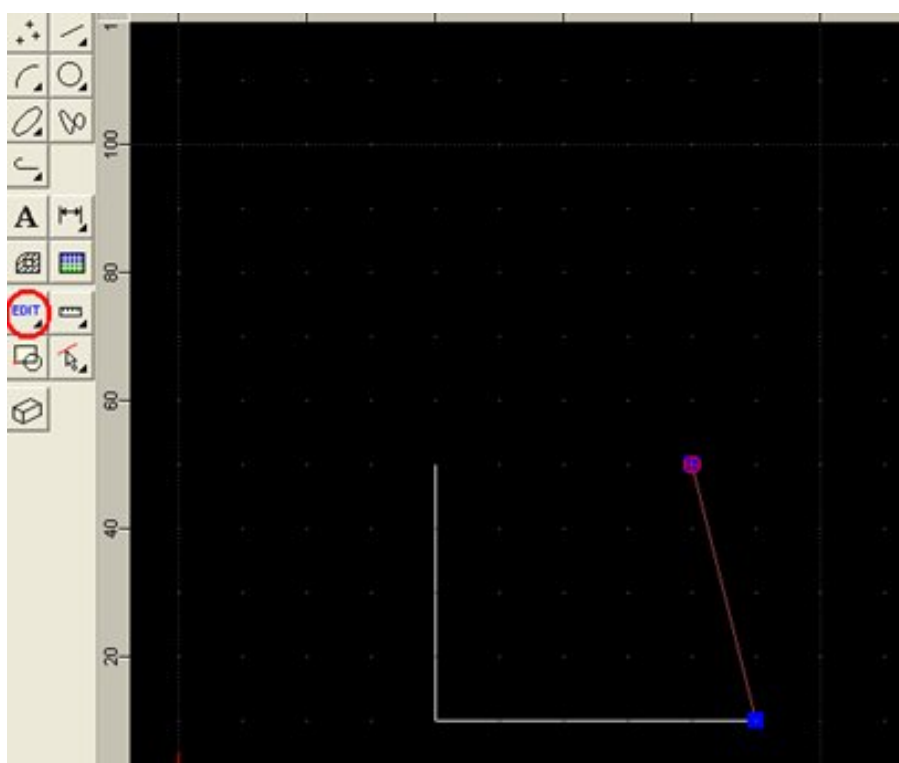
Slika 8.12: Natančnost risanja (točke)



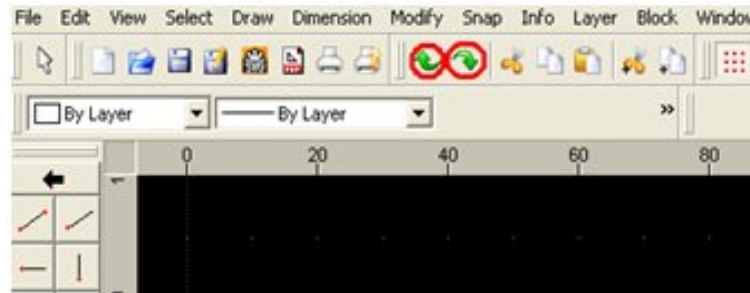
Slika 8.13: Merjenje razdalj



Slika 8.14: Dodajanje teksta na skico



Slika 8.15: Brisanje črt



Slika 8.16: Razveljavitev ukaza



Slika 8.17: Pomoč

Najpomembnejše pri programskem orodju je, da omogoča pogled v priročnik - pomen orodja in njihova uporaba. Na takšen način še podrobneje spoznamo kakovost programskega orodja. Pot do priročnika prikazuje Slika 8.17.

Slika 8.18 prikazuje skico skladišča, kjer so lokacije označene s številkami, katerih pomen je zapisan z besedami v legendi poleg skice.

Število 1 na Sliki 8.18 prikazuje transportno vozilo, ki dobavlja material v določenih količinah in po določenih dobavnih rokih. Pod številom 2 prikazujemo električne viličarje, ki omogočajo manipulacijo z materialom. Na lokaciji, ki jo prikazuje število 3 so nameščene prazne palete, ter ostale stvari, potrebne za manipuliranje in uskladiščenje materiala. Vhod in izhod iz skladišča, tako zaposlenih kot tudi viličarjev in ostalih manipulacijskih naprav, prikazuje število 4. Rastovarjanje in natovarjanje materiala prikazuje število 5. Število 6 prikazuje lokacijo regala, kjer je material uskladiščen. Puščice, ki se nahajajo poleg regalov, kažejo smer pomikanja materiala. Vsako skladišče ima tudi pisarno, za vso potrebno dokumentacijo in informacijskim sistemom za vodenje zalog, kar prikazuje število 7. Za manipulacijo blaga v skladišču se uporabljamo električni viličarji, katere je potrebno polniti na polnilni postaji, ki jo prikazuje število 8. Število 9 oz. rumene črte, označujejo transportno pot



Slika 8.18: Skica skladiščnega prostora

viličarjev.

Povzetek

Program Qcad omogoča natančno risanje najrazličnejših tematskih področij v dveh dimenzijah na enostaven način. S programskim orodjem narišemo skladišče uskladiščenih avtomobilskih pnevmatik, platišč in vijakov. Pri risanju skic uporabimo različna orodja. Ta orodja smo s pomočjo "print screenov" prikazali in opisali v samem priročniku.

Pri opisu programskega orodja Qcad smo uporabili še dodatne vire in literaturo: [83].

Poglavje 9

SIMPLE WAREHOUSE MAPPER- 3D simulacijsko orodje



Simulacijsko orodje
Vizualizacija procesa skladiščenja
Primer: postavitev skladiščnega prostora

9.1 Teoretično ozadje

Logistika kot orodje odločanja, bistveno vpliva na izbiro modela poslovnega procesa, od samega začetka nabave do končne izvršitve prodaje. Poslovni procesi, ki se odvijajo v organizacijah morajo biti smiselno zastavljeni, organizirani, vodeni in kontrolirani. Njihovo izvajanje ima vpliv na celotno poslovanje organizacije.

Informacijska orodja za učinkovito izvajanje in kontroliranje poslovnih procesov so le del orodij, ki postavljajo temelje odličnosti. Orodja izbira vodstvo, ki hkrati opredeli tudi sredstva, ki so potrebna za doseg zastavljenega cilja. Želja po minimalizaciji in racionalizaciji poslovnih stroškov teži k izbiri informacijskega orodja, po načelu največ za najmanj, kar pomeni, da moramo izbrati najboljša orodja za najugodnejšo ceno. Takšen način varčevanja nima vedno želenega učinka, saj je včasih potrebno investirati več, da dobimo želeno kvaliteto.

Umetnost pristopa k rešitvi določenega problema je uporaba dostopnih (brezplačnih) informacijskih orodij, ki jih ponuja splet. Informacijska orodja moramo smiselno povezati in uporabiti v kombinaciji še z drugimi orodji. Na takšen način vizualno prikažemo obstoječo problematiko in vizualiziramo rešitev oz. predstavimo model.

Simple Warehouse Mapper je 3D simulacijsko orodje, ki je dostopno brezplačno na svetovnem spletu in omogoča vizualizacijo realnih skladiščnih procesov in materialnih tokov. Program je namenjen za izdelavo grafičnega prikaza proizvodnega in skladiščnega procesa. V kombinaciji s skladiščnim (Warehouse Management System) in poslovnim (Enterprise Resource Planning) informacijskim sistemom nudi odlično podporo za dokumentiranje poslovnih potreb. Program ni zahteven za uporabo in z njim v kombinaciji z informacijskim tokom učinkovito optimiziramo materialne tokove v proizvodnem in skladiščnem procesu [66].

9.1.1 3D vizualizacija prostora

Vizualizacija je tehnika izdelave slik, diagramov, animacij. Pri vizualizaciji se prepleta realni in navidezni svet, prav tako tudi abstraktne in konkretne ideje. Omogoča jasnejšo predstavo same prostorske umestitve objektov. Najpogosteje uporabljamo računalniško podprte vizualizacije, ki lahko z različnimi orodji približajo navidezni svet realnemu. Področja uporabe vizualizacije so različna, najprimernejše je za prostorsko planiranje, gradnjo objektov ipd. S pomočjo vizualizacije lahko prikažemo statistične in dinamične procese. Uporaba vizualizacije je vsekakor priporočljiva pri logističnih procesih in oskrbni verigi, kjer predstavimo prostor in procese. V našem primeru smo se odločili za

enostavno vizualizacijo skladiščnega prostora, s katero prikažemo prostorsko umestitev objektov in predmetov ter materialni tok [64].

9.2 O programskem orodju

Simple Warehouse Mapper je enostavno 3D simulacijsko orodje za ustvarjanje skladiščnih modelov. Program uporablja preprosto 3D simulacijsko tehniko, znano kot izometrični zemljevid, zagotavlja pa realističen pogled na distribucijski center. S *Simple Warehouse Mapper* ustvarimo vizualno podobo objekta s pripadajočimi podatki.

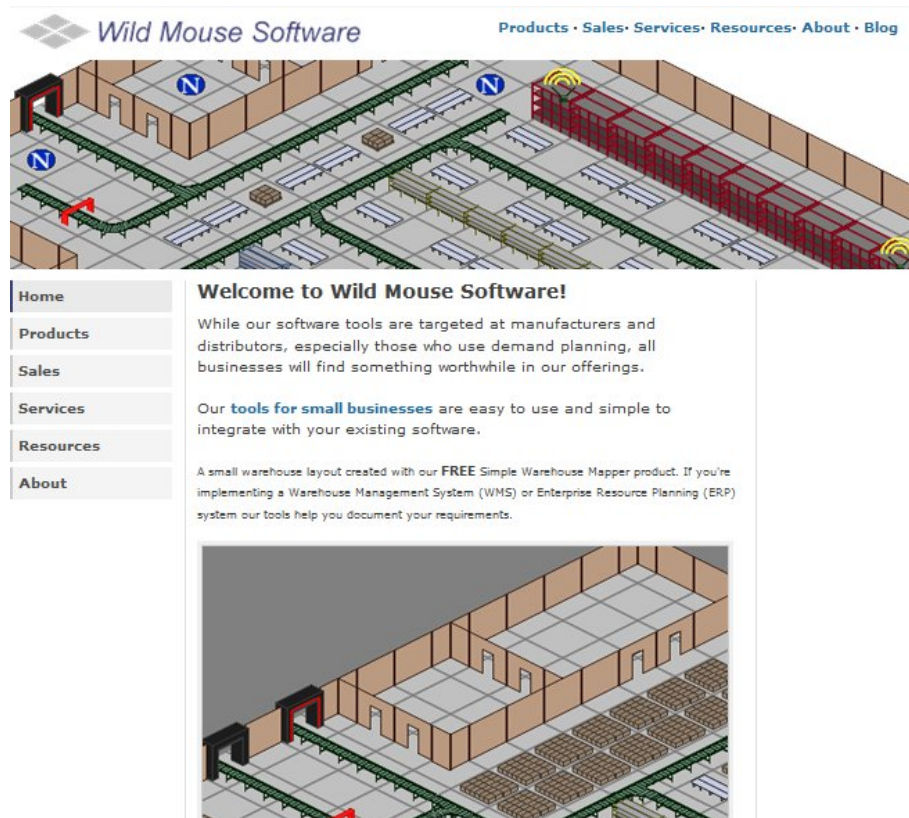
Vizualizacija objekta ali določenega problema v nekaterih situacijah poda hitro rešitev, saj človek dokazano hitreje reši problem s pomočjo slike kot besedila. Z grafičnim prikazom prav tako hitreje ugotovimo in identificiramo morebitne napake in zastoje v procesu.

Prenos in namestitvev

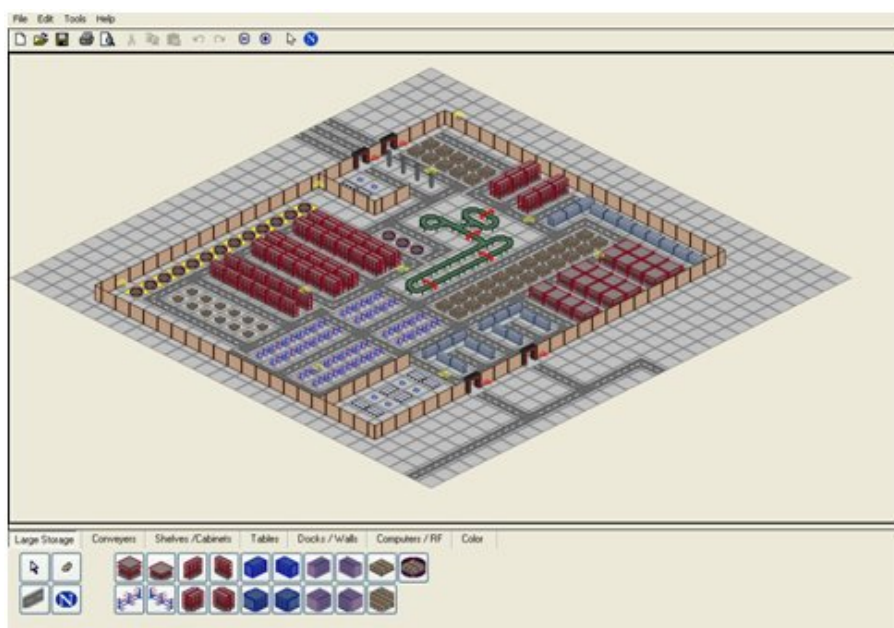
Programsko orodje *Simple Warehouse Mapper* je brezplačno orodje, ki je dostopno na spletnem naslovu Wild Mouse Software. Prenos programskega orodja je preprost. Na spletni strani poiščemo razdelek *Download* (Slika 9.3) in pričnemo s postopkom prenosa in namestitve. Razdelek *Download* najdemo v meniju *Products* na levi strani uvodne spletne strani. Programsko orodje je narejeno za okolje Windows (Windows Vista, Windows XP, Windows 2000), vendar ga lahko s pomočjo orodja Wine prenesemo in uporabljamo tudi v Ubuntu.

Problem

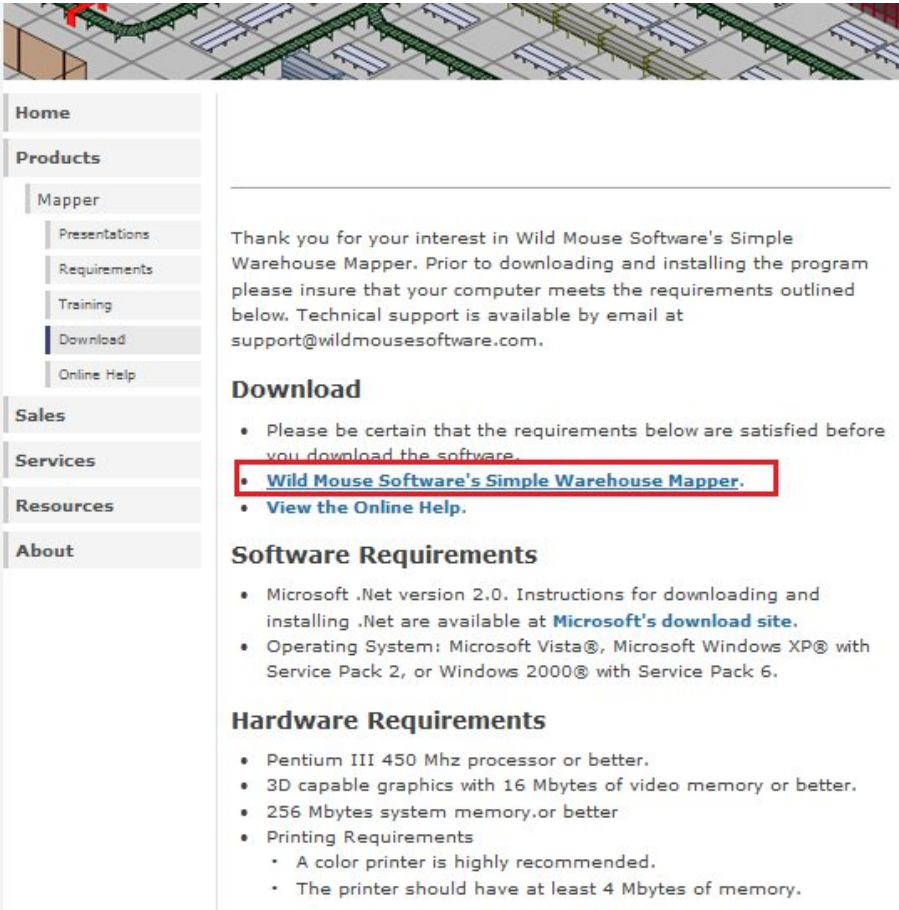
V namišljenem podjetju je zaradi cilja optimizacije stroškov poslovanja, predvsem skladiščnih in transportnih, potrebno pripraviti 3D vizualizacijo procesa skladiščenja. S pomočjo 3D vizualizacije prostora in procesov bodo določene pomanjkljivosti hitreje identificirane in lažje odpravljene. Potrebno je pripraviti temeljit posnetek obstoječega stanja, predvsem skladiščnega prostora in skladiščnih procesov. Posnetek stanja opravimo na terenu in s pomočjo pedhodno pridobljenih gradbenih načrtov stavbe.



Slika 9.1: Spletna stran Simple Warehouse Mapper



Slika 9.2: Programsko orodje Simple Warehouse Mapper



Home

Products

- Mapper
- Presentations
- Requirements
- Training
- Download
- Online Help

Sales

Services

Resources

About

Thank you for your interest in Wild Mouse Software's Simple Warehouse Mapper. Prior to downloading and installing the program please insure that your computer meets the requirements outlined below. Technical support is available by email at support@wildmousesoftware.com.

Download

- Please be certain that the requirements below are satisfied before you download the software.
- [Wild Mouse Software's Simple Warehouse Mapper.](#)
- [View the Online Help.](#)

Software Requirements

- Microsoft .Net version 2.0. Instructions for downloading and installing .Net are available at [Microsoft's download site](#).
- Operating System: Microsoft Vista®, Microsoft Windows XP® with Service Pack 2, or Windows 2000® with Service Pack 6.









Hardware Requirements

- Pentium III 450 Mhz processor or better.
- 3D capable graphics with 16 Mbytes of video memory or better.
- 256 Mbytes system memory.or better
- Printing Requirements
 - A color printer is highly recommended.
 - The printer should have at least 4 Mbytes of memory.

Slika 9.3: Spletna navodila za namestitev programskega orodja

9.3 Uporaba


Programsko orodje uporabljamo za vizualizacijo skladiščnih tokov in za prikaz skladiščnega sistema. Najprej s klikom na *File* in *New* v menijski vrstici odpremo nov dokument. V kolikor imamo dokument kreiran, le tega poiščemo v podatkovni bazi s klikom na *File* in *Open* v menijski vrstici (Slika 9.4 in 9.5).


Za lažje delo je priporočljivo uporabljati orodja, ki jih najdemo v menijski in orodni vrstici na Sliki 9.6. Z uporabo orodne vrstice hitreje odpremo nov dokument s klikom na , shranjen dokument poiščemo s klikom na , ustvarjen izdelek shranimo s klikom na , s klikom na  tiskamo, razveljavimo korake s klikom na  , pogled povečamo ali zmanjšamo s klikom na   itd.

Potrebno je določiti še velikost grafične mreže, na kateri vizualiziramo primer skladiščnega sistema (Slika 9.7 in 9.8).

Ko pripravimo in nastavimo ustrezno velikost grafične mreže, pričnemo z vizualizacijo skladiščnega sistema. Številna orodja ponujajo možnosti za celovito vizualizacijo. *Orodja* so razdeljena na področja: regali, stojala, stene, pisarniška oprema, informacijska tehnologija in transportne poti.

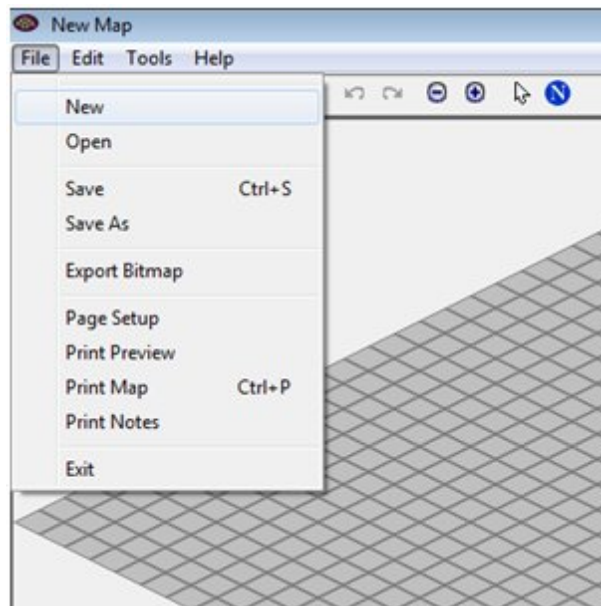
Na grafično mrežo poljubno vstavljamo različne predmete in oblikujemo skladiščni prostor. Oblikovanje skladiščnega sistema je v realnih primerih posnetek obstoječega stanja. Vizualizacija obstoječega stanja omogoča hitri pregled in identifikacijo problemov. Široka paleta različnih predmetov omogoča natančen posnetek stanja in odličen prikaz integracije RFID tehnologije v skladiščne procese in prostore.

Poleg vstavljanja različnih predmetov imamo možnost dodajanja besedila k predmetom (Slika 9.9). Ta storimo s klikom na  v orodni vrstici. Dodajanje besedila k predmetom omogoča lažje razumevanje izdelane 3D vizualizacije prostora, če dodatno pojasni, zakaj je določen predmet na določenem mestu. Dodajanje besedila je zelo priporočljivo, kadar izdelek kasneje predstavljamo partnerjem, kupcem...

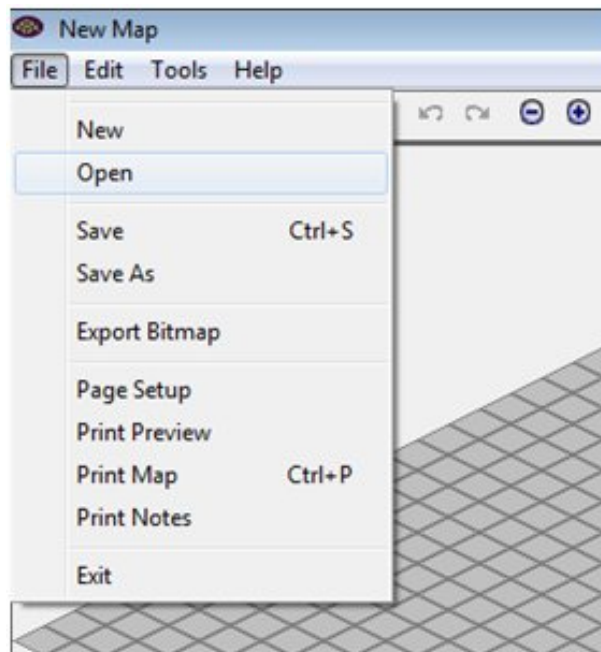
Pri izdelavi 3D vizualizacije prostora delo olajšamo in pospešimo s pomočjo kopiranja predmetov, ki jih uporabimo in vstavimo. Izbrano možnost izberemo, kadar želimo multiplicirati določene predmete (cesta, stena, regali na Sliki 9.10) ali v primeru, ko želimo vizualizirati dva identična prostora oz. objekta. Kopiranje izvedemo tako, da s klikom na ikono  aktiviramo miško, izberemo področje kopiranja, ga označimo in kopiramo. Kopijo vstavimo na poljubno izbrano lokacijo na mreži (Slika 9.11).

Pogled na mrežo je možno povečati in zmanjšati s klikom na okno + oz. - (Slika 9.12).

Povečava pogleda omogoča podrobnejšo vizualizacijo vsakega kotička v skladiščnem prostoru, pomanjšan pogled pa omogoča vizualizacijo in pregled



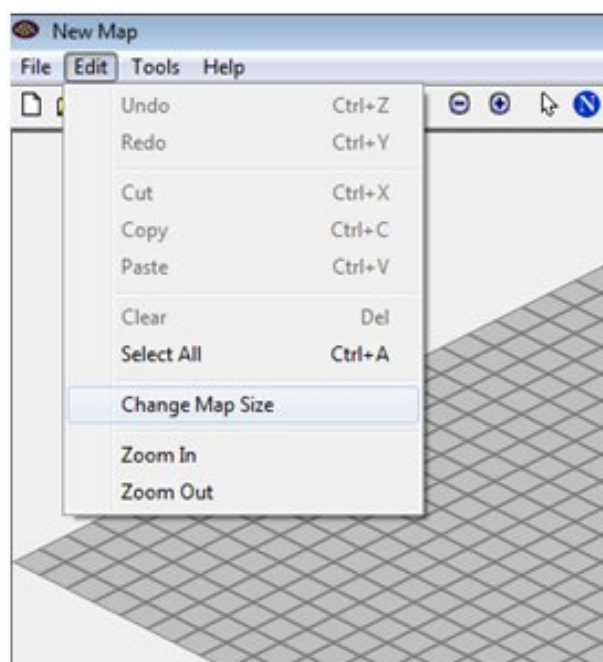
Slika 9.4: Nov dokument



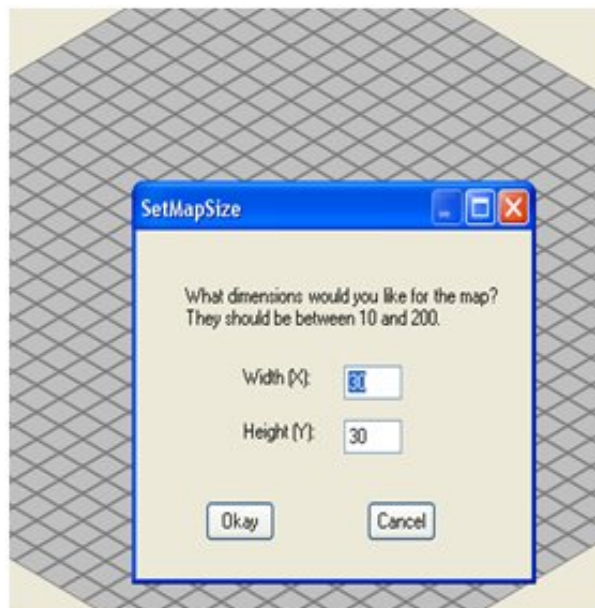
Slika 9.5: Izbor dokumenta



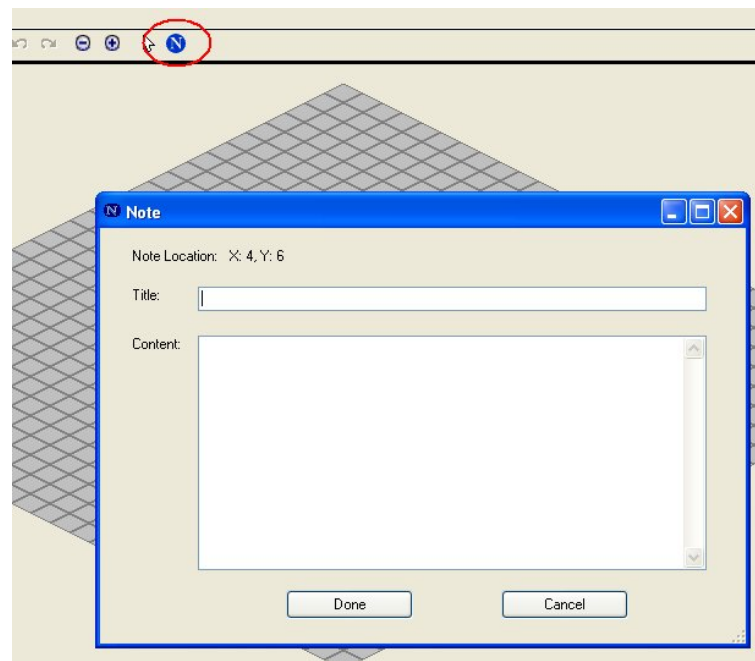
Slika 9.6: Menijska in orodna vrstica



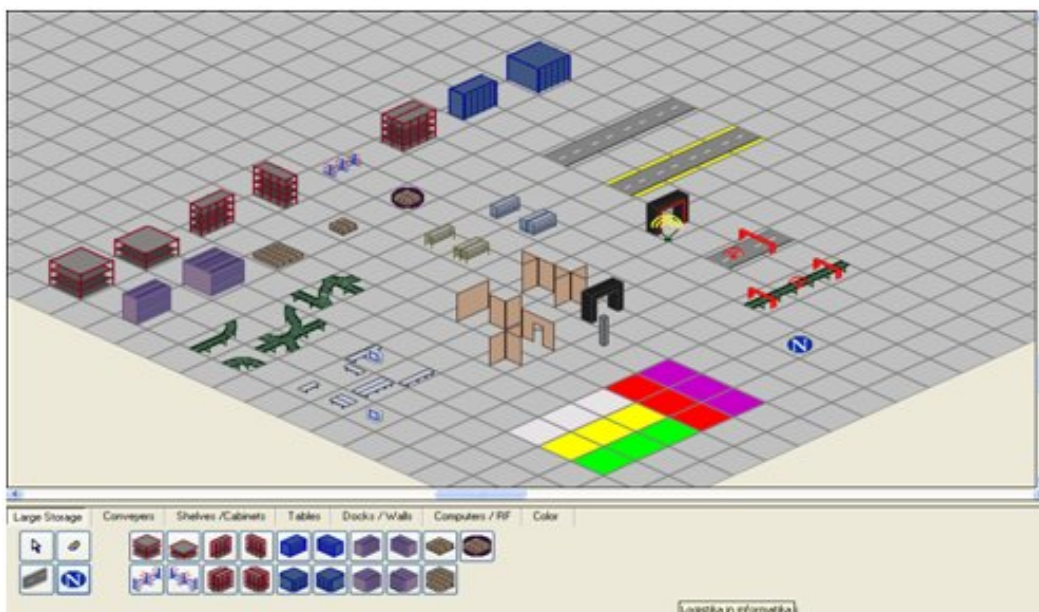
Slika 9.7: Sprememba nastavitve velikosti grafične mreže



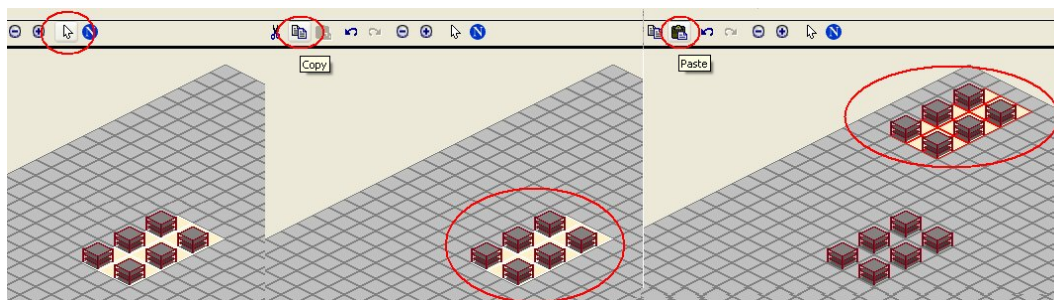
Slika 9.8: Določitev velikosti grafične mreže



Slika 9.9: Vstavljanje besedila



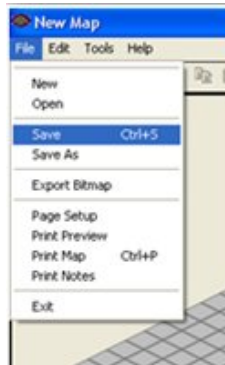
Slika 9.10: Različni predmeti za vizualizacijo prostora



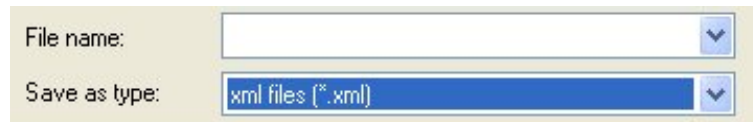
Slika 9.11: Kopiranje predmetov



Slika 9.12: Povečanje/zmanjšanje pogleda




Slika 9.13: Kako shraniti datoteko



Slika 9.14: Oblika shranjene datoteke

celotnega skladiščnega prostora.

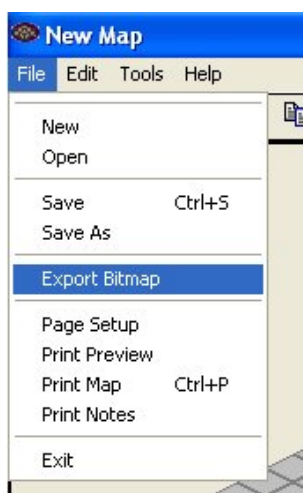
Ko vizualiziramo obstoječe stanje, dokument preprosto shranimo v podatkovno bazo s klikom na  v orodni vrstici ali *File, Save* v menijski vrstici (Slika 9.13). Datoteke se shranjujejo kot .xml, kar omogoča izmenjavo in hitrejšo procese (Slika 9.14).

Programsko orodje Simple Warehouse Mapper omogoča tudi izvoz dokumenta v .bmp obliko, kar olajša vstavljanje in izvažanje kopij slik (map) za predstavitev druge dokumente (Slika 9.15).

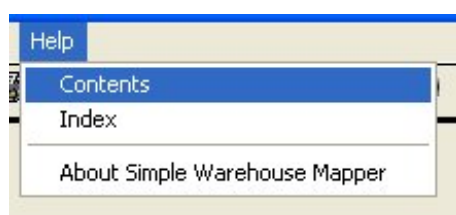
Dodatna pojasnila in pomoč najdemo v menijski vrstici. S klikom na *Help*, prikazano na Sliki 9.16 in nadalje *Contents*, dostopamo do vseh podrobnejših navodil za izdelavo 3D vizualizacije prostora.

Praktični prikaz uporabe programa Simple Warehouse Mapper

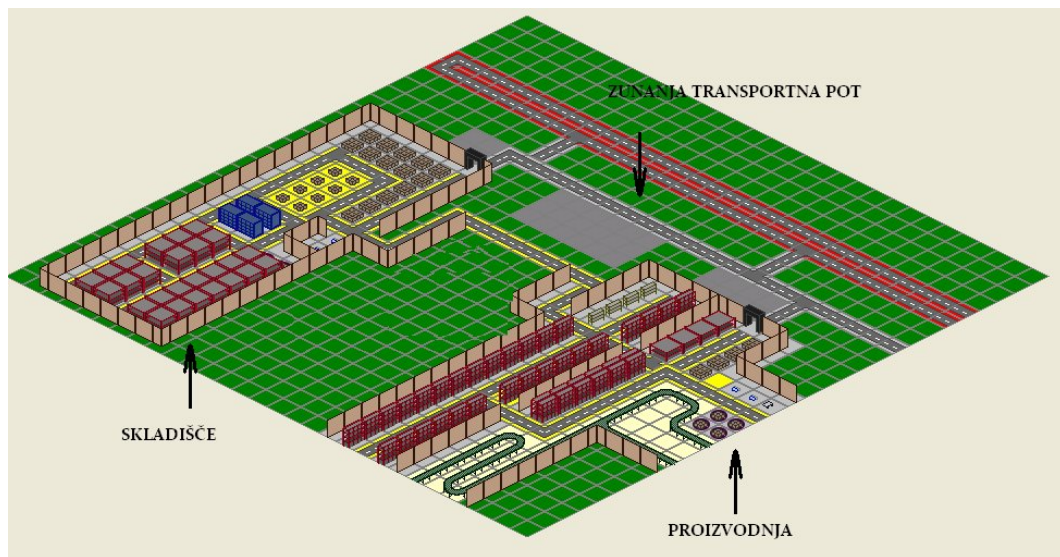
Za izdelavo vizualizacije posnetka stanja našega podjetja uporabimo različna orodja in predmete. Za prikaz okvira skladiščnega sistema izberemo stene in zidove (na sliki kremne barve), za prikaz skladiščnih transportnih poti izberemo cesto, za prikaz skladiščnih sistemov (skladiščenje) izberemo različne vrste skladiščnih enot (regali), za prikaz proizvodnje linije izberemo tekoči trak (zelene barve), za prikaz oblikovanih komisij smo izbrali različne vrste



Slika 9.15: Izvoz datoteke



Slika 9.16: Pomoč



Slika 9.17: Praktični prikaz vizualizacije procesa skladiščenja

palet, za prikaz uporabe sodobne informacijske tehnologije izberemo RFID tehnologijo, ki smo jo namestili na vhodne in izhodne točke skladiščnega sistema in jih povezali z centralnim informacijskim sistemom. Prikaz skladišča na Sliki 9.17.

Z uporabo programskega orodja Simple Warehouse Mapper vizualiziramo skladišče prostor - pridobimo vizualno kontrolo nad skladiščnimi procesi. Vizualizacija omogoča pregled nad materialnimi tokovi v skladiščnem prostoru, torej nad točkami, kjer se izvajajo določene skladiščne aktivnosti (prevzem, skladiščenje, transport, odprema itd.).

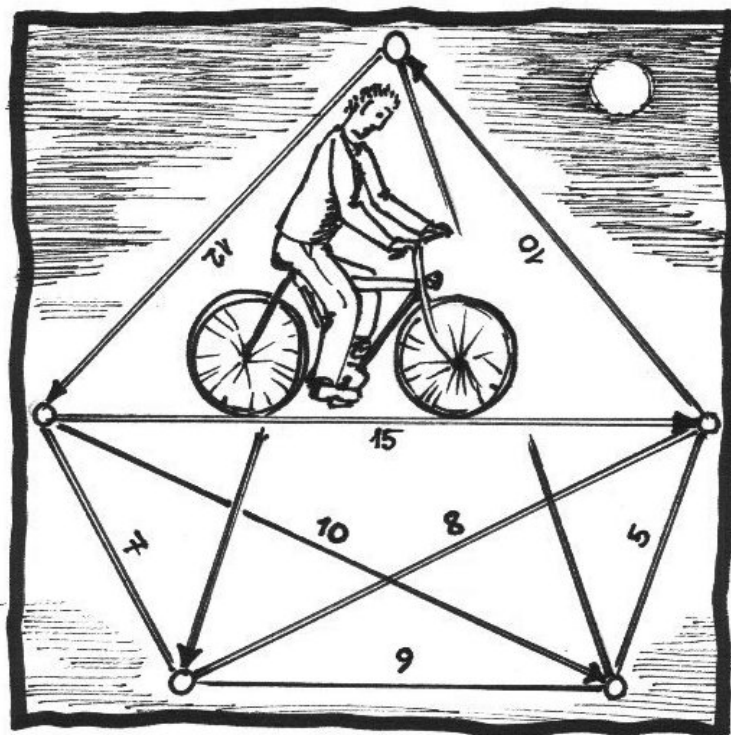
Povzetek

Programsko orodje Simple Warehouse Mapper je odlično 3D orodje za vizualizacijo materialnih tokov v skladiščnem in proizvodnem procesu. Vizualizacija prostora omogoča hitro identificiranje napak in učinkovito planiranje skladiščnega sistema. Prav tako omogoča lažjo predstavitev dejanskega stanja v realnem svetu. Z uporabo programa Simple Warehouse Mapper prikažemo:

- uporabo programskega orodja;
- vizualizacijo realnega primera;
- posnetek obstoječega stanja skladiščnega in proizvodnega sistema;
- umestitev informacijske tehnologije;
- ureditev in razporeditev skladiščnega prostora.

Poglavje 10

PETERSEN - odločanje na podlagi teorije grafov



Teorija grafov
Minimalno vpeto drevo
Primer: iskanje najkrajših poti z izbranimi algoritmi

10.1 Teoretično ozadje

10.1.1 Teorija grafov

Kadar govorimo o teoriji grafov, govorimo o veji matematike in računalništva, ki raziskuje lastnosti grafov. Začetke razvoja beležimo vse od druge polovice 19. stoletja. Razvila se je iz potrebe po reševanju konkretnih primerov v znanosti in tehnologiji. Graf definiramo kot množico objektov in reči, imenovane točke (vozlišča, vozil...), ki so povezane s povezavami (robovi, veje...). S teorijo grafov rešujemo številne primere v praksi. Grafe lahko razširimo z vpeljavo uteži, ko so pozitivna števila, prirejena vsaki povezavi. Če npr. graf predstavlja mrežo cest, lahko uteži predstavljajo dolžino vsake ceste. Če grafu dodamo utežene povezave, govorimo o mreži. Grafe uporabljamo predvsem pri metodi mrežnega planiranja.

Rečemo lahko, da graf G sestavljata neprazna množica elementov, ki jih imenujemo točke ali vozlišča grafa in seznam (neurejenih) parov teh elementov, ki jih imenujemo povezave grafa. Grafe si enostavno predstavljamo z njihovimi diagrami. Točke (vozlišča) ponazorimo s krogci, sosednji točki pa povežemo s črto.

Minimalno vpeto drevo

Minimalno vpeto drevo je pomembno za reševanje problemov načrtovanja optimalnih (najpogosteje najcenejših) prometnih in drugih omrežij. Prometni problem lahko rešujemo s pomočjo neusmerjenega grafa. Vozlišča v grafu predstavljajo mesta, ki jih želimo povezati, povezave pa so označene z razdaljami, časom oz. cenami povezave med dvema krajema. Ker za povezavo vseh mest zadošča, da v grafu obstaja ena pot med vsakim krajem, problem definiramo kot iskanje minimalnega (najcenejšega) podgrafa, ki izpolnjuje ta pogoj. Takšen podgraf imenujemo minimalno vpeto drevo. Iskanje minimalnega vpetega drevesa v povezanem neusmerjenem grafu izvedemo s pomočjo dveh znanih algoritmov - Kruskalov in Primov.

Kruskalov algoritem

Kruskalov algoritem je model, ki se uporablja za iskanje minimalnega vpetega drevesa v omrežju. Drevo začnemo graditi iz najkrajše povezave, ki je že sama zase vpeto drevo in nato naprej iščemo najkrajše povezave, dokler nimamo celega vpetega drevesa (pri tem smo pozorni, da ne pride do cikla). V izbranem načinu gradimo drevo tako, da na vsakem koraku v rešitev dodamo povezavo z minimalno vrednostjo, ki ima največ eno izmed točk, ki so v enem delu rešitve. V

primeru, ko je prva točka povezave v enem delu rešitve in druga točka povezave v drugem delu rešitve, ta dva dela rešitve združimo.

Primov algoritem

Tudi Primov algoritem se uporablja za iskanje minimalnega vpetega drevesa v omrežju. Drevo začnemo graditi iz poljubnega vozlišča, ki je že samo zase vpeto drevo in nato dodajamo najkrajše povezave, dokler ne dobimo celega vpetega drevesa. V grafu poiščemo povezave, s katerimi povežemo vse točke, ki imajo najmanjšo vsoto povezav. Postopek računanja poteka tako, da v celotnem grafu izberemo najkrajšo povezavo med vozlišči v doslej zgrajenemu drevesu in vozlišči, ki še niso del drevesa in z njo ne naredimo cikla, kar je osnoven korak. Če sta dve povezavi z enakimi vrednostmi, izberemo tisto, ki ne naredi cikla. Če nobena povezava ne naredi cikla, izberemo eno izmed njiju. To povezavo dodamo in postopek ponavljamo, dokler obstajajo prosta vozlišča.

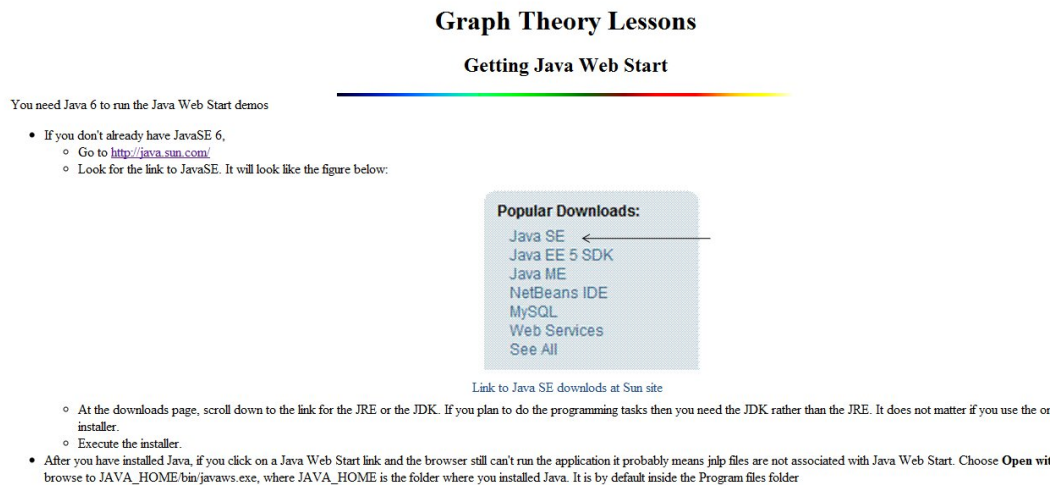
10.2 O programskem orodju

Petersen programsko orodje je nastalo pod okriljem avtorja Chrisa Mawata. Izdelujemo, urejamo in manipuliramo enostavne grafe ter preučujemo njihove lastnosti. Prikazujemo podatke o grafu (število točk in njihove stopnje), matriko sosednosti in število sestavnih delov; preverimo, če je graf dvostranski, če sta dva grafa izomorfna ali če je graf podgraf drugega. Petersen dokazuje Euler in Hamilton vezja, algoritme za iskanje minimalnih vpetih dreves ipd.

Pri pisanju programskega orodja je avtor poskušal rešiti čim več izobraževalnih in učnih problemov. Eden izmed razlogov nastanka je prikaz velikih in zapletenih grafov. Za pripravo takšnih grafov je potreben bistveno daljši čas, saj je že samo pridobivanje informacij iz velikih grafov mučno in dolgotrajno. Algoritmi v programu niso novi. Edina razlika med Petersen programskim orodjem ter poslovnimi in raziskovalnimi aplikacijami je v tem, da je njegov glavni cilj dobiti rezultate v najkrajšem možnem času. V izobraževalnih ustanovah pa je primarni cilj, da študent spozna in vidi grafičen prikaz nekega problema.

Prenos in namestitvev

Programsko orodje Petersen je dostopno na spletnem naslovu Mathcove [33]. Deluje preko spletne aplikacije Java. V kolikor Jave nimamo že predhodno nameščene, na spletni strani (*Getting Java Web Start*) izberemo možnost *go to Java* [22] (glej Sliko 10.1).



Slika 10.1: Prenos Java

Brezplačno Petersen programsko orodje prenesemo tako, da v razdelku *Getting Petersen*, iz spletne strani Mathcove [33] (glej Sliko 10.2), v 2. poglavju izberemo možnost *download petersen-3.2.1.jar* (glej Sliko 10.2).

Programsko okno

Pred pričetkom uporabe programskega orodja se informiramo o funkcijah in orodjih, ki so na voljo. Ob zagonu programskega orodja se v začetnem oknu pojavi menijska vrstica, katere funkcije razložimo v nadaljevanju (glej Sliko 10.3).

Menijska vrstica *Graph* omogoča številne možnosti urejanja in oblikovanja grafov. V primeru, ko izberemo možnost *Named Graph* in v nadaljevanju *Null Graph* se prikaže možnost določitve točk (*How many vertices?*). V okence vpišemo število točk, ki bodo vključene v graf (glej Sliko 10.4).

V razdelku menijske vrstice *Graph - Named Graph* v *Complete* izbiramo med tremi razdelki. Izberemo možnost izrisa grafa z določitvijo števila povezav, ki bodo povezane z že narisanimi točkami (glej Sliko 10.5).

Ob izbiri razdelka *Complete Bipartite Graph* zapišemo število točk na levi in desni strani (glej Sliko 10.6).

V razdelku *Complete Tripartite Graph* izbiramo in zapišemo 3 vrste števil: število točk v prvi skupini, število točk v drugi skupini in število točk v tretji skupini (glej Sliko 10.7).

Razdelek *Circuit Graph* izriše graf, ki je povezan v cikel. Izberemo zgolj možnost števila točk, za katere želimo, da bodo vsebovane v določenem

Graph Theory Lessons

Getting the Petersen Program

1. Check that you have Java 6 or later properly installed

You need Java 6 to run the Petersen program. To check that you do have Java 6 installed, get a command console and type

```
java -version
```

The output should indicate Java version 1.6 or later. You can also do a more thorough test at [the Java test page](#).

2. Download the program

Here is the link to download the program: [download petersen-3.2.2.jar](#)

You can get the md5 checksum here: [md5 for petersen-3.2.2.jar](#)

3. Run the program

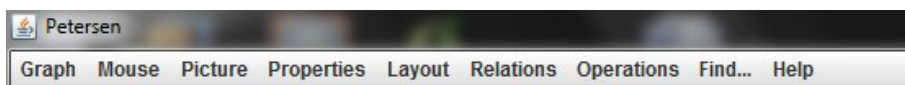
There is no "installer" to run since the program only needs Java to be present in order to run.

- Windows users can just double click on the jar file from Windows Explorer (assuming Java 6 or later is properly installed).
- Alternatively, users on any platform can get a command console and type
`java -jar petersen-3.2.2.jar`

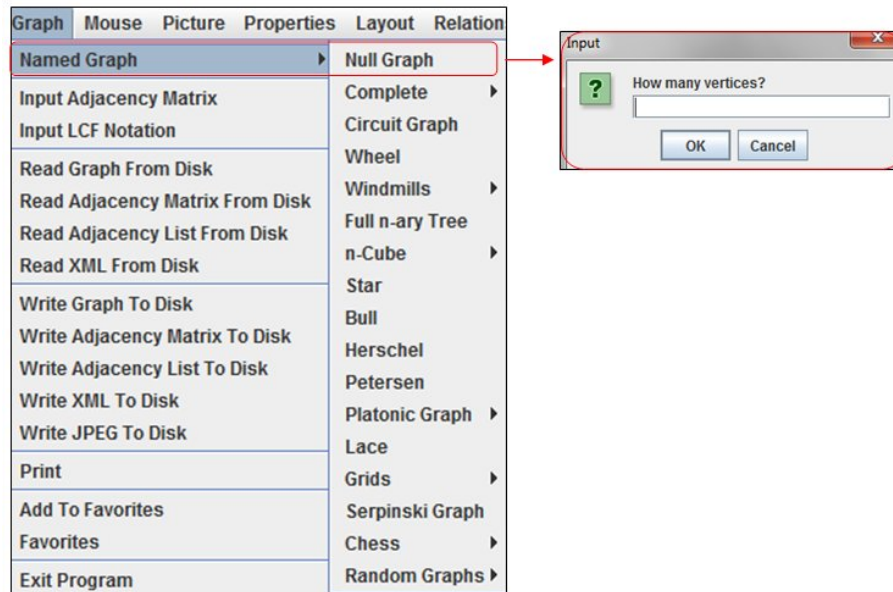
4. (Optional) Get a free 30 vertex license

The bare bones program will only handle 15 vertices. This might be enough for the casual reader. You can get a short term free license for 30 vertices at [Get Free License](#). You then paste the text of the license dialog under the menu item **Help** | **License** of the Petersen program. Institutions using the program for a course will probably want students to investigate larger graphs and should consider getting a site license allows the program to handle 100 vertices.

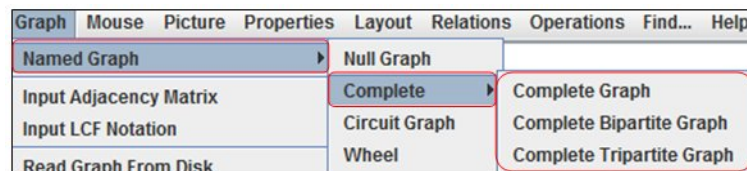
Slika 10.2: Prenos Petersen-ovega programskega orodja



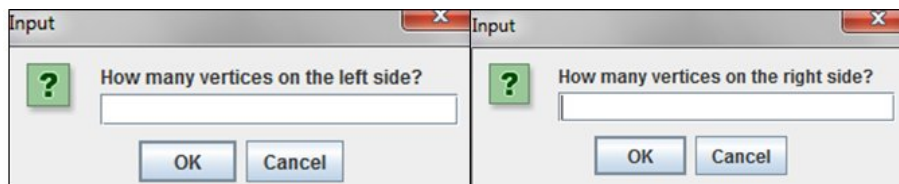
Slika 10.3: Menijska vrstica



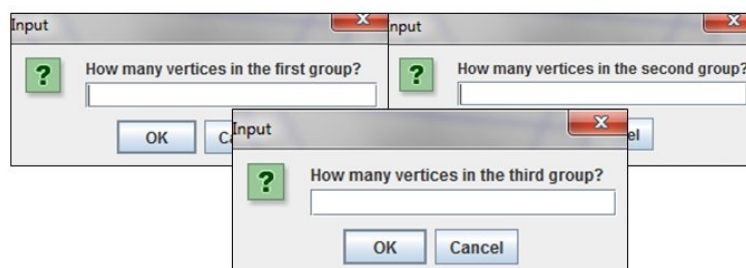
Slika 10.4: Določitev števila točk – Null Graph



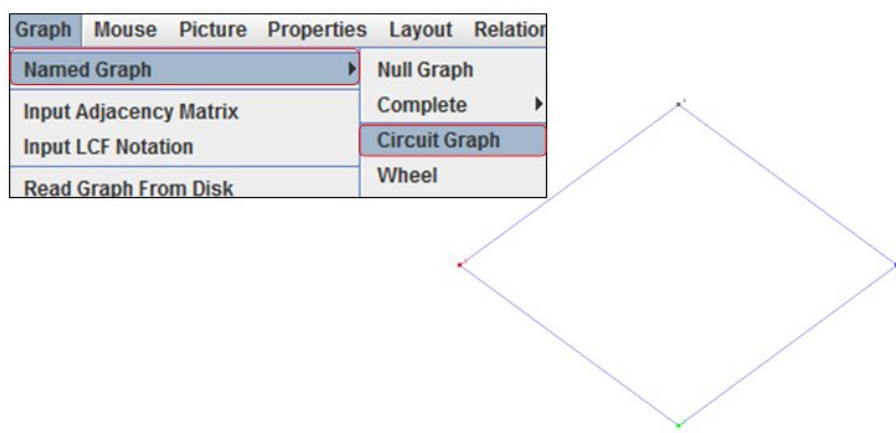
Slika 10.5: Določitev števila točk – Complete



Slika 10.6: Complete – Complete Bipartite Graph



Slika 10.7: Complete-Complete Tripartite Graph



Slika 10.8: Circuit Graph

cikličnem grafu (glej Sliko 10.8).

Ob izbiri razdelka *Wheel* se na grafu (npr. 4 povezane točke) izriše določeno število točk (npr. 5 točk). Točke se izrišejo na način, da je ena točka povezana z vsemi ostalimi na grafu (glej Sliko 10.9).

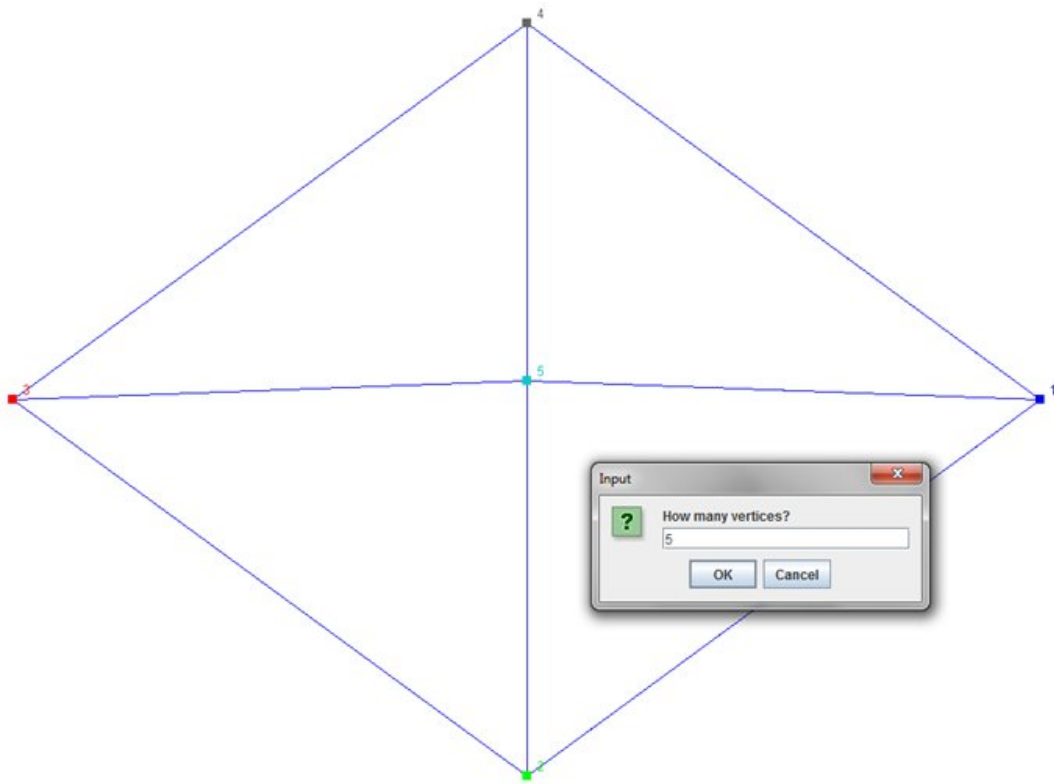
Izbiramo lahko med posameznimi oblikami grafov (*Windmills*). Ti se lahko izrišejo v obliki metuljčka (*Butterfly*) ipd. (glej Sliko 10.10).

Full n-ary Tree prikazuje možnost izrisa razcepljenega grafa. Izberemo možnosti razcepov s po 2 točkama (glej Sliko 10.11).

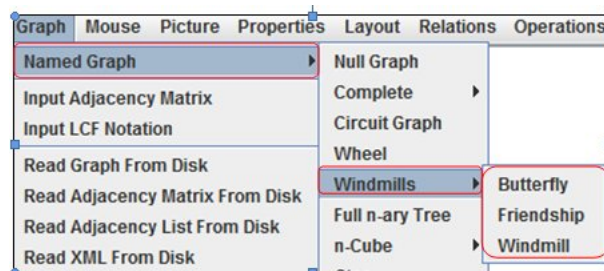
Izbiramo lahko tudi med izrisom različnih vrst grafov, kot so: *n-Cube*, *Star* (graf v obliki zvezde), *Bull*, *Herschel in Petersen graf* (glej Sliko 10.12).

Podana je še možnost izrisa (*Platonic Graph*) v obliki tetraedra, oktaedra ipd. (glej Sliko 10.13).

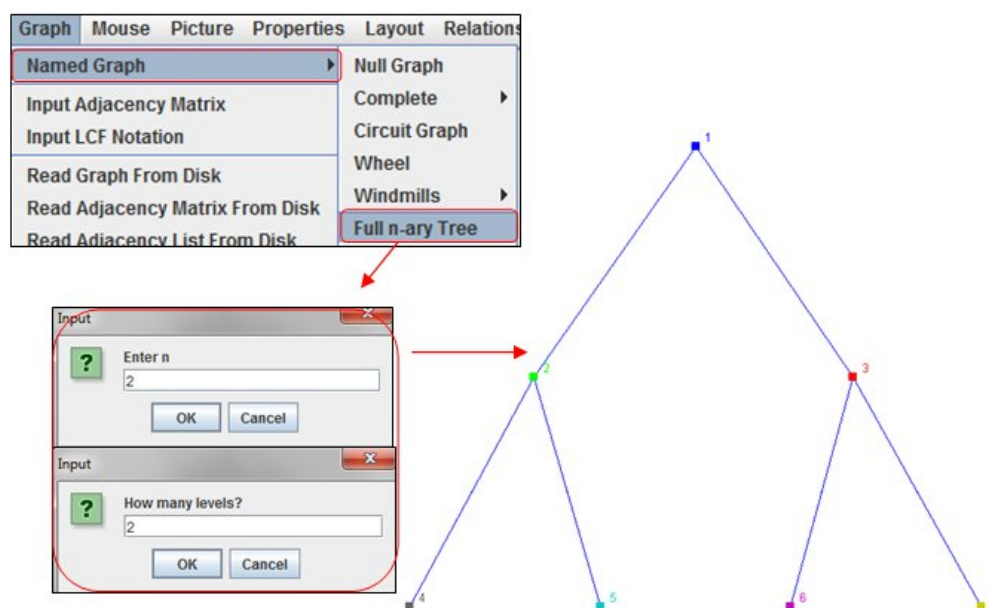
V menijski vrstici *Graph* so na voljo tudi druge možnosti: vnos podatkov za vhodno matriko sosednosti (*Input Adjancency Matrix*) in LCF zapis (*Input LCF*



Slika 10.9: Wheel



Slika 10.10: Windmills



Slika 10.11: Full n-ary Tree

Notation). V programu lahko odpremo že shranjen dokument na disku (graf - *Read Graph From Disk*, matrika sosednosti – *Read Adjacency Matrix From Disk* ipd.). V danem razdelku izberemo tudi možnost tiskanja grafa (*Print*) in izhoda iz programskega orodja (*Exit Program*) (glej Sliko 10.14).

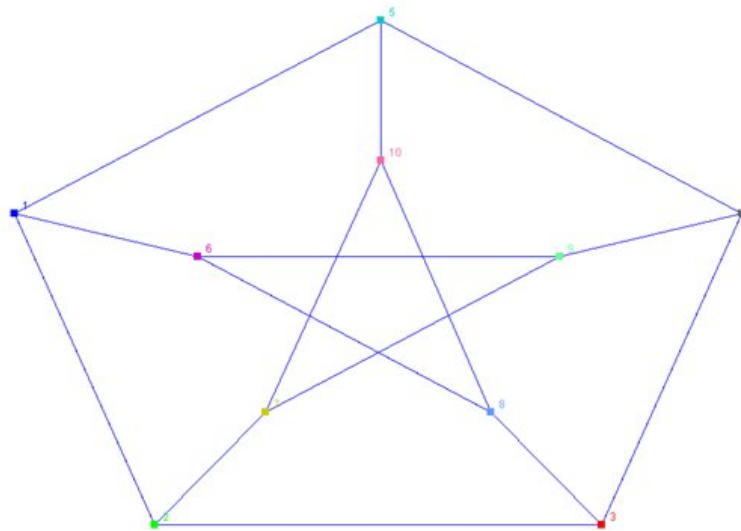
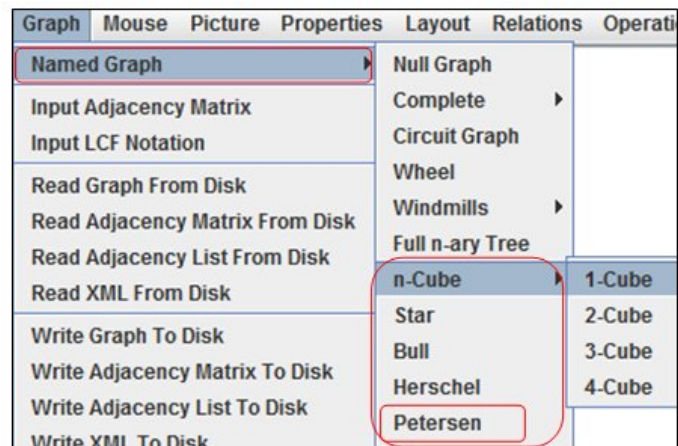
Na koncu menijske vrstice se nahaja razdelek *Pomoč* oz. *Help* (Slika 10.15). V razdelku *About* je naslov spletne strani, kjer so na voljo informacije za uporabo programskega orodja.

Na spletni strani najdemo še informacije iz posameznega področja (npr. klik na dano spletno stran: Mathcove [33], prikaže meni s številnimi podatki).

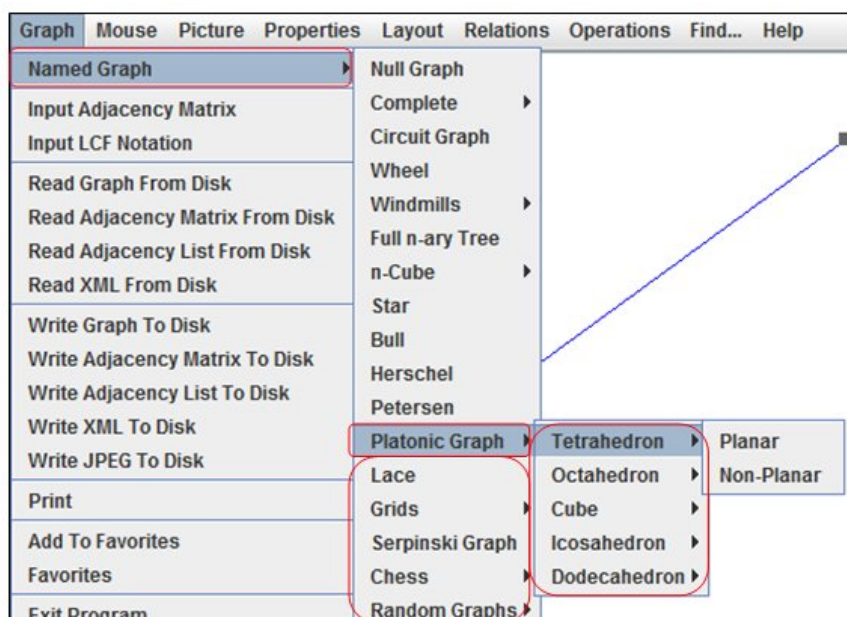
Pri poglobljenem proučevanju določenega področja je smiselno prebrati besedilo, zapisano pod posamezno lekcijo. Na koncu vsake lekcije je povezava, ki direktno preusmeri na področje, ki nas zanima. Na Sliki 10.16 je prikazana teorija o minimalno vpetem drevesu.

S klikom na povezavo (vijoličaste barve) se prikaže okno programskega orodja (glej Sliko 10.17).

Izriše se prazno modro okno, kjer imamo na voljo povsem lastno izbiro dodajanja ali brisanja povezav in linij (glej Sliko 10.18).



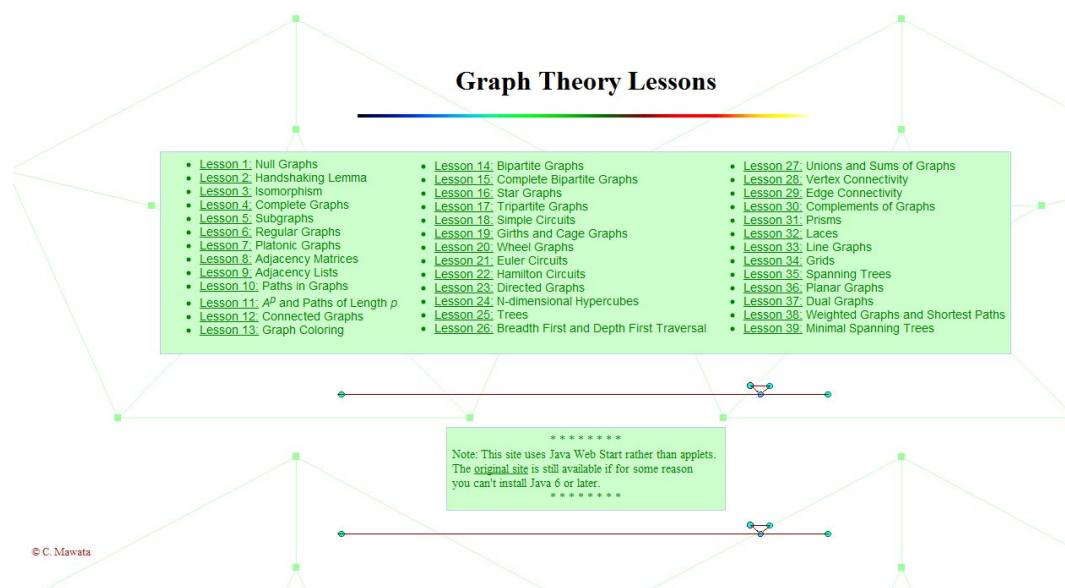
Slika 10.12: Različne vrste grafov – 1. del



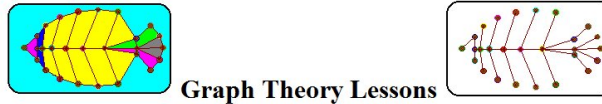
Slika 10.13: Različne vrste grafov – 2. del

Graph	Mouse	Picture	Properties
Named Graph			
Input Adjacency Matrix			
Input LCF Notation			
Read Graph From Disk			
Read Adjacency Matrix From Disk			
Read Adjacency List From Disk			
Read XML From Disk			
Write Graph To Disk			
Write Adjacency Matrix To Disk			
Write Adjacency List To Disk			
Write XML To Disk			
Write JPEG To Disk			
Print			
Add To Favorites			
Favorites			
Exit Program			

Slika 10.14: Menijska vrstica in razdelek Graph



Slika 10.15: Dodatna pomoč pri uporabi programa



Lesson 35: Spanning Trees

A *spanning tree* for a *connected graph* G is a subgraph of G that is a *tree*, (i.e., is connected and has no circuits) and contains all the vertices of G . If G is a tree then it has only one spanning tree (G itself) G is not a tree it will have more than one spanning tree.

Petersen Activity:

We will first illustrate a depth first approach to getting a spanning tree. Get the grid $Grid_{4,5}$. Now click **Properties | Spanning Trees | Depth First Spanning Tree**. You will see the depth first spanning should get a picture like figure 35.1

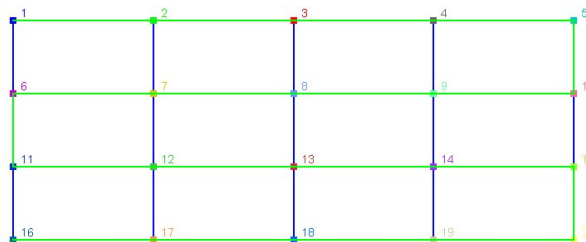


Fig 35.1. A depth first spanning tree for $Grid_{4,5}$ (displayed in green).

Click **Graph Exit** to get back to the main Petersen frame and now find a breadth first spanning tree by selecting **Properties | Spanning Trees | Breadth First Spanning Tree**. You should get a picture like figure 35.2

Slika 10.16: Dodatna literatura na temo minimalnega vpetega drevesa

Java Web Start Activity:

The Java Web Start application below will help you to understand the difference between a depth first and breadth first spanning tree. Draw a connected graph in the left pane. The depth first spanning be displayed in the middle pane and the breadth first tree will be displayed in the right pane.

[Java Web Start Application 35.1: Depth First and Breadth First Spanning Trees](#)

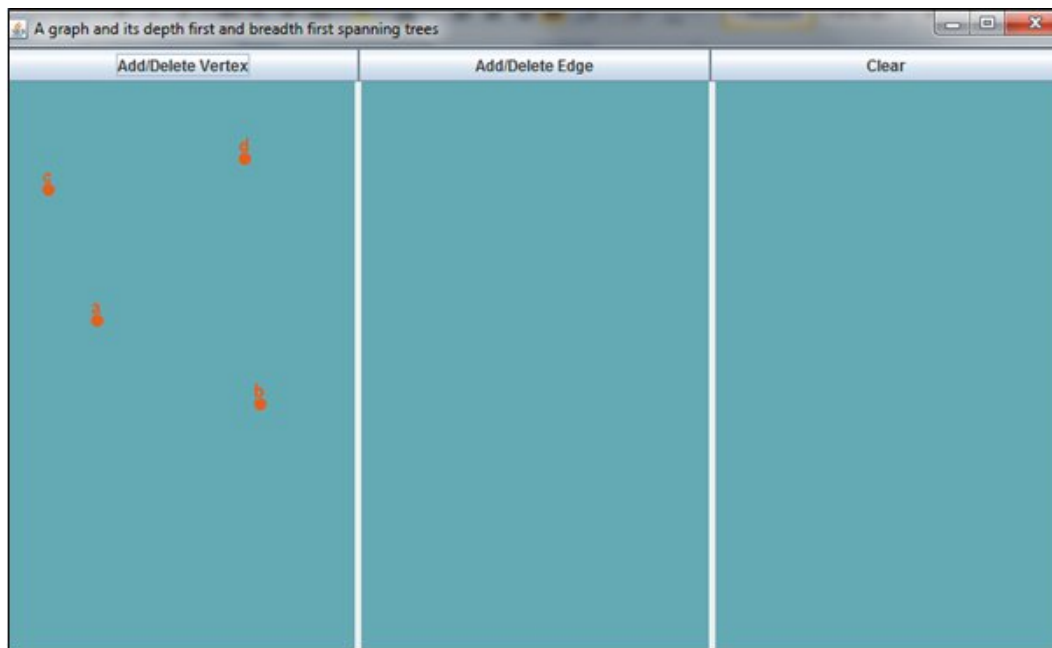
Questions:

1. For the Herschel graph, draw a depth first and a breadth first spanning tree and then two other spanning trees.
2. Find a graph on five vertices that is not a tree but such that all of its spanning trees are isomorphic.
3. Which graph is isomorphic to the breadth first spanning tree of the *complete graph* K_n ?

[Answers](#)

[Comments](#)

Slika 10.17: Klik na direktno povezavo



Slika 10.18: Posamezna programska okna

Problem

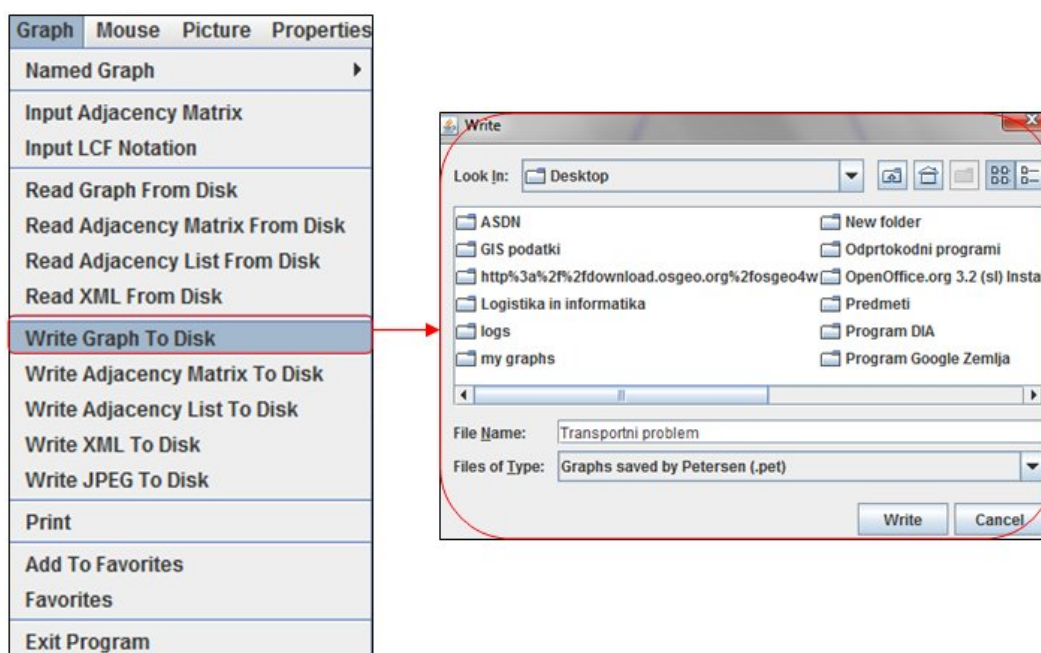
Z danim programskim orodjem v nadaljevanju prikažemo problem, opažen v resnični situaciji. V namišljenem podjetju OpenStorage moramo poskrbeti za transport komponent, namenjenih za izdelavo avtomobilov znamke X. Komponente (natančneje platišča) se v podjetje dostavljajo s cestnim transportom (prevoz s tovornjaki), pri čemer so mnogokrat izpostavljene oviram na poti, zato je potrebno izbrati pot, kjer ovire obidemo z minimalni stroški.

V ta namen s Petersen programskim orodjem prikažemo problem dostave platišč od dobavitelja do podjetja v Bohinju. Izračunamo najkrajšo pot po kateri dobavitelj pripelje platišča na dostavno mesto, preizkusimo pa še nekaj izmed drugih možnosti, ki jih programsko orodje omogoča.

10.3 Uporaba

V menijski vrstici *Graph* izberemo razdelek *Write Graph To Disk*, kjer narisan graf shranimo na poljubno mesto (glej Sliko 10.19).

Ob ponovnem odpiranju shranjenega dokumenta odpremo Petersen



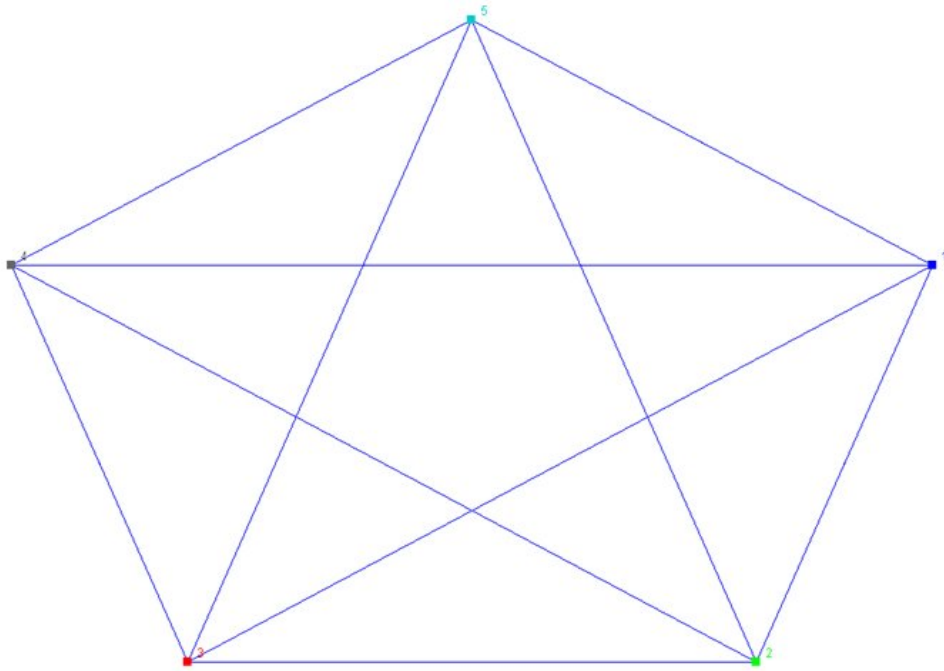
Slika 10.19: Shranjevanje datoteke

programsko orodje v Javi in znotraj njega izberemo možnost (*Read Graph From Disk*). Za prikaz problema izbranim točkam določimo razdalje (uteži na povezavah). Določimo 5 točk, ki jih med sabo povežemo. Risanje grafa prične postopoma, tako da najprej zapišemo in povežemo 5 točk. Te narišemo tako, da v razdelku *Named Graph – Null Graph* izberemo možnost 5 točk. Nato jih med sabo, z ukazom v menijski vrstici *Complete* razdelek *Complete Graph* povežemo (ponovno izberemo možnost 5) (glej Sliko 10.20).

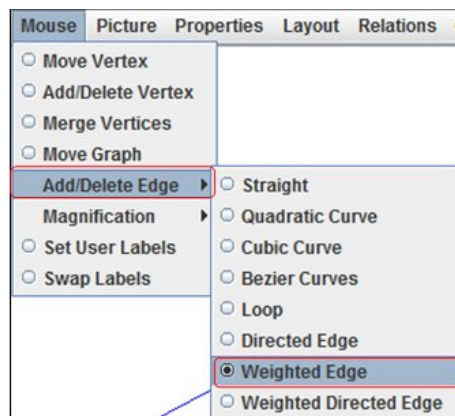
Razdelek *Mouse* omogoča, da z miško spreminjamo točke in povezave, jih dodajamo ali brišemo (*Move Vertex, Add/Delete Vertex, Merge Vertices*). Podana je možnost premikanja grafa (*Mode Graph*), prav tako pa lahko grafom dodajamo določene uteži. Kadar želimo uveljaviti spremembe na grafu, enostavno kliknemo na zeleno možnost, npr. dodajanje uteži (*Weighted Edge*) (glej Sliko 10.21).

Najprej dodamo utež med točkama 4 in 5. Z miško se postavimo na točko 4 in se premikamo do točke 5. Program nas vpraša po vrednosti uteži, ki jo vpišemo v prazno okence. V danem primeru točki 4 in 5 pomenita vozlišče med dvema krajema z razdaljo 4 kilometrov. Enak postopek izvedemo še za določitev ostalih povezav (glej Sliko 10.22).

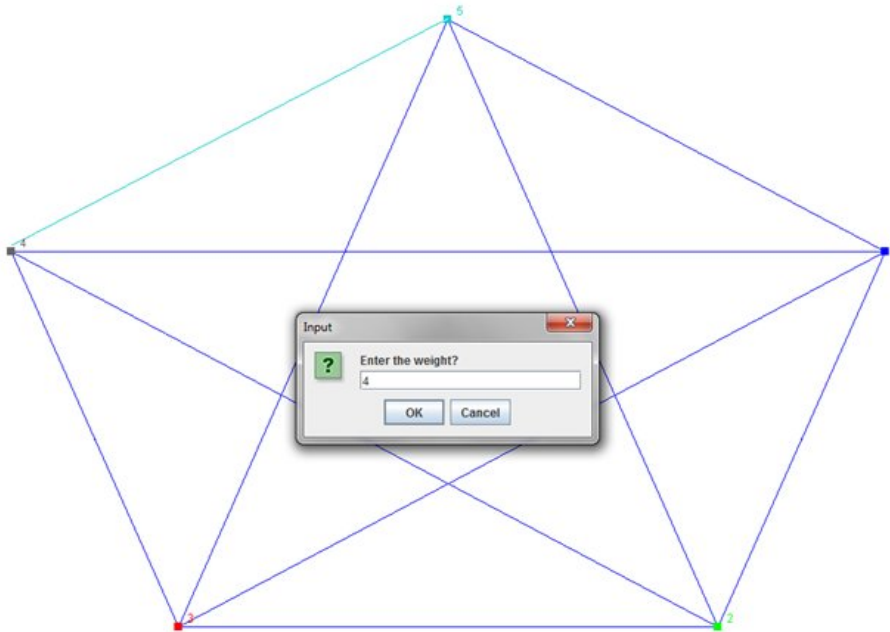
V primeru omejitev izberemo drugo vrsto uteži - *Weighted Directed Edge*.



Slika 10.20: Izris grafa s petimi točkami in povezavami



Slika 10.21: Dodajanje uteži



Slika 10.22: Zapis uteži

Postopek je popolnoma enak, kakor je predhodno zapisano, le da so povezave zapisane od ene do druge točke. Če želimo, da bo povezava obojestranska se z miško postavimo iz ene na drugo točko (npr. $3 \rightarrow 4$) in nato še v obratni smeri (npr. $4 \rightarrow 3$). V izbranem problemu uporabimo obojestranske povezave.

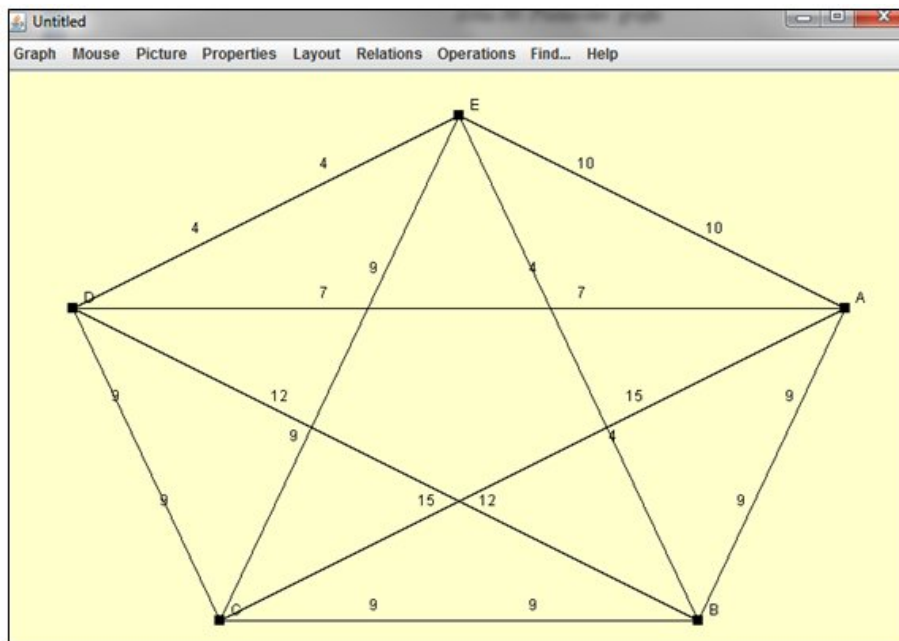
Slika 10.23 prikazuje vrednosti dvosmernih povezav. Da ne bi prišlo do zapletov, vozlišča označimo s črkami, saj so črke bolj razvidne.

Ob zapisu povezav in razdalj med posameznimi točkami izbiramo med možnostmi, ki jih ponuja programsko orodje. Razdelek *Change title* omogoča spreminjanje naslova dokumenta. Razdelek *Background Color* omogoča spreminjanje ozadja grafa - na voljo imamo najrazličnejše barvne sheme (*Color Scheme*) in linije (*Edges*) (glej Sliko 10.24).

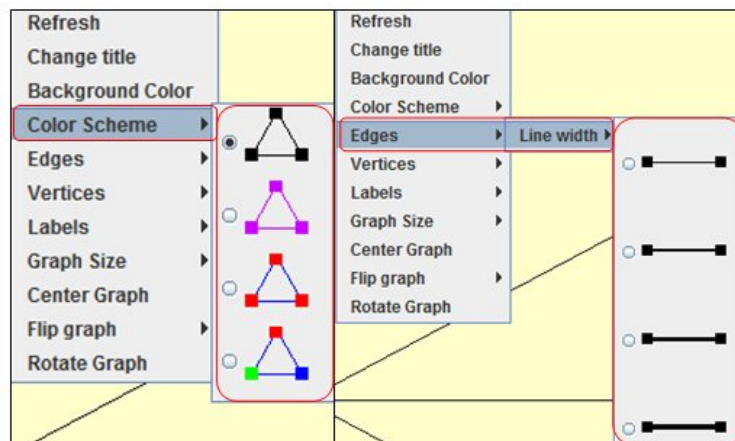
V razdelku *Labels* določimo oznake vozlišč: brez oznak (*No labels*), s številkami (*Numbers*), s črkami (*Letters*) ali z že uporabljenimi oznakami (*User's Labels*). Menijska vrstica *Graph Size* ponuja možnost prikaza grafa (glej Sliko 10.25).

Razdelek *Flip graph* ponuja možnost horizontalnega, vertikalnega in diagonalnega premika. V razdelku *Picture* se nahaja razdelek *Rotate Graph*, ki omogoča poljubno rotacijo grafa (glej Sliko 10.26).

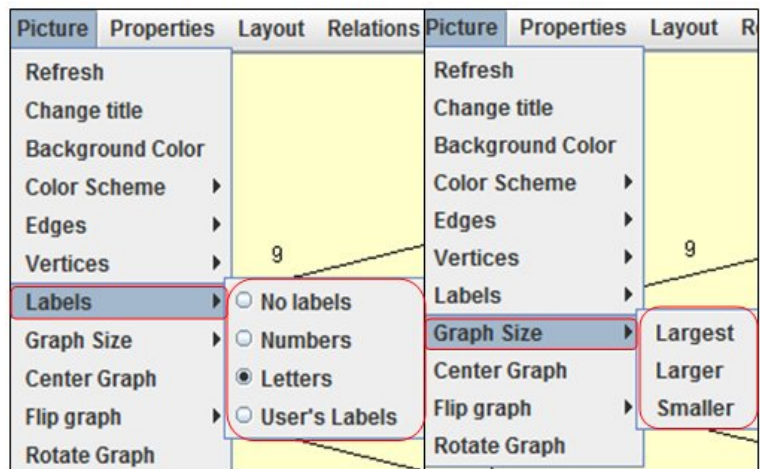
V menijski vrstici se nahaja še meni *Properties*. Razdelek *Statistics* prikazuje



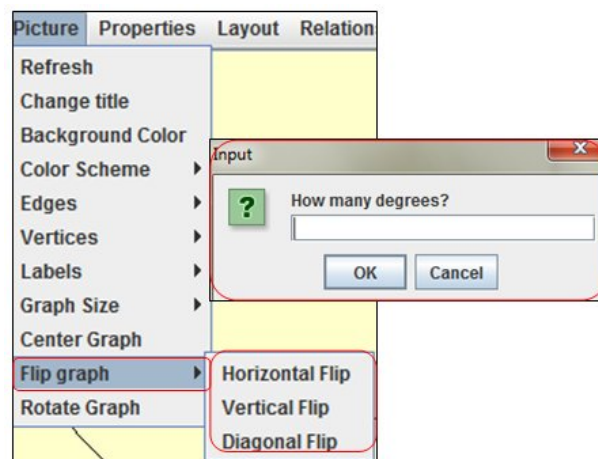
Slika 10.23: Vrednosti povezav



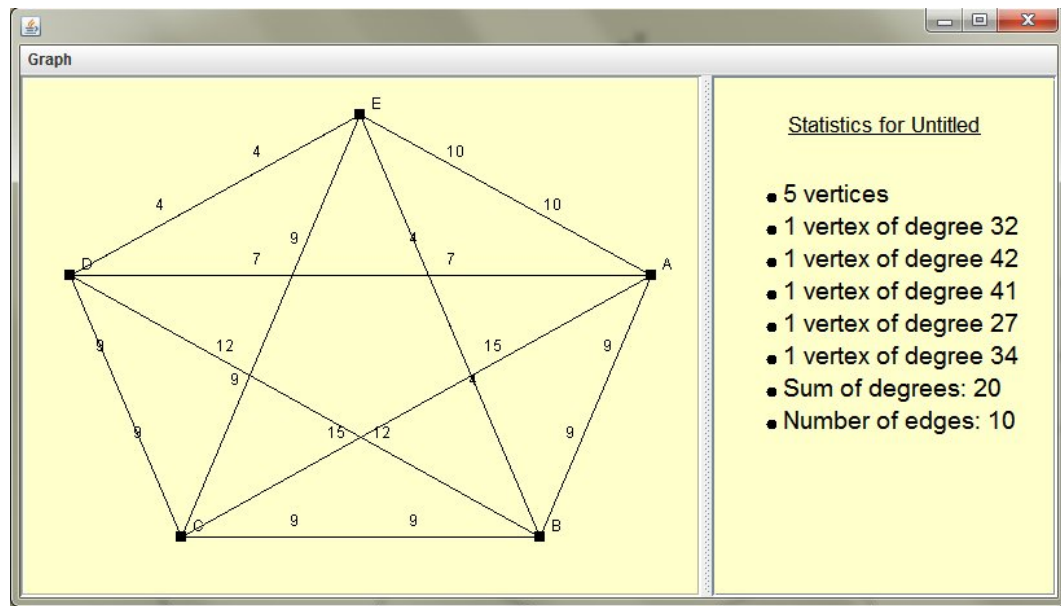
Slika 10.24: Možnosti grafa – 1. del



Slika 10.25: Možnosti grafa – 2. del



Slika 10.26: Možnosti grafa – 3. del



Slika 10.27: Properties- Statistics

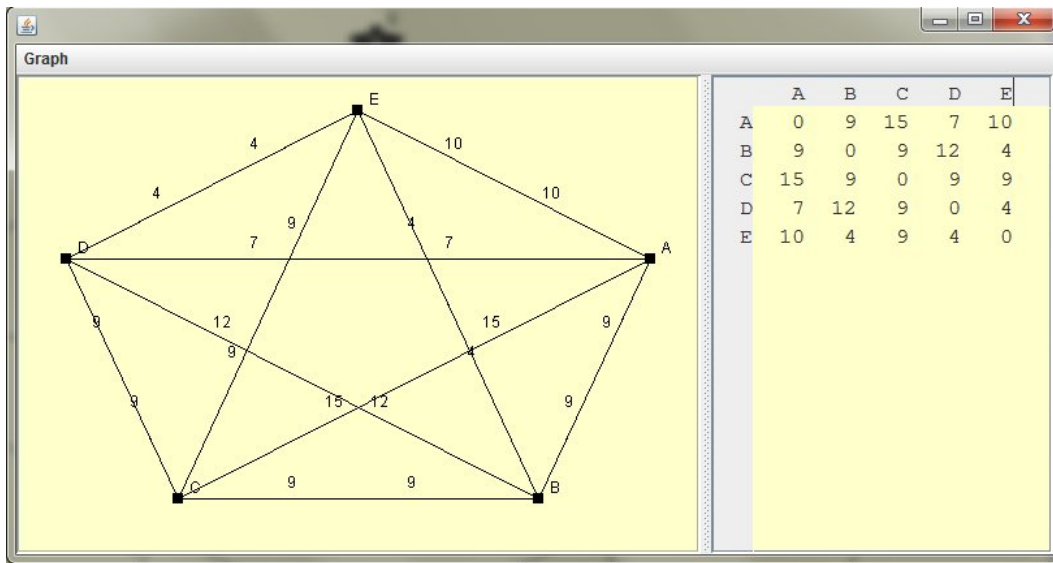
statistične podatke, ki se nanašajo na dani transportni problem (glej Sliko 10.27).

Sledi prikaz matrike sosednosti glede na razdalje. S klikom na dano možnost (*Adjacency Matrix*) se izriše tabela, prikazana na Sliki 10.28.

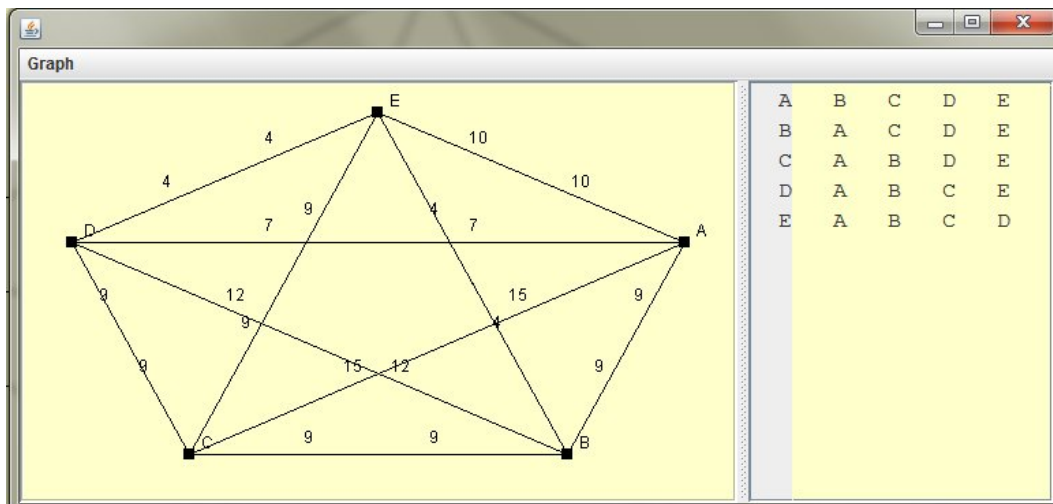
Slika 10.29 prikazuje izpis matrike sosednosti glede na povezave (*Adjacency List*).

Z danim grafom ustvarjamo in izpišemo rezultate, glede na številne možnosti, ki jih ponuja programsko orodje. Ena izmed njih je izpis kromatičnega števila (*Chromatic Number*) (glej Sliko 10.30).

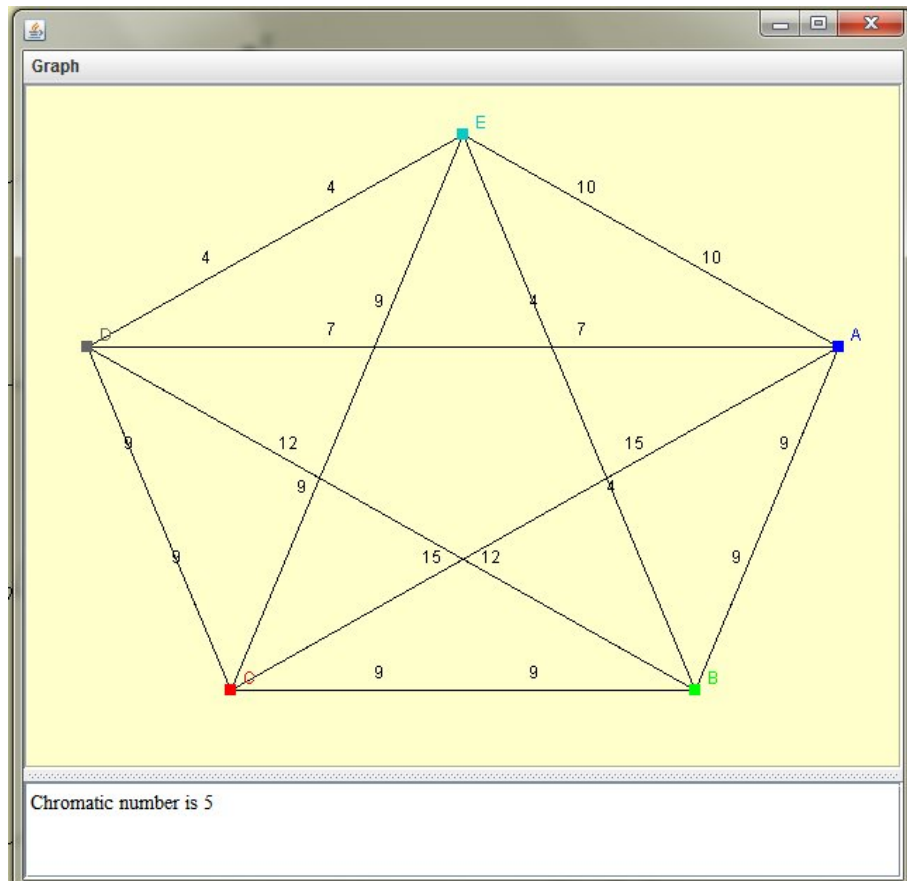
Seveda programsko orodje ponuja še številne druge možnosti, izhajajoče iz teorije grafov, npr.: Hamiltonov cikel, Eulerjev graf. Hamiltonova pot je po teoriji grafov pot v neusmerjenem grafu, ki gre skozi vsako točko na grafu natanko enkrat. Če sta začetna in končna točka poti enaki, jo imenujemo Hamiltonov cikel. Ime je dobila po irskem matematiku William Rowan Hamilton. V danem primeru se le ta ne izriše saj je graf usmerjen. Teorija Eulerjevega grafa temelji na dejstvu, da gremo po vsaki točki natanko enkrat in zaključimo na isti točki. Obhod je torej Eulerjev, če vsebuje vsako povezavo grafa natanko enkrat (in se zaključi v začetni točki).



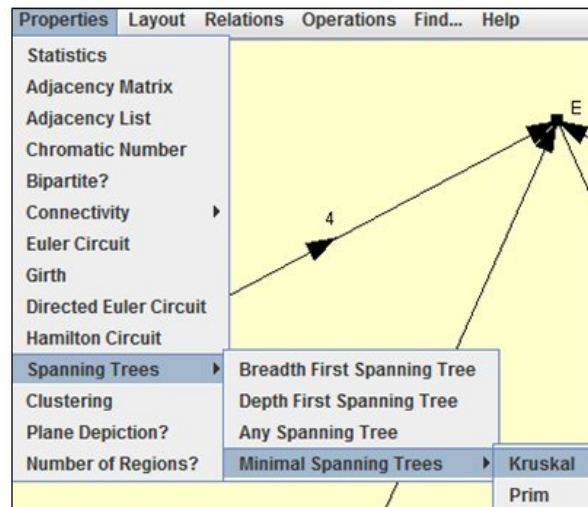
Slika 10.28: Matrika sosednosti glede na vrednosti



Slika 10.29: Matrika sosednosti glede na povezave



Slika 10.30: Kromatično število



Slika 10.31: Minimalno vpeto drevo

Iskanje najkrajše poti

V nadaljevanju se osredotočimo na problem iskanja najkrajše poti. V praksi poznamo dva algoritma, ki jih podrobneje opišemo v samem uvodu. Problem voznika, ki v podjetje dostavlja platišča je lahko povsem preprost. Prevoz opravi s tovorim vozilom, pri čemer upošteva razdalje med lokacijami.

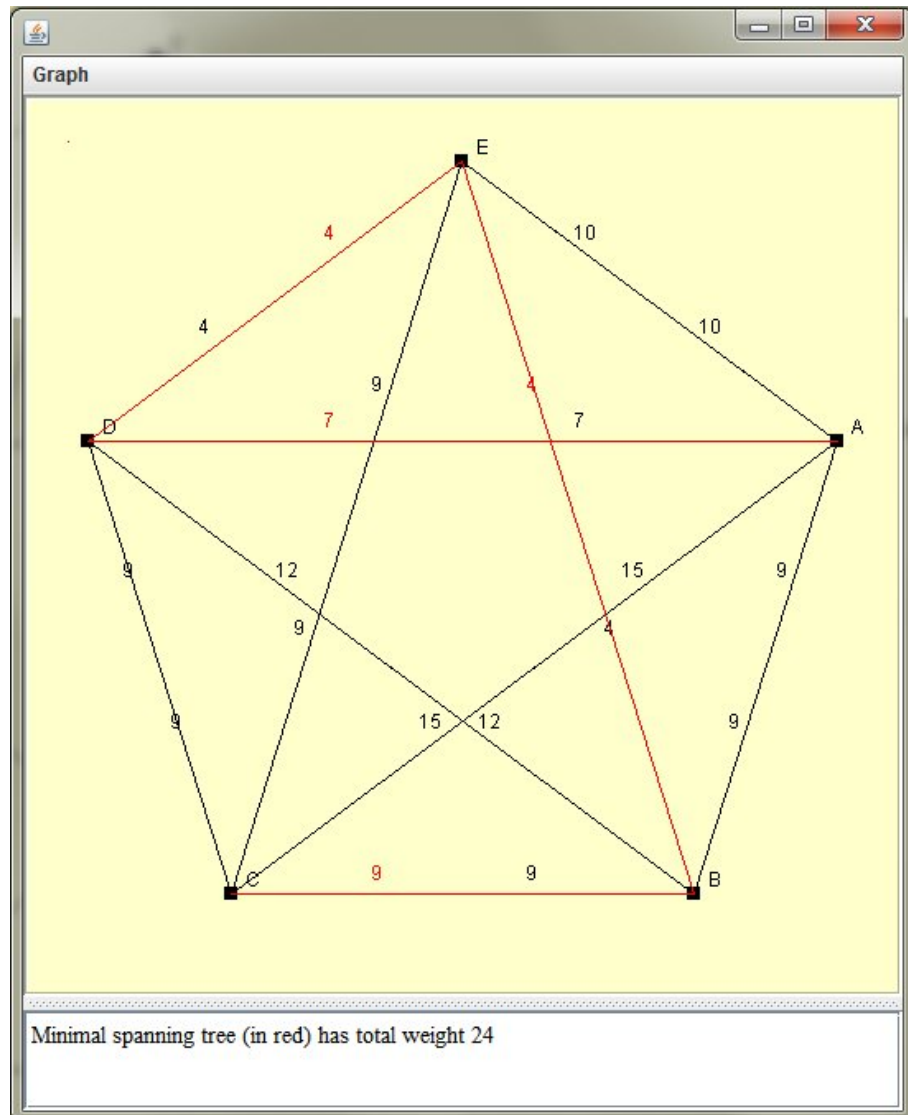
V menijski vrstici *Properties* izberemo razdelek *Spanning Trees*, v nadaljevanju pa *Minimal Spanning Trees*. Odločimo se med dvema možnostima - reševanje s Kruskalovim ali Primovim algoritmom (glej Sliko 10.31).

Na Sliki 10.32 prikazujemo izračun minimalnega vpetega drevesa s Kruskalovim algoritmom. Programsko orodje glede na izbrane lokacije prikaže, da je minimalno vpeto drevo oz. minimalno število km, ki jih voznik tovornjaka opravi 24. Minimalno vpeto drevo lahko poiščemo tudi s Primovim algoritmom, ki v danem primeru izriše enako pot. Običajno poti nista enaki. V danem primeru imamo na voljo manjše število povezav, kar pomeni, da je možnosti drugih poti bistveno manjša.

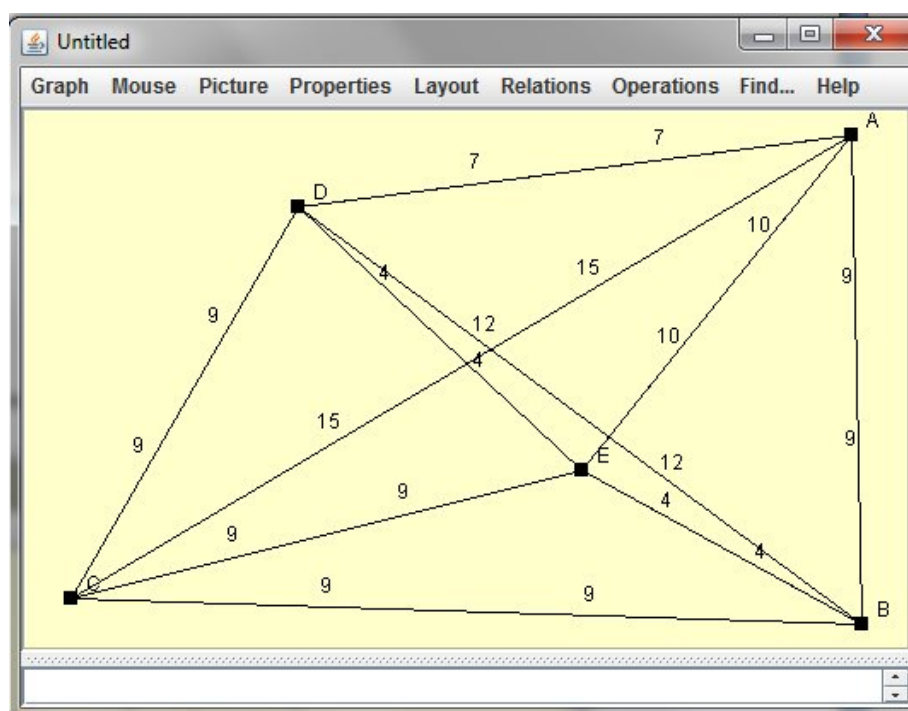
Druge možnosti programa

Menijska vrstica vsebuje še meni različnih postavitev programa (*Layout*), kar je prikazano na Sliki 10.33.

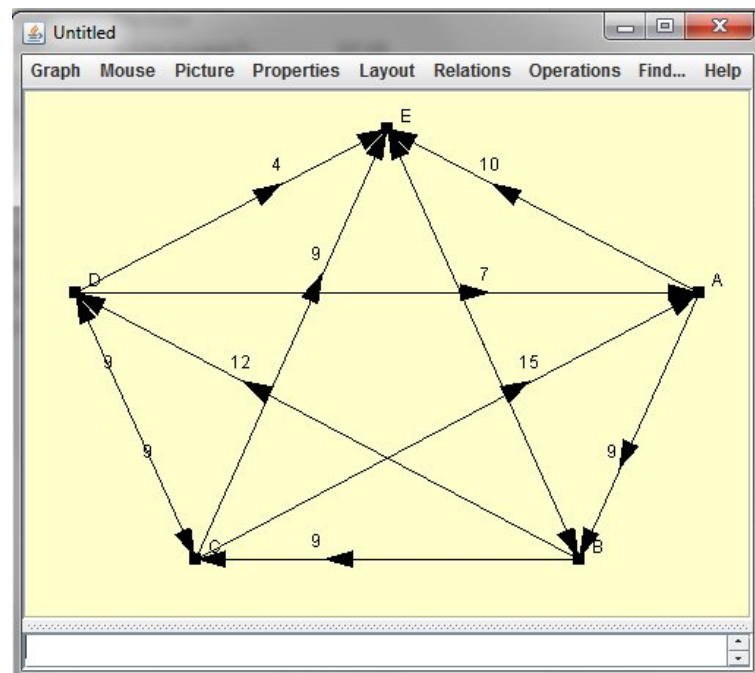
Menijska vrstica *Relations* prikaže razmerja med grafi. Če imamo več grafov (npr. 2), lahko ta med sabo primerjamo. Pogledamo ali sta



Slika 10.32: Minimalno vpeto drevo – Kruskalov algoritem



Slika 10.33: Postavitev grafa



Slika 10.34: Graf z omejitvami

izomorfna (*Isomorphism*) in ugotovimo ali sta subgrafa (*Subgraph*). Menijska vrstica *Operation* omogoča možnost prikaza komplementa narisane grafa (*Complement*), linijskega grafa med dvema točkama (*Line Graph*), prizme (*Prism*) ipd.

Menijska vrstica *Find* z razdelkom *Search* omogoča pregled grafa v globino in širino. Pomen iskanja točk na takšen način bo zagotovo znan znanstveniku, ki se podrobneje ukvarja s teorijo grafov. Pregled grafa v širino (*Breadth First*) je algoritem, ki začne v izbrani točki, katero pregleda. Vse "sosede" da v FIFO podatkovno strukturo in si zapomni h komu spada posamezna točka. Nato vzame prvega iz FIFO strukture in ponovi postopek. Postopek se konča, ko je FIFO struktura prazna. Pregled grafa v globino (*Depth First*) je algoritem, ki začne v izbrani točki in jo pregleda. Prav tako da vse sosede v LIFO podatkovno strukturo. Zapomni si h komu spada posamezna točka. Vzame prvega iz LIFO strukture in ponovi postopek. Konča, ko je LIFO struktura prazna. Obstaja še možnost iskanja najkrajše poti na podlagi Dijkstrovega algoritma (*Shortest Path* → *Dijkstra*). V danem algoritmu iščemo najkrajšo pot od V_1 do vseh ostalih točk. V izbranem primeru ni razlik v določitvi poti. Če bi imeli postavljen model z omejitvami, bi lahko le te tudi upoštevali (glej Slika 10.34).

Povzetek

Petersen programsko orodje je učinkovito, praktično, enostavno in brezplačno s pomočjo katerega izdelujemo, urejamo in manipuliramo z enostavni grafi in preučujemo njihove lastnosti. Njegovo uporabo priporočamo vsem, ki se ukvarjajo z danim področjem. Z njegovo pomočjo na enostaven in hiter način preverimo rezultate, ki smo jih pridobili z analitičnim izračunom.

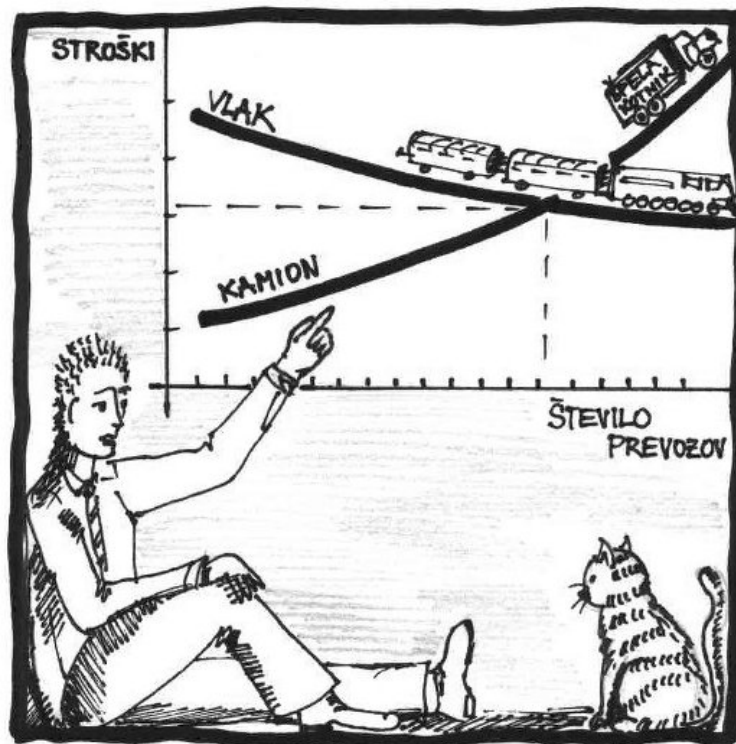
Prikažemo podatke o grafu, npr. število točk in njihove stopnje, matriko sosednosti, število sestavnih delov; preverimo, če je graf dvostranski, ali sta dva grafa izomorfna, če je graf podgraf drugega, poiščemo najkrajše poti ipd. Program omogoča pridobitev risbe grafov, do katerih je drugače skoraj nemogoče priti, uporaba animacije pa omogoča enostaven prikaz nekaterih pojmov.

Z izbranim programskim orodjem na podlagi teorije grafov prikažemo izračun optimalne transportne poti iz lokacije A do lokacije E in nekaj drugih možnosti, ki jih ponuja programsko orodje.

Pri opisu programskega orodja Petersen smo uporabili še dodatne vire in literaturo: [89].

Poglavje 11

LINDO - optimizacija stroškov



Linearno in celoštevilsko programiranje
Optimizacijski model
Primer: optimizacija stroškov izbire transporta

11.1 Teoretično ozadje

11.1.1 Linearno programiranje

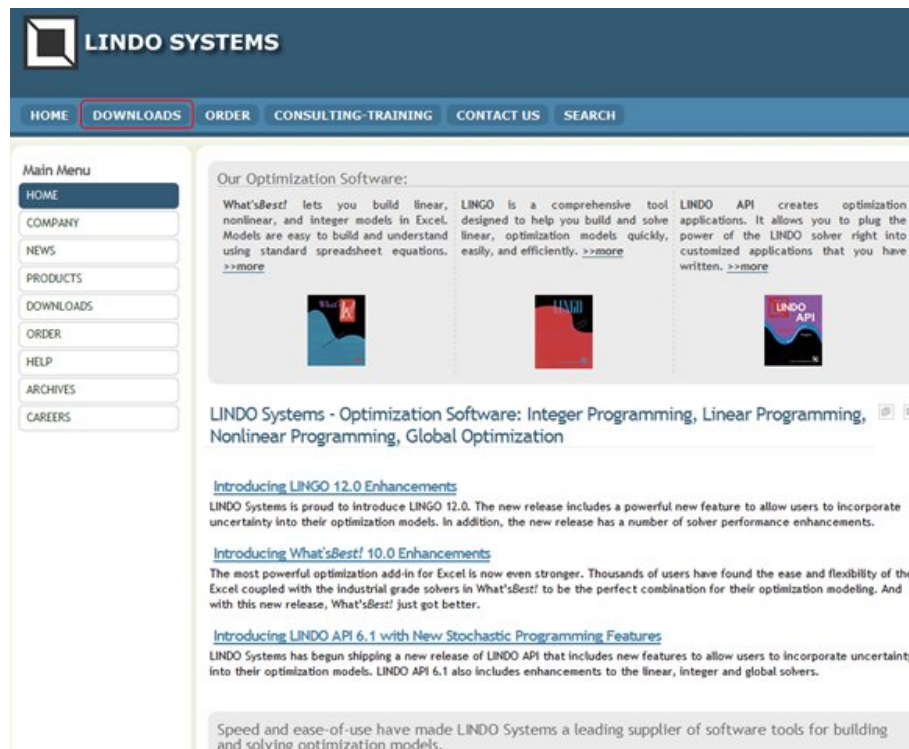
Linearno programiranje (v nadaljevanju LP) je področje matematike, ki se ukvarja s problemom optimizacije z omejitvami. Predstavlja specifičen razred optimizacijskih problemov, kjer maksimiziramo (minimiziramo) linearno funkcijo, pri čemer upoštevamo linearne omejitve. Pobudnik razvoja linearnega programiranja (1930) je Leonid Kantorovic, z metodo reševanja problema planiranja proizvodnje. Za začetnika linearnega programiranja velja George B. Dantzig, ki je leta 1947 razvil metodo simpleksov, pomembno vlogo pa ima tudi John von Neumann, ki je istega leta postavil temelje teorije dualnosti [70].

V Združenih državah Amerike se je linearno programiranje razvilo med drugo svetovno vojno z namenom, da reši zapletene probleme načrtovanja logistike vojaških operacijah. Doprinos k razvoju linearnega programiranja prištevamo ekonomistu Tjalling Koopmansu (rojen na Nizozemskem leta 1940, pozneje preseljen v Združene države Amerike). Matematik Kantorovic in ekonomist Koopmans sta leta 1975 dobila Nobelovo nagrado za ekonomijo - za prispevke k teoriji optimalne izrabe sredstev, kjer je linearno programiranje igralo glavno vlogo. Veliko industrijskih podjetij uporablja linearno programiranje kot standardno orodje (npr. za optimalno razporejanje končnih sredstev).

Gre torej za zelo pogosto uporabljeno metodo pri reševanju optimizacijskih problemov z omejitvami. V samem postopku ločimo tri pomembne korake, in sicer [90]:

- formulacija problema (postavitev problema v pravilni obliki);
- rešitev (izračun optimalnih možnosti);
- senzitivnostna analiza (kaj bi se zgodilo, če bi se pogoji našega zastavljenega problema malo spremenili).

Linearno programiranje je metoda za iskanje optimalne namenske funkcije, ko so omejitve (in namenska funkcija) dane v obliki sistema linearnih neenačb. Rešitev takšnega sistema je mogoče dobiti v grafični obliki, le v primeru, ko imamo zgolj dve odločitveni spremenljivki, sicer pa je potreben računski postopek. Splošna računsko metoda za reševanje linearnega programiranja je metoda simpleksov, ki jo je leta 1984 razvil matematik George B. Dantzig.



Slika 11.1: Prenos programa Linda – 1. del

vendar ni prosto dostopen, je pa eden izmed najbolj znanih in svetovno razširjenih programov za reševanje problemov linearnega programiranja. Oba programa sta nastala pod okriljem organizacije Lindo Systems.

Prenos in namestitve

Program Lindo prenesemo s spletnega mesta Lindo [27], kjer iz zgornjega modrega menija izberemo *Downloads* (glej Sliko 11.1).

Ob kliku na dano možnost se pojavi možnost namestitve štirih različnih verzij - izbiramo med plačljivimi in brezplačnimi. Na dnu seznama se nahaja možnost prenosa klasičnega programa Lindo (*Download Classic LINDO*), katero izberemo (glej Sliko 11.2).

Programsko okno

Ob zagonu programa Lindo se odpre osnovno okno (Slika 11.3), ki je sestavljeno iz naslovne vrstice (1), menijske vrstice (2), orodne vrstice (3) in "okenca" za

LINDO SYSTEMS

HOME DOWNLOADS ORDER CONSULTING-TRAINING CONTACT US SEARCH

Main Menu
 HOME
 COMPANY
 NEWS
 PRODUCTS
 DOWNLOADS
 ORDER
 HELP
 ARCHIVES
 CAREERS

Download trial versions of our products

From this page, you can download and try out all our software products FREE of charge. The trial versions have all the features and functionality of the standard versions, but the problem capacities have been limited. These trial versions allow you to set up and solve small problems, become familiar with the software and all of its features, and make sure you make the right choice.

Download What'sBest!
 What'sBest! lets you build linear, nonlinear, and integer models in Excel. Models are easy to build and understanding standard spreadsheet equations. Excel users can often begin building models within minutes of installing the What'sBest! program, and the problem capacities of the larger What'sBest! versions allow large scale, real world problems to be solved.

Download LINGO
 LINGO is a comprehensive tool designed to help you build and solve linear, nonlinear, and integer optimization models quickly, easily, and efficiently. LINGO includes a powerful modeling language, a full-featured environment for building and editing problems, the ability to read and write to Excel and databases, and a set of fast built-in solvers.

Download LINDO API
 With the LINDO API, you can easily create your own optimization applications. It allows you to plug the power of the LINDO solver right into customized applications that you have written.

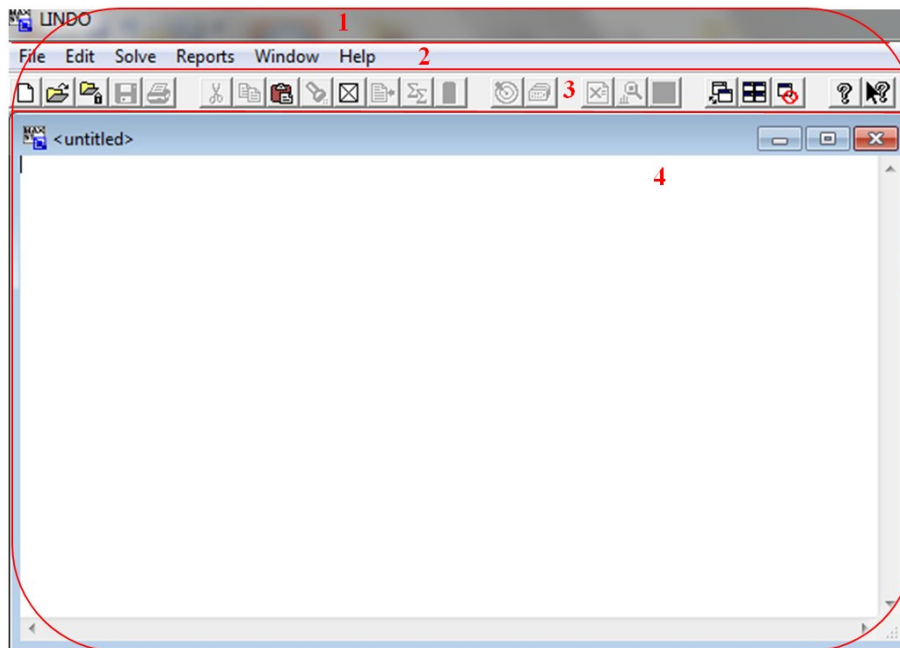
Download Classic LINDO
 Classic LINDO's simple interface and straightforward model expression style make it easy to learn and use. Classic LINDO is appropriate for building and solving moderately sized linear and integer models.

Trial Version Capacities:

	Constraints	Variables	Integer Variables	Nonlinear Formulas	Global Variables
Classic LINDO	150	300	30	N/A	N/A
LINDO API	150	300	30	30	5
LINGO	150	300	30	30	5
What'sBest!	150	300	30	30	5

LINDO Documentation:
[Click here](#) to download *Optimization Modeling with LINGO* by Linus Schrage.

Slika 11.2: Prenos programa Linda – 2. del



Slika 11.3: Osnovno okno ob zagonu

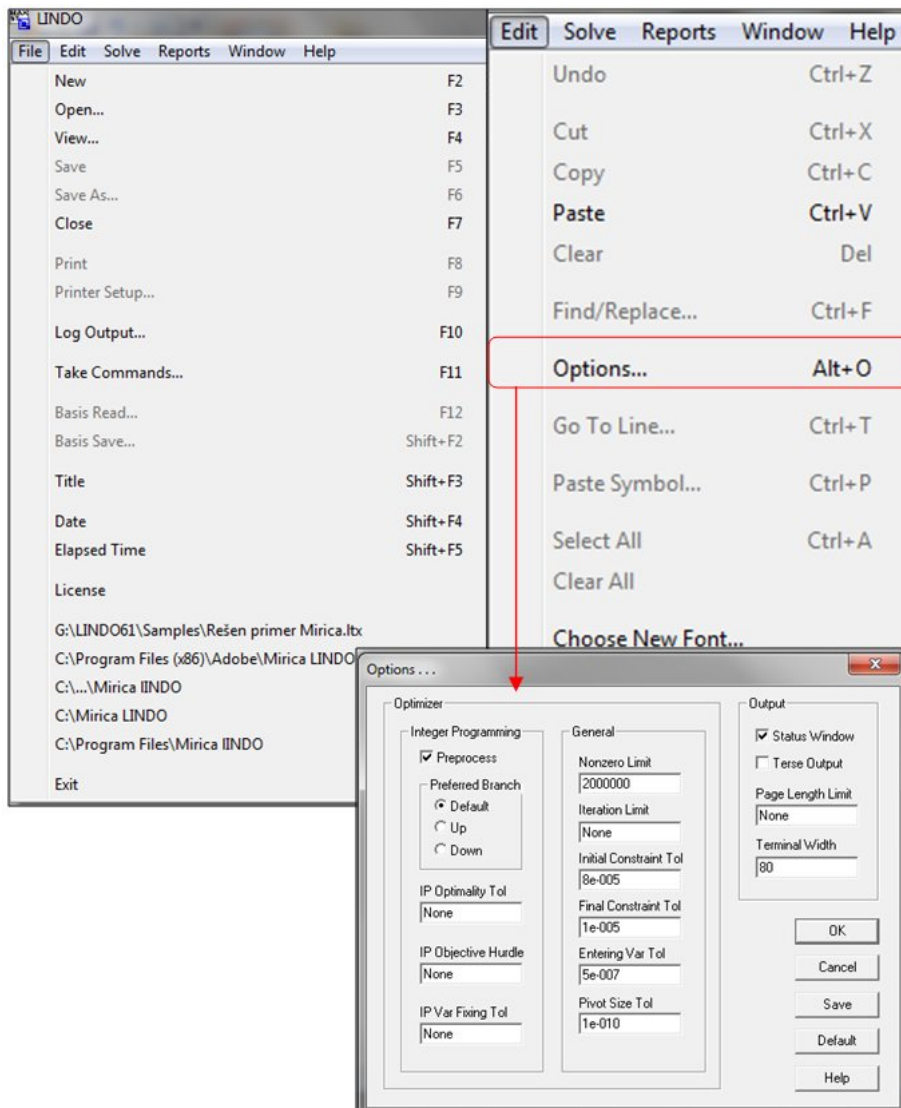
zapis funkcije (4).

Menijska vrstica vsebuje 6 različnih menijev. Meni *Datoteka (File)* omogoča odpiranje novega delovnega lista (*New*), odpiranje že shranjenega dokumenta na določenem mestu (*Open*), zapiranje dokumenta (*Close*) ipd. Ob koncu menijev se izpišejo možnosti odpiranja predhodnih dokumentov, ne da bi jih bilo potrebno poiskati na že shranjenih mestih. Meni *Urejanje (Edit)* omogoča rezanje (*Cut*), kopiranje (*Copy*) in lepljenje (*Paste*) določenega besedila oz. modela. Meni *Options* omogoča spreminjanje nastavitvev za celoštevilsko programiranje (*Integer Programming*), splošne nastavitve (*General*) in izhodne nastavitve (*Output*) (glej Sliko 11.4).

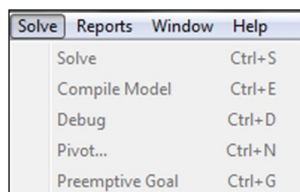
Sledi meni *Reševanje (Solve)*, katerega uporabimo, ko imamo model že zapisan. S klikom na meni se izpišejo rezultati (glej Sliko 11.5).

Meni *Poročilo (Reports)* uporabljamo, ko je model že izdelan. Meni *Okno (Windows)* omogoča ogled in odpiranje posameznih oken. V meniju *Pomoč (Help)* izbiramo med možnostmi izbire pomoči (*Search for Help On. . .*, *How to Use Help*) in posodabljanjem programa (*AutoUpdate*) (glej Sliko 11.6).

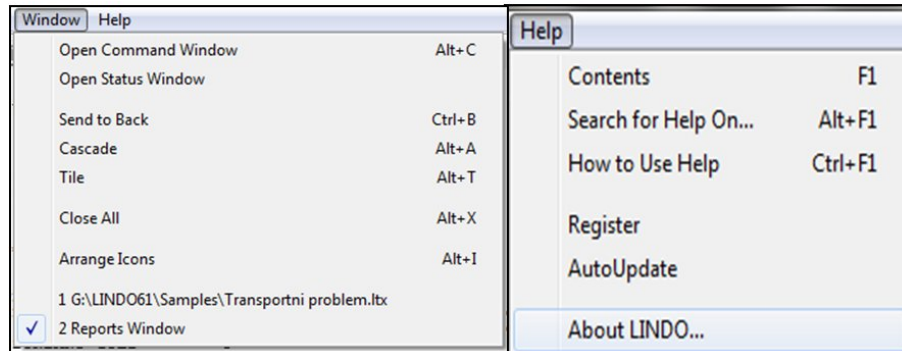
V orodni vrstici se nahajajo bližnjice, potrebne za nadaljnje delo. Prikazana so osnovna orodja za odpiranje novega oz. že shranjenega dokumenta, možnost ponovnega shranjevanja in tiskanja (glej Slike 11.7 in 11.8).








Slika 11.4: Menija Datoteka in Urejanje



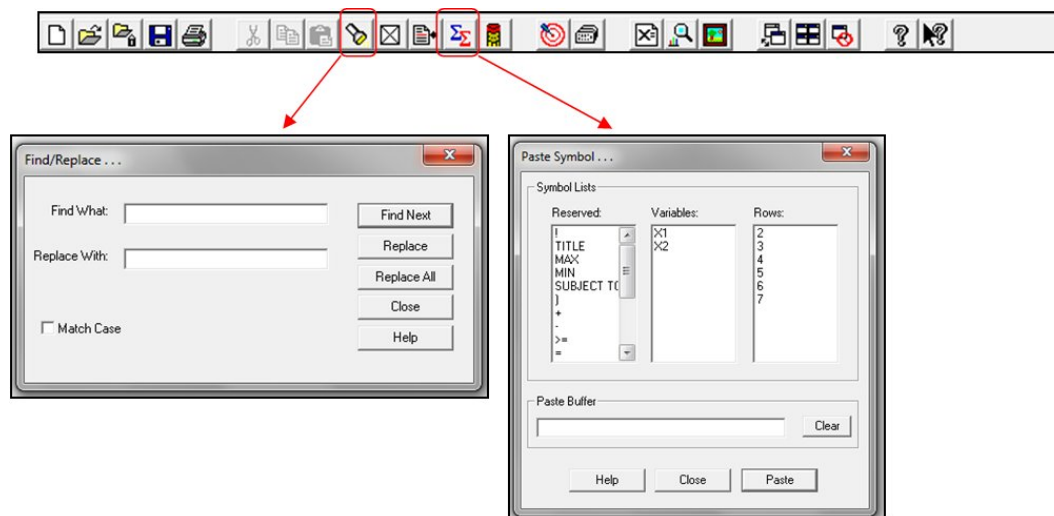
Slika 11.5: Meni Solve



Slika 11.6: Meni Window in Help

	Spreminjanje podatkov v modelu
	Nastavitev celoštevilskega programiranja
	Dodajanje simbolov
	Zapis modela in pregled rešitev
	Pomoč

Slika 11.7: Bližnjice (pomen)



Slika 11.8: Orodna vrstica

Problem

V podjetju OpenStorage, potrebujemo, glede na potrebe proizvodnje 15.410 platišč, ki jih mora dobavitelj dostaviti v času 5 dni. Prvotno je določeno, da se platišča dostavljajo zgolj s tovornjakom, vendar preizkusimo tudi možnost dostavljanja z vlakom. Z eno vožnjo bi po železnici (z vlakom) pripeljali maksimalno 800 kosov platišč, po cesti (s tovornjakom) pa 390 kosov, pri čemer upoštevamo določene omejitve.

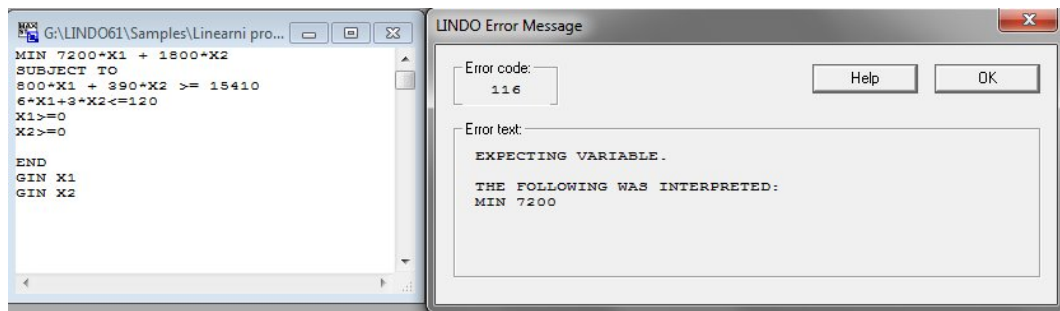
Čas, ki ga porabimo za eno vožnjo z vlakom, od dobavitelja v tujini do našega namišljenega podjetja, znaša 6 ur. Čas, ki ga porabimo za vožnjo s tovornjakom po cesti znaša 3 ure, pri čemer transport ne sme trajati več kot 120 ur (torej manj ali enako 5 dni). Enkratni stroški, ki nastanejo pri prevozu z vlakom znašajo 7.200 €, stroški transporta s tovornjakom pa 1.800 €.

Omejitve, ki jih je upoštevamo pri danem problemu so naslednje: vrednosti za spremenljivki železnica in cesta morata biti pozitivni in celi števili.

11.3 Uporaba

Zapis linearnega modela

Izračun transportnega modela z analitičnim izračunom bi bil dolgotrajen, hkrati pa bi se pojavilo večje število možnosti prikaza napak, zato se raje "poslužimo" izračuna s programskim orodjem. V primeru, da imamo model že zapisan, ga



Slika 11.9: Primer napačno zapisanega modela

enostavno vnesemo v programsko orodje Lindo. Zapišemo dani transportni model.

Linearni program

```

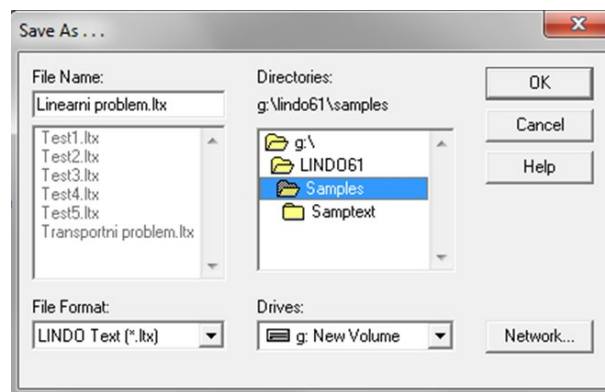
Min Z = 7200x1+1800x2 ----- namenska funkcija
800x1+390x2>=15.410
6x1+3x2<=120
x1>=0
x2>=0
x1,x2 element celega števila - pogoji in omejitve

```

Če smo že programirali s pomočjo programskega orodja Lingo je situacija povsem podobna, le z določenimi manjšimi popravki, kar prikažemo na konkretnem izmišljenem problemu. Programsko orodje Lingo natančno opozori in usmeri na napake, ki so pri modeliranju nastale. Lindo pa sporoči zgolj napako brez podrobnejšega opisa.

Prednost programskega orodja Lindo je v tem, da model zapišemo skoraj povsem podobno, kakor na list papirja. Ob zapisu namenske funkcije dodamo ukaz SUBJECT TO in na koncu zapisa END, kar pomeni konec. GIN X1 in GIN X2 sta ukaza s pomočjo katerih program izračuna celoštevilsko vrednost spremenljivk. Več o ukazih razložimo proti koncu priročnika.

V "okence" zapišemo dani linearni problem z vsemi pogoji in omejitvami. Pojavi se napaka. Okvirček, ki pove za kakšno napako gre, zgolj opisuje v katerem polju se nahaja določena napaka. Nahaja se v zapisu namenske funkcije, pa tudi v zapisu pogojev in omejitev (glej Sliko 11.9).



Slika 11.10: Shranjevanje datoteke

Ob ponovnem preverjanju in preizkušanju modela ugotovimo, da je ena izmed napak zapis vrednosti = za MIN, kar v programu Lindo ni dovoljeno. Prav tako ni dovoljeno dodajanje * (znak za množenje) med vrednostjo in izbrano spremenljivko. Ko popravimo napake, program ustrezno shranimo.

Linearni program

```

MIN 7200X1 + 1800X2
SUBJECT TO
800X1 + 390X2 >= 15410
6X1+3X2<=120
X1>=0
X2>=0

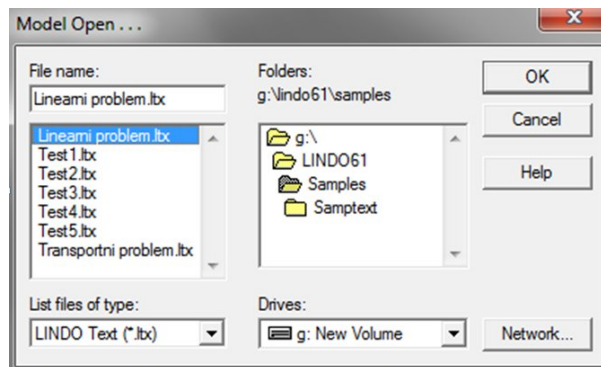
END
GIN X1
GIN X2

```

Zapisan model shranimo na želeno mesto na disku. V meniju *File Name* se izpiše kratica pred katero zapišemo ime datoteke (*Linearni problem*). Shranimo jo na poljubno mesto, v formatu LINDO Text (.ltx) (glej Sliko 11.10).

Že shranjeno datoteko odpremo tako, da v meniju *File - Open*, izberemo disk, kjer je datoteka shranjena (*Drives*). Nato iz seznama glede na ime datoteke (*File name*) izberemo datoteko (glej Sliko 11.11).

Po zapisu programa potrebujemo še rešitve problema. Ob pravilnem zapisu modela v orodni vrstici izberemo možnost *Solve*. V naslednjem okencu se izpišejo rezultati.



Slika 11.11: Odpiranje datoteke

Izpis rezultatov

LP OPTIMUM FOUND AT STEP 1
 OBJECTIVE VALUE = 71123.0781

NEW INTEGER SOLUTION OF 72000.0000 AT BRANCH 0 PIVOT 3
 BOUND ON OPTIMUM: 72000.00
 ENUMERATION COMPLETE. BRANCHES= 0 PIVOTS= 3

LAST INTEGER SOLUTION IS THE BEST FOUND
 RE-INSTALLING BEST SOLUTION...

OBJECTIVE FUNCTION VALUE

1) 72000.00

VARIABLE	VALUE	REDUCED COST
X1	0.000000	7200.000000
X2	40.000000	1800.000000

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	190.000000	0.000000
3)	0.000000	0.000000
4)	0.000000	0.000000
5)	40.000000	0.000000

NO. ITERATIONS= 3
 BRANCHES= 0 DETERM.= 1.000E 0

Prikazani rezultati povedo naslednje:

- Linearni problem programiranja je poiskan v prvem koraku (LP OPTIMUMFOUND AT STEP 1). Zapisane so še druge vrednosti navezujoče na dani problem.

```

LP OPTIMUM FOUND AT STEP      1
OBJECTIVE VALUE =  71123.0781

NEW INTEGER SOLUTION OF  72000.0000    AT BRANCH      0 PIVOT      3
BOUND ON OPTIMUM:  72000.00
ENUMERATION COMPLETE. BRANCHES=      0 PIVOTS=      3

LAST INTEGER SOLUTION IS THE BEST FOUND
RE-INSTALLING BEST SOLUTION...

```

- Vrednost zapisana pod številko 1) 72.000,00 pove, da je minimalni strošek transporta 15.410 kosov platišč 72.000,00 €. Z linearnim programiranjem izračunamo, da platišč ne bomo transportirali z vlakom (0x), pač pa s tovornjakom, in sicer 40x (*Value*) ob danih pogojih. Spremenljivka X1 torej pomeni transport z vlakom in spremenljivka X2 transport s tovornjakom. *Reduced Cost* prikazuje stroške posameznega transporta.

OBJECTIVE FUNCTION VALUE

1)	72000.00	
VARIABLE	VALUE	REDUCED COST
X1	0.000000	7200.000000
X2	40.000000	1800.000000

- Dopolnilna spremenljivka *Slack or Surplus* v rešitvi pove, za koliko bi še bilo potrebno levo stran v omejitvenih neenačbah prvotnega linearnega programa povečati ali zmanjšati, da bi namesto neenačb dobili enačbe. To količino v primerih neenakosti "manj ali enako" imenujemo pomanjkanje (*Slack*), v neenačbah oblike "več ali enako" pa presežek (*Surplus*). Če je omejitev v linearnem programu podana z enačbo, sta presežek ali pomanjkanje vedno po vrednosti enaka nič. Za neenačbo

$$800x_1 + 390x_2 \geq 15.410$$

lahko rečemo, da ni potrebno dodajanje enot, da dobimo enačbo. Podobno lahko razložimo tudi za ostale vrednosti.

- Dualno ceno (*Dual Price*) razumemo kot tisto vrednost, za katero bi se spremenila namenska funkcija, če bi se omejitev v konkretni (ne)enačbi povečala za eno enoto. Namesto termina *Dual Price* uporabljamo tudi pojem senčna cena (*Shadow Price*), ker ta podatek pove, koliko bi bili pripravljeni plačati za povečanje omejitve za eno dodatno enoto. Senčna cena je vedno nenegativna (pozitivna ali nič) za omejitve tipa "manjše ali enako" in vedno nepozitivna (negativna ali nič) za neenačbe oblike "večje ali enako". Če so omejitve dane v obliki enačb, je senčna cena lahko negativna, pozitivna ali enaka nič. V našem primeru so vrednosti dualnih cen enake 0.

ROW	SLACK OR SURPLUS	DUAL PRICES
2)	190.000000	0.000000
3)	0.000000	0.000000
4)	0.000000	0.000000
5)	40.000000	0.000000

NO. ITERATIONS=	3
BRANCHES=	0 DETERM.= 1.000E 0

Kot dodatek je zapisan model v programu Lingo, ki prikazuje le nekaj razlik med zapisom v Lindu.

Zapis linearnega programa v Lindu

```
Zmin=7200* x1+1800*x2;  
800*x1+390*x2 >= 15410;  
6*x1+3*x2<=120;  
x1>=0;  
x2>=0;  
@gin(x1);  
@gin(x2);
```

Ob preučevanju literature ugotovimo, da za program Lindo ni popolnega priročnika v eni datoteki. Na spletu je dostopen priročnik za program Lingo in program Lindo Api s katerima si lahko pomagamo.

Uporabniku programa predlagamo ogled slednjih spletnih strani povezave, ki so v pomoč pri uporabi programskega orodja: Lindo - splet 1 [29], Lindo - splet 2 [28], Lindo - splet 3 [40].

Navedene spletne strani omogočajo odpiranje številnih datotek, kjer so opisani posamezni postopki uporabe programa, zapisi modelov ipd. (glej Sliko 11.12).

The screenshot shows a web browser window displaying the search results for 'tutorial lindo or' on the PDFDatabase.com website. The browser's address bar shows the URL 'http://pdfdatabase.com/search/tutorial-lindo-or.html'. The search bar contains the text 'tutorial lindo or' and has a dropdown menu set to 'All files' and an 'OK' button. Below the search bar, the page displays 'Pdf & Doc book search results 1 - 25 of 7769 for tutorial lindo or'. There are several 'Related Searches' listed, including 'french learning book or tutorial', 'photoshop in 24 hours tutorial or ebook', 'cielito lindo piano score', and 'drupal 7 tutorial/'. The search results are sorted by 'Size' and 'Alphabet'. The first result is a sponsored link for 'tutorial lindo or [New Version]', 'tutorial lindo or [FullVersion]', '[HIGHSPEED] tutorial lindo or', and '[TRUSTED DOWNLOAD] tutorial lindo or'. The following results are individual PDF files with their sizes and download links:

File Name	Size	Download Link
Lindo.pdf	24 KB	http://www.icaen.uiowa.edu/~Private/lindo.pdf
Lindo.doc	276 KB	http://www.uniovi.es/~Descargas/lcde/lLindo.doc
Lindo.pdf	163 KB	http://www.ismp.minihub.org/~Lindo.pdf
2968f-Lindo.pdf	543 KB	http://www.ecoscience.ulaval.ca/~admin/imagephp/2968f-lindo.pdf
2968r-Lindo.pdf	586 KB	http://www.ecoscience.ulaval.ca/~admin/imagephp/2968r-lindo.pdf
Lindo-08.pdf	197 KB	http://www.timberlake.co.uk/~Lindo-08.pdf
Lindo-Aca08.pdf	173 KB	http://www.timberlake.co.uk/~Lindo-Aca08.pdf

Slika 11.12: Spletna stran za prenos datotek, ki so v pomoč pri uporabi programa Lindo

Povzetek

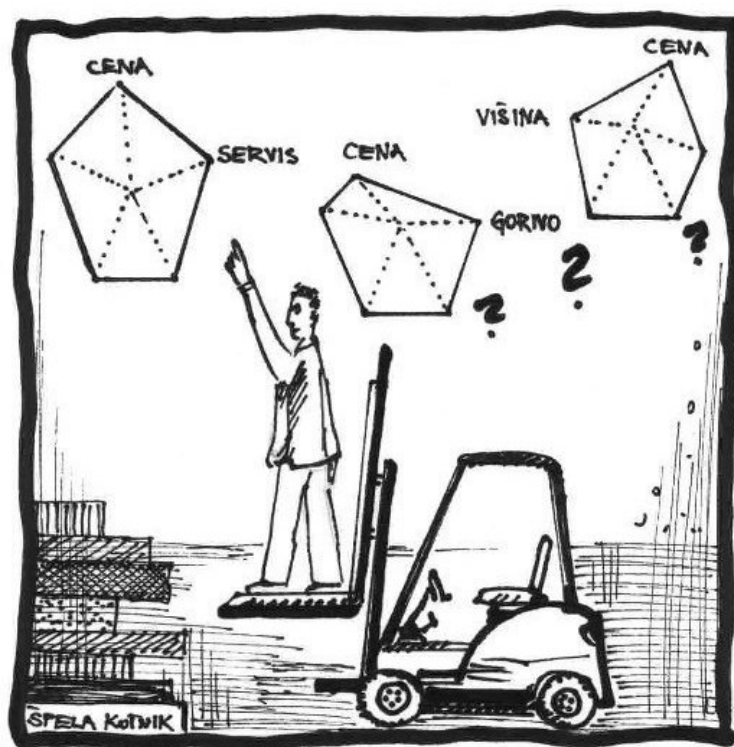
Programsko orodje Lindo je prosto dostopno orodje za reševanje problemov linearnega programiranja, izdelano pri organizaciji Lindo Systems. Njihova programska orodja uporabljajo številna podjetja po vsem svetu; za povečanje dobička in zmanjšanje stroškov, za odločitve, ki vključujejo načrtovanje proizvodnje, prevoza, financ, portfolio dodeljevanja kapitala proračuna, mešanje, načrtovanje, inventar, dodeljevanje sredstev in več. Predhodno pripravljen model omogoča enostaven zapis in pregled rešitve v Lindo. Še posebej je primeren za študente, ki se z dano problematiko srečujejo pri študiju ali v praksi. Ni zahteven, vendar za pravilno delovanje potrebuje predhodno znanje iz področij optimizacije in linearnega programiranja.

Z izbranim programskim orodjem predstavimo problematiko izbire transporta 15.410 platišč, ki jih je potrebno dostaviti v času 5 dni. Izdelamo model, ki prikazuje, katero prevozno sredstvo, ob izbranih omejitvah, povzroča nižje stroške transporta. Z izdelanim modelom dokažemo, da je v času maksimalno 120h prevoz možno opraviti s tovornjakom (40x vožnja), s skupnimi stroški v znesku 72.000 €. Ugotovimo, da vožnja z vlakom povzroča bistveno večje stroške.

Pri opisu programskega orodja Lindo smo uporabili še dodatne vire in literaturo: [95].

Poglavje 12

DEXI - odločitveni model



Večparametrski odločitveni model
Programsko orodje DEXi
Primer: izbira najboljšega viličarja

12.1 Teoretično ozadje

Odločanje opredelimo kot proces izbire najboljše variante, ki ustreza zastavljenim pogojem in kriterijem [53]. Za izbiro so potrebne najmanj dve ali več variant, med katerimi lahko izberemo najprimernejšo, najugodnejšo, najkoristnejšo. S pomočjo večparametrskega odločitvenega modela skušamo v prvi vrsti izbrati najboljšo varianto, v drugi pa s pomočjo pridobljenih podatkov potrditi in utemeljiti zakaj je izbrana varianta najboljša. Cilji, ki jih skušamo doseči so: predstavitev variant, kriterijev in njihova natančna opredelitev (utežitev posameznih kriterijev), na podlagi katerih izboljšamo najboljšega med njimi.

Vsakodnevne situacije in spremembe trendov pri poslovanju zahtevajo od logistika nenehno prilagajanje in upravljanje procesov, ki so vitalnega pomena za nemoteno in učinkovito izvajanje delovanja bodisi dela poslovnega procesa bodisi celotne oskrbne verige. Zagotavljanje informacij o spremembah in njihova nadaljnja transformacija zagotavlja možnost takojšnjega ukrepanja v realnem času. Vsaka sprememba izzove dejanje, ki ga ni moč predvideti, zato je lahko predhodno planiranje možnih scenarijev odločilnega pomena. Prav tako je za celotno poslovanje pomembna pravilna odločitev v pravem času. Sprememba, kakršna je npr. okvara viličarja ali drugega transportnega sredstva, nas prisili k takojšnji odločitvi oz. izbiri novega transportnega sredstva, saj le tako zagotovimo nemoteno delovanje poslovnega procesa. Zagotovitev nemotenega oz. neprekinjenega poslovanja zahteva od logistika, da predvidi možne situacije in da ob nastopu le teh pravočasno ukrepa.

Ob predpostavki, da v fazi skladiščenja vsakodnevno uporabljamo za logistične manipulacije tri viličarje na plinski pogon, kateri nemoteno delujejo 24 ur in je njihova obremenitev 100 %, je tveganje za okvaro enega izmed njih zelo veliko. Obseg dela v fazi skladiščenja se na letni ravni povečuje za 10 % v primerjavi s predhodnimi leti. Stroški tveganja so v primerjavi s stroški nakupa novega viličarja manjši, zato je ob nadaljevanju trenda rasti obsega delovanja potrebno v podjetju na podlagi cene, kakovosti in uporabnosti izbrati novega viličarja.

Pri izbiri novega viličarja moramo upoštevati načela uporabnosti transportnega sredstva, kakovost (vzdržljivost, okretnost, vodljivost) in ceno (stroški vzdrževanja, nabavna cena). Zaradi načina skladiščenja, ki zaradi specifik blaga poteka samo v zaprtih prostorih, izberemo viličarja na plinski ali električni pogon. Prav tako moramo upoštevati še ostale značilnosti kot so vodljivost, hitrost dviga in spusta, višina dviga, poraba goriva itd.

12.1.1 Večparametrski odločitveni model

V vsakdanjem življenju se neprestano soočamo s procesom odločanja. Odločitve sprejemajo za to odgovorni ljudje, ki upravljajo podjetje. Njihova primarna naloga je odločanje in sprejemanje odločitev - za te so tudi plačani. Ocenjevanje upravljavca in/ali upravljavske skupine se izvaja predvsem glede na kakovost odločitev v kompleksnih okoljih.

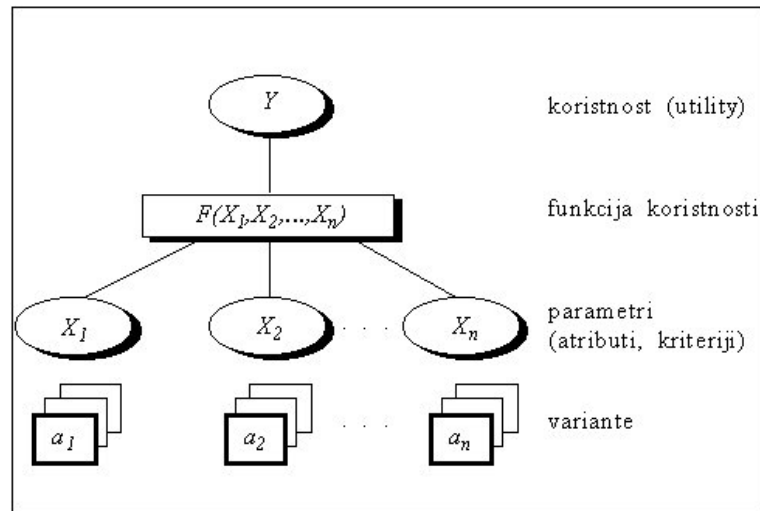
Odločanje je proces, v katerem je potrebno izmed več variant (alternativ, inaic, možnosti) izbrati tisto, ki najbolj ustreza postavljenim ciljem, oz. zahtevam [53]. Z odločanjem se srečujemo vsak dan, vsako uro. Primeri sprejemanja odločitev: izbira pravilne transportne poti, izbira najustrežnejšega ponudnika mobilne telefonije, izbira novega ustreznega avtomobila, izbira restavracije za poslovno kosilo ipd.

Zgornji primeri kažejo na to, da smo vsi neprestano vpeti v procese odločanja. Nekatere odločitve so za nekoga pomembnejše, spet druge so manj pomembne; ene imajo dolgoročne posledice, spet druge niso bistvene; enih se lotevamo tako, da preračunavamo možnosti in posledice, spet drugih se lotevamo po "občutku". Pri odločanju se navadno pojavijo naslednja glavna vprašanja:

- kaj ali kdo sploh pride v izbor za ocenjevanje – možne *variante*;
- katere lastnosti bomo ocenjevali – možni *parametri*;
- kakšni so *kriteriji* za določitev vrednosti posameznih parametrov;
- kakšna so merila za ocenjevanje – *funkcija koristnosti*, iz katere dobimo končno oceno.

Večparametrsko odločanje temelji na razgradnji odločitvenega problema na manjše podprobleme. Variante razgradimo na posamezne parametre (kriterije, atribute) in jih ločeno ocenimo glede na vsak parameter. Končno oceno variante dobimo z nekim postopkom združevanja. Tako izpeljana vrednost je osnova za izbor najustrežnejše variante [53].

Vrednotenje variant pri večparametrskem odločanju poteka na osnovi *večparametrskega odločitvenega modela*, ki je v splošnem sestavljen iz treh komponent (Slika 12.1). Vhod v model predstavljajo *parametri (atributi, kriteriji) X_i* . To so spremenljivke, ki ponazarjajo podprobleme odločitvenega problema, se pravi tiste dejavnike, ki opredeljujejo kvaliteto variant. *Funkcija koristnosti F* je predpis, po katerem se vrednosti posameznih parametrov združujejo v spremenljivko Y , ki ponazarja *končno oceno* ali *koristnost* variante. *Variante* opišemo po osnovnih parametrih z vrednostmi. Na osnovi teh vrednosti funkcija koristnosti določi končno oceno vsake variante. Varianta, ki dobi najvišjo oceno, je praviloma najboljša [53] [80].



Slika 12.1: Večparametrski odločitveni model

Vir: [80]

V zahtevnejših primerih, ko je parametrov ali variant več (na primer nekaj deset), je navadno bolje, če posežemo po katerem izmed namenskih programskih orodij za podporo večparametrskega odločanja. Ti imajo že vgrajena orodja, ki odločevalcu pomagajo pri definiciji parametrov, oblikovanju funkcij koristnosti in zajemanju podatkov o variantah. Najpomembnejšo operacijo - vrednotenje variant - dodatno podpirajo z vrsto koristnih pripomočkov za analizo dobljenih rezultatov, kot so analiza občutljivosti in stabilnosti odločitvenega modela, generator variant, analize tipa *kaj-če* ter najrazličnejši grafični prikazi in poročila. Odločitveni proces je proces sistematičnega zbiranja in urejanja znanja. Zagotovil naj bi dovolj informacij za primerno odločitev, zmanjšal možnost, da bi kaj spregledali, pohitril in pocenil proces odločanja ter dvignil kakovost odločitve. Praviloma poteka po fazah [53]:

- identifikacija problema,
- identifikacija kriterijev,
- definicija funkcij koristnosti,
- opis variant,
- vrednotenje in analiza variant.

Identifikacija problema

Ta faza je rezultat spoznanja, da je nastopil odločitveni problem, ki je dovolj težak, da ga je smiselno reševati na sistematičen in organiziran način. V tej fazi poskušamo definirati problem ter opredeliti cilje in zahteve. Oblikujemo odločitveno skupino, katere jedro sestavljajo odločevalci (t.i. lastniki problema). Pri zahtevnejših problemih je priporočljivo v delo skupine vključiti tudi eksperte, odločitvenega analitika in druge predstavnike tistih segmentov na katere vpliva odločitev.

Identifikacija kriterijev

V tej fazi določimo kriterije, na osnovi katerih ocenjujemo variante in zasnujemo strukturo odločitvenega modela. Posebej pomembno je, da pri tem ne spregledamo kriterijev, ki bistveno vplivajo na odločitev (načelo polnosti).

Definicija funkcij koristnosti

V tej fazi definiramo funkcije, ki opredeljujejo vpliv nižjenivojskih kriterijev na tiste, ki ležijo višje v drevesu, vse do korena drevesa, ki predstavlja končno oceno variant. Najpogosteje se uporabljajo preproste funkcije, kot so utežena vsota in razna povprečja.

Opis variant

Vsako varianto opišemo z vrednostmi osnovnih kriterijev, to je tistih, ki ležijo na listih drevesa.

Vrednotenje in analiza variant

Vrednotenje variant je postopek določanja končne ocene variant na osnovi njihovega opisa po osnovnih kriterijih. Vrednotenje poteka od spodaj navzgor v skladu s strukturo kriterijev in funkcijami koristnosti. Varianta, ki dobi najvišjo oceno, je praviloma najboljša. Verč [84] pravi: "Vrednotenje alternativ poteka s pomočjo tristopenjskega procesa, ki skupaj sestavlja večparametrski odločitveni model."

Sistemi za podporo odločanju predstavljajo procese in tehnologije, ki podpirajo sprejemanje odločitev. To so računalniški informacijski sistemi, ki olajšajo proces odločanja [100]. Dodatno znanje o večparametrskem modeliranju in odločanju jemo pridobiti v naslednjih člankih:

- Članek 1;

- Članek 2;
- Članek 3.

12.2 O programskem orodju

Z uporabo programskega orodja DEXi predstavimo odločitveni model, s katerim določimo optimalni izbor za nakup viličarja. Program DEXi je namenjen delu z odločitvenimi modeli in omogoča:

- izdelavo in preurejanje drevesa kriterijev;
- urejanje zalog vrednosti kriterijev in odločitvenih pravil;
- različne variante in njihovo vrednotenje;
- tabelarni pregled rezultatov;
- grafični prikaz rezultatov.

Programsko orodje DEXi je izobraževalno-računalniški program za večparametrsko odločitve. Namen programa je interaktiven razvoj kvalitativnih odločitev večparametrskih modelov in vrednotenje njihovih možnosti. Uporaba programa je odlična podpora pri kompleksnih odločitvah, saj lahko osnovni kompleksni problem razčlenimo na manjše podprobleme, katere je enostavneje rešiti [5].

Osnovni nalogi DEXi sta [5]:

- razvoj kvalitativnih večparametrskih modelov in
- uporaba modelov za vrednotenje in analizo.

V fazi ocenjevanja in analize DEXi omogoča [5]:

- opis možnosti: določitev vrednosti osnovnih atributov;
- oceno možnosti: združevanje vrednosti "od spodaj navzgor";
- analizo možnosti: kaj-če analizo, "plus-minus-1" analizo, selektivno razlago in primerjavo možnosti;
- poročanje: grafična in tekstovna predstavitev modelov, možnosti in vrednotenja rezultatov.

Ostale programske opreme [5]:

- DEX je predhodnik DEXi;
- JDEXi je odprtokodni program za izvajanje, razčlenjevanju modelov DEXi in oceno možnosti. Namenjen uporabnikom Linux (operacijski sistem Ubuntu, Fedora itd.);
- DEXiTree je program za risanje dreves DEXi;
- DEXiEval je program za oceno serije možnosti uporabe modela DEXi.

Programsko orodje DEXi uporabimo za reševanje izbire kompleksnih problemov, kadar ti vsebujejo [5]:

- veliko atributov;
- veliko variant;
- kvalitativno presojo;
- skupinsko odločanje s potrebno komunikacijo in razlago.

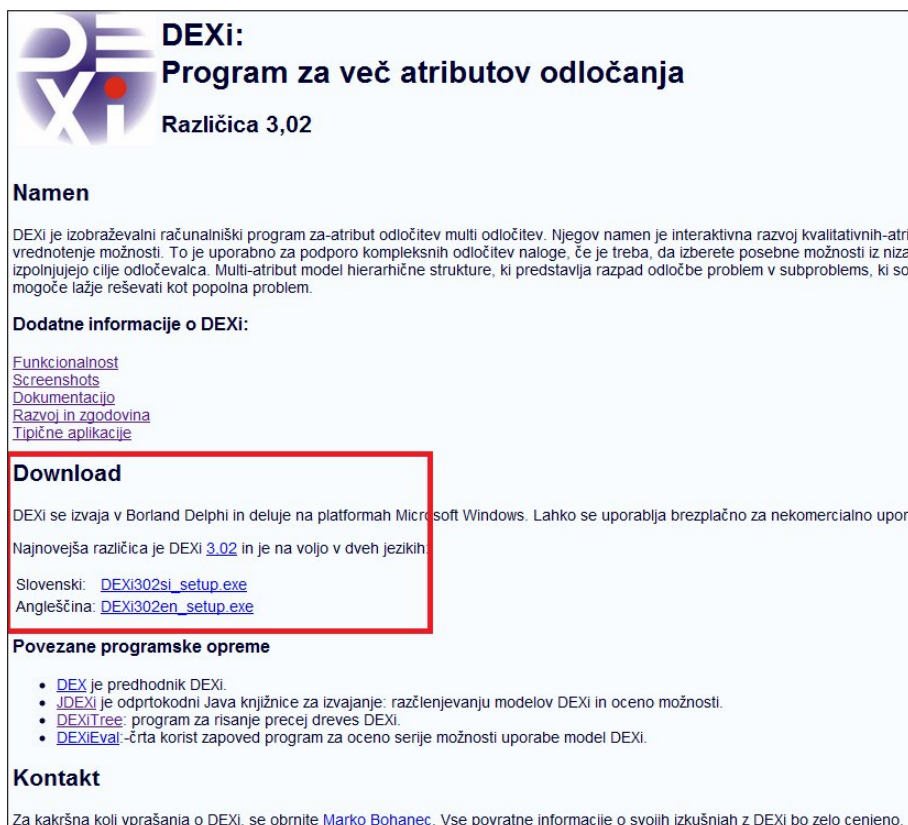
Nekatera področja, kjer se DEXi uporablja [5]:

- informacijska tehnologija (razvoj računalnikov, programov in spletnih portalov);
- projekti (razvoj projektov, investicije, portfolijo ipd.);
- podjetništvo (izbira partnerjev);
- medicina (tveganja, diagnoze, prognoze).

Prenos in namestitev

Programsko orodje DEXi je dostopno na spletnem naslovu DEXi [5]. Po končanem prenosu ga moramo še namestiti. DEXi lahko uporabljajo tako uporabniki operacijskega sistema Windows kot uporabniki operacijskega sistema Ubuntu. Na izbiro je slovenska in angleška verzija programa. Je brezplačen za lastno uporabo (nekomercialne zadeve).

Ustvarjalci programskega orodja DEXi ponujajo tudi odprtokodno rešitev DEXi - JDEXi pod GNU GPL licenco, vendar brez jamstva o delovanju. Programsko orodje JDEXi poiščemo pod ostalo programsko opremo, s klikom na JDEXi se pojavi nova spletna stran JDEXi [23]. S klikom na JDEXi.zip, datoteko prenesemo na računalnik (Slika 12.2).



DEXi:
Program za več atributov odločanja
Različica 3,02

Namen

DEXi je izobraževalni računalniški program za-atribut odločitev multi odločitev. Njegov namen je interaktivna razvoj kvalitativnih-atribut vrednotenje možnosti. To je uporabno za podporo kompleksnih odločitev naloge, če je treba, da izberete posebne možnosti iz niza izpolnjujejo cilje odločevalca. Multi-atribut model hierarhične strukture, ki predstavlja razpad odločbe problem v subproblems, ki so mogoče lažje reševati kot popolna problem.

Dodatne informacije o DEXi:

[Funkcionalnost](#)
[Screenshots](#)
[Dokumentacija](#)
[Razvoj in zgodovina](#)
[Tipične aplikacije](#)

Download

DEXi se izvaja v Borland Delphi in deluje na platformah Microsoft Windows. Lahko se uporablja brezplačno za nekomercialno uporabo.

Najnovejša različica je DEXi [3.02](#) in je na voljo v dveh jezikih:

Slovenski: [DEXi302si_setup.exe](#)
Angleščina: [DEXi302en_setup.exe](#)

Povezane programske opreme

- [DEX](#) je predhodnik DEXi.
- [JDEXi](#) je odprtokodni Java knjižnice za izvajanje: razčlenjevanju modelov DEXi in oceno možnosti.
- [DEXITree](#): program za risanje precej dreves DEXi.
- [DEXiEval](#):-črta korist zapoved program za oceno serije možnosti uporabe model DEXi.

Kontakt

Za kakršna koli vprašanja o DEXi, se obrnite [Marko Bohanec](#). Vse povratne informacije o svojih izkušnjah z DEXi bo zelo cenjeno.

Slika 12.2: Prenos programskega orodja DEXi



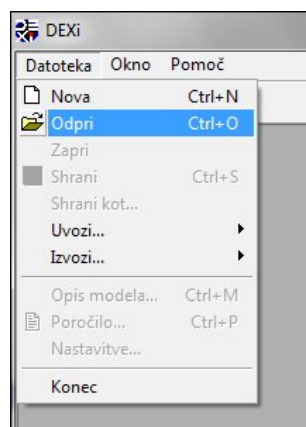
	Nov dokument
	Odpri shranjeni dokument
Datoteka Okno Pomoč	Orodna vrstica z orodji

Tabela 12.1: Orodja



Slika 12.3: Kako odpremo že ustvarjeni dokument

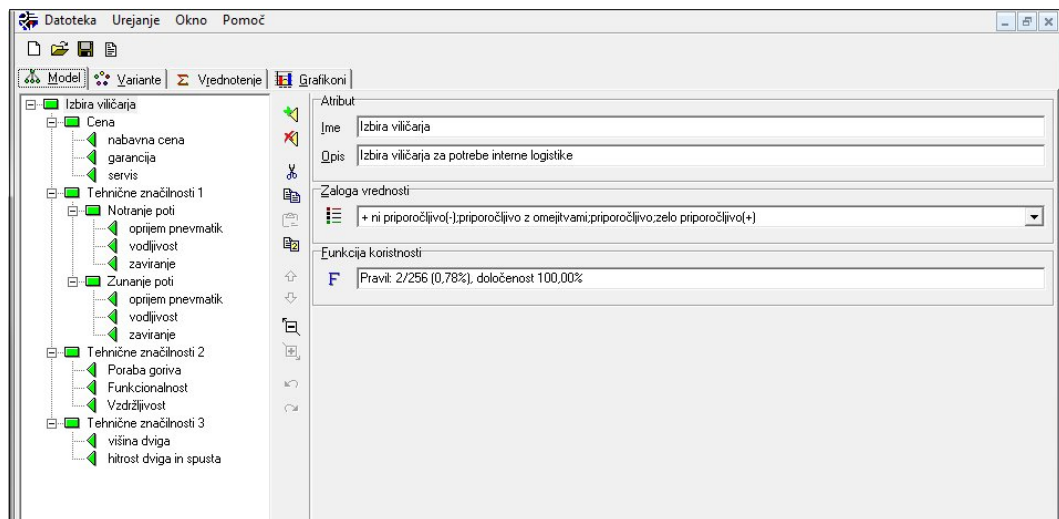
Preden pričnemo z uporabo, moramo (po končanem prenosu) programsko orodje še namestiti. V našem primeru izberemo programsko orodje DEXi, zaradi nekomercialnega namena uporabe in jamstva o delovanju, katerega JDEXi žal ne zagotavlja.

Za začetek je potrebno poznati različne funkcije in orodja, ki jih sam program omogoča.

Ob zagonu programa se v začetnem oknu pojavijo naslednja orodja (Tabela 12.1).

V kolikor smo že izdelali večparametrski odločitveni model, ga poiščemo in odpremo v razdelku *Odpri* (Slika 12.3). Izbrani model nato še dopolnujemo in preoblikujemo (izbira kriterijev, določitev uteži kriterijem, izbira variant itd.). V kolikor šele začenjamo z uporabo programa DEXi in izdelavo odločitvenega modela, je potrebno le tega še ustrezno nastaviti (določiti kriterije, variante itd.). Uporabo programskega orodja DEXi prikažemo na primeru reševanja izbranega odločanja izbire viličarja, ki ga potrebujemo za procese skladiščenja in manipulacije blaga (Slika 12.4).

Rezultat vrednotenja različnih variant, funkcij in kriterijev je odvisen od določitve drevesa kriterijev. Paziti je potrebno na obseg kriterij (drevo kriterijev), saj kompleksnejši problemi, obsežnejšo drevo kriterijev in njihova razčlenitev




Slika 12.4: Primer večparametrskega modela za izbiro viličarja

zahteva večje število možnosti variant, kar pomeni slabšo primerjavo. Potrebno je jasno in natančno opredeliti kriterije, saj je kasnejše dopolnjevanje in spreminjanje lahko zamudno, pri čemer se tudi kaj hitro "izgubimo".

Zaključimo z izdelavo končnega poročila, s katerim predstavimo vsa dejstva o izbiri viličarja. S končnim poročilom upravičimo izbiro in nakup novega viličarja pri vodstvenemu managementu (Slika 12.5).

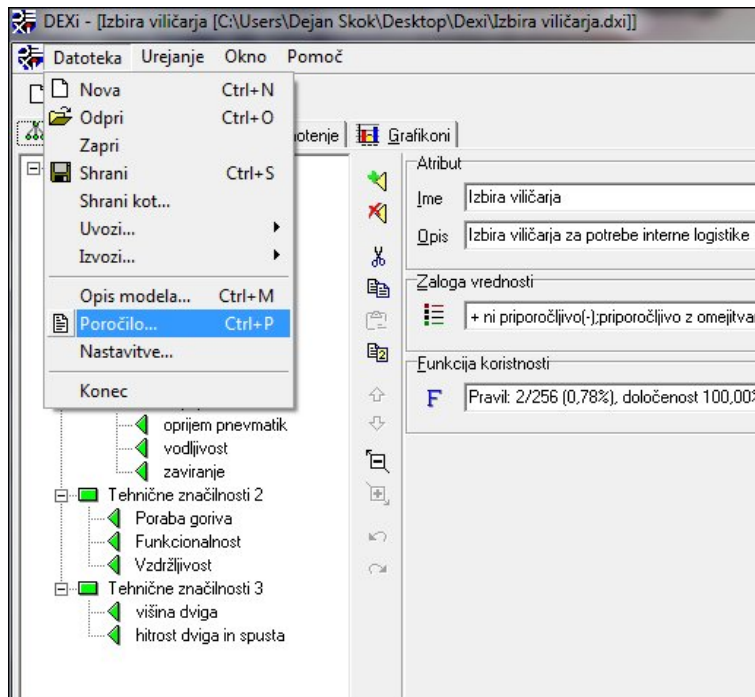
Zaključno poročilo si ogledamo ali natisnemo s klikom na razdelek *Datoteka* in *Poročilo*. Pri pripravi poročila je možno izbirati in poljubno nastaviti tehnične značilnosti poročila, kot so barva pisave, prikaz funkcij (Slika 12.6).

V razdelku *Datoteka* najdemo *Nastavitve*, kjer nastavimo obliko poročila (pisava, prikaz funkcij). Končni izdelek shranimo v poljubno mapo na disku v razdelku *Datoteka*, *Shrani* ali s klikom na ikono  (Slika 12.7).

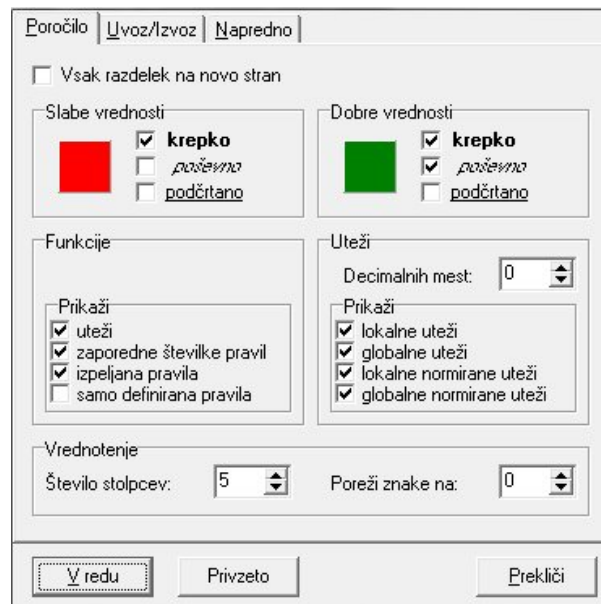
Ustvarjene datoteke je možno shraniti kot (Slika 12.8): .dxi dokument, .xml dokument, .dax dokument ipd. Različne možnosti oblik datoteke ponujajo kompatibilnost z drugimi programskimi orodji, npr. z odprtokodni program jDEX.

V menijski vrstici programa DEXi najdemo različne funkcije in orodja. Najpomembnejši so razdelki, *Datoteka* (za shranjevanje datotek, odpiranje datotek, izdelava zaključnega poročila itd.), *Urejanje* (dodajanje/odstranjevanje atributov-kriterijev, kopiranje, iskanje, zaloga vrednosti, funkcija koristnosti itd.), *Okno* (priprava pogleda) in *Pomoč* (navodila o uporabi programa) (Slika 12.9).

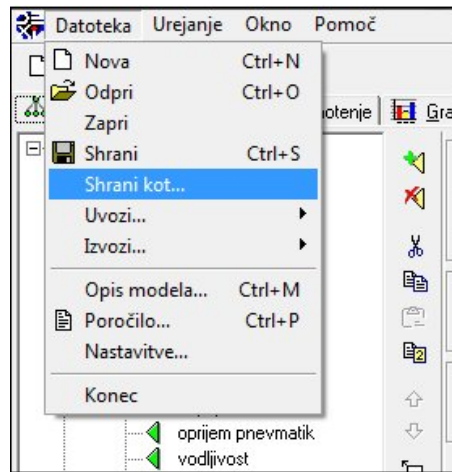
Dodatna navodila in pomoč pri uporabi programa DEXi najdemo v orodni



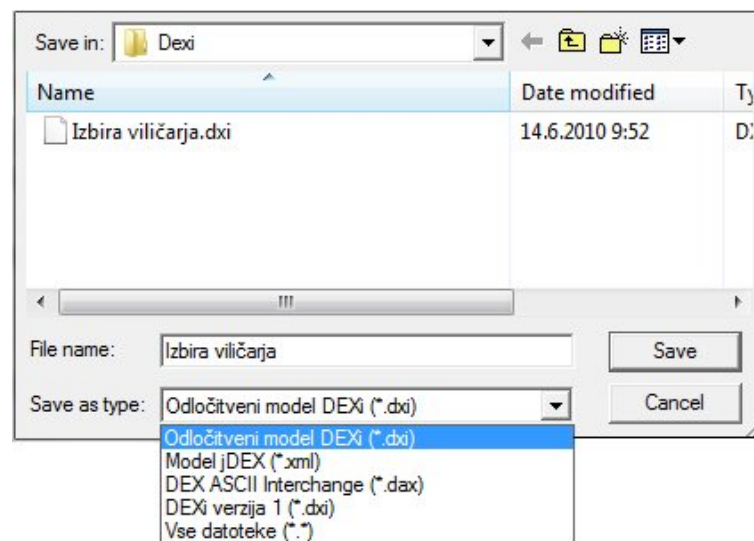
Slika 12.5: Izdelava poročila rezultatov vrednotenja



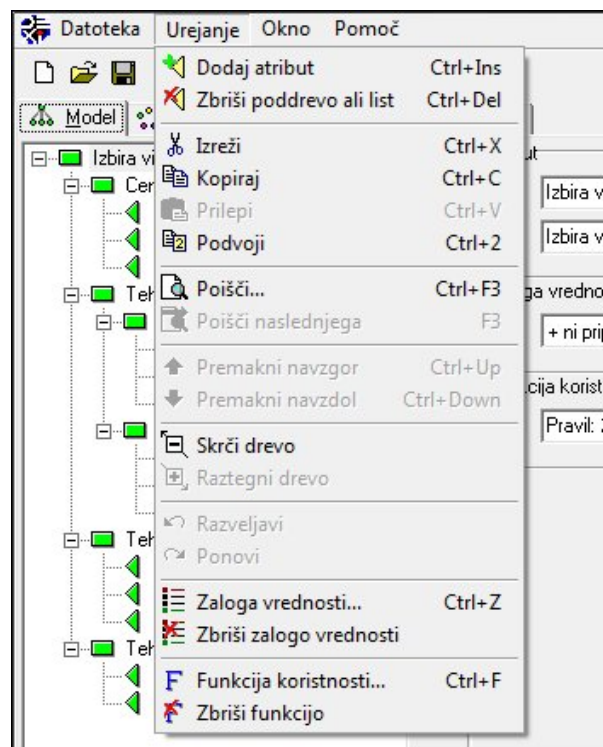
Slika 12.6: Nastavitve funkcij programskega orodja



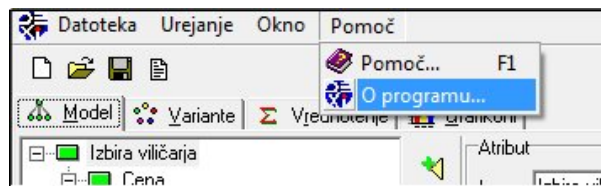
Slika 12.7: Kako shraniti datoteko



Slika 12.8: Možni načini shranjevanja



Slika 12.9: Funkcije razdelka Urejanje



Slika 12.10: Pomoč

vrstici pod razdelkom *Pomoč* oz. s klikom na tipko F11 (Slika 12.10).

Problem

V fazi skladiščenja vsakodnevno uporabljamo tri viličarje na plinski pogon, kateri nemoteno delujejo 24 ur s 100 % obremenitvijo. Zaradi povečanja obsega dela v fazi skladiščenja, je stopnja tveganja okvare visoka. S primerjavo stroškov tveganja in stroškov investicije v nakup sodobnega viličarja potrdimo smotrnost nakupa sodobnega viličarja, zato je potrebna hitra izbira najboljše variante. Izbiramo med tremi različnimi ponudbami z naslednjimi značilnostmi v Tabeli 12.2.

Za optimalnejšo izbiro najboljšega ponudnika uporabimo programsko orodje DEXi, kjer iz podanih podatkov izdelamo večparametrski odločitveni model.

Ponudba - Lastnosti	Viličar 1	Viličar 2	Viličar 3
Cena	26.500 €	22.000 €	18.700 €
Garancija	3000 ur	2000 ur	1000 ur
Servis	12 ur	24 ur	48 ur
Oprijem pnevmatik N	priporočljivo	priporočljivo	priporočljivo
Vodljivost N	zelo priporočljivo	priporočljivo	priporočljivo
Zaviranja N	priporočljivo	zelo priporočljivo	priporočljivo
Oprijem pnevmatik Z	zelo priporočljivo	zelo priporočljivo	zelo priporočljivo
Vodljivost Z	zelo priporočljivo	priporočljivo	priporočljivo
Zaviranje Z	zelo priporočljivo	zelo priporočljivo	priporočljivo
Funkcionalnost	priporočljivo	zelo priporočljivo	zelo priporočljivo
Vzdržljivost	zelo priporočljivo	priporočljivo	priporočljivo
Poraba goriva (v l)	12 litrov	16 litrov	20 litrov
Višina dviga teleskopa (v m)	6 m	5 m	5 m
Hitrost dviga in spusta (v m/s)	2,2 m/s	2 m/s	3 m/s

Opomba: Z-zunanje poti, N-notranje poti

Tabela 12.2: Osnovni podatki

12.3 Uporaba

Za izdelavo večparametrskega odločitvenega modela moramo najprej določiti: drevo kriterijev, attribute, zalogo vrednosti in funkcije koristnosti.

Potrebno je določiti kriterije, s katerimi ocenjujemo variante in zasnujemo strukturo odločitvenega modela. Pri tem ne smemo pozabiti na pomembnejše kriterije, ki bistveno vplivajo na izbiro variante, po načelu polnosti (Slika 12.11).

Kriterije za izbiro najboljšega viličarja opredelimo s ceno in tehničnimi značilnostmi. Zaradi kompleksnosti problema značilnosti razdelimo na tri podveje: "Tehnične značilnosti 1", "Tehnične značilnosti 2" in "Tehnične značilnosti 3". Takšna členitev kriterijev omogoča boljši pregled in nudi ugodnejšo rešitev (Slika 12.12).

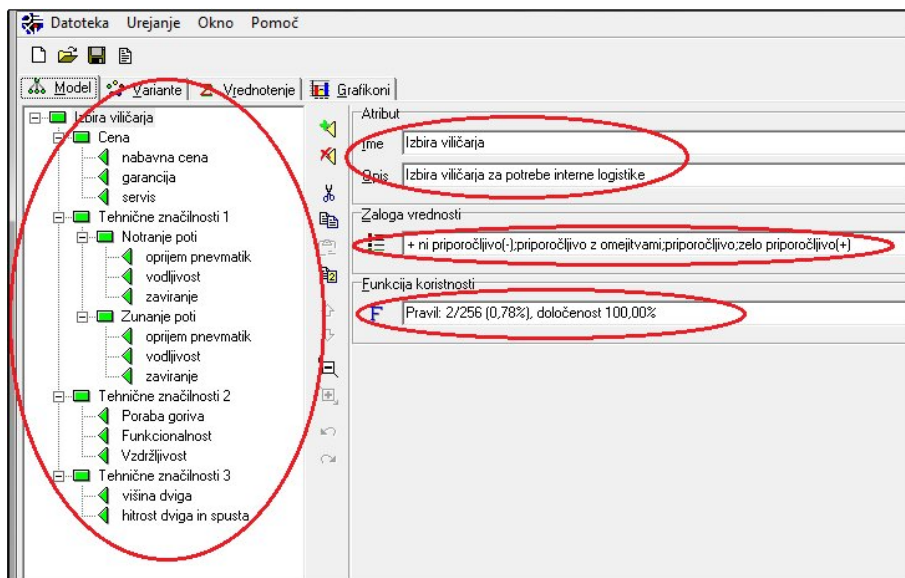
Vsakemu kriteriju dodamo težo (Slika 12.13), odvisno od pomembnosti. Predpostavljamo, da je v kriznem času potrebno optimizirati stroške, predvsem investicijske in se odločimo, da je kriterij cena najpomembnejši pri izbiri viličarja (Tabela 12.3). Pri tem zanemarimo dejstvo, kot je kakovost izdelka. Kriterije za izbiro viličarja smo utežili po naslednjem vrstnem redu v Tabeli 12.3. Prav tako utežimo še kriterije za "Tehnične značilnosti 1", "Notranje poti" in "Zunanje poti".

Atributi

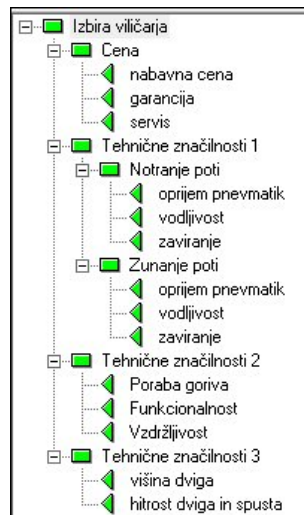
Atributi so lastnosti, na osnovi katerih ocenimo variante. V tem delu določimo imena atributov oz. kriterijev, katerim lahko dodamo še krajši opis (zaželen pri zaključnem poročilu) (Slika 12.14).

Cena	45 %
Tehnične lastnosti 1	18 %
Tehnične lastnosti 2	18 %
Tehnične lastnosti 3	18 %

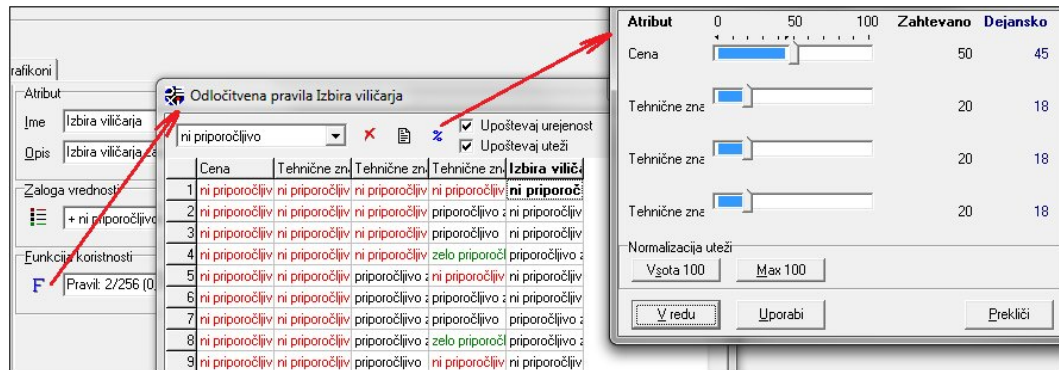
Tabela 12.3: Kriteriji za izbiro viličarja



Slika 12.11: Model (primer za izbiro viličarja)



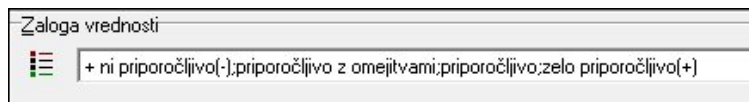
Slika 12.12: Drevo kriterijev



Slika 12.13: Določitev uteži kriterijev za izbiro viličarja

Atribut	
Ime	Izbira viličarja
Opis	Izbira viličarja za potrebe interne logistike

Slika 12.14: Vnos podatkov za attribute



Slika 12.15: Zaloga vrednosti

Zaloga vrednosti

Zaloga vrednosti je vrednost, ki jo zavzame kriterij. Vsakemu kriteriju določimo zalogo vrednosti, s klikom na ikono. Zaradi različnosti kriterijev so tudi zaloge kriterijev različne (vsebinsko in številčno). Za zalogo vrednosti kriterija cene smo npr. izbrali naslednje vrednosti (Slika 12.15 in 12.16):

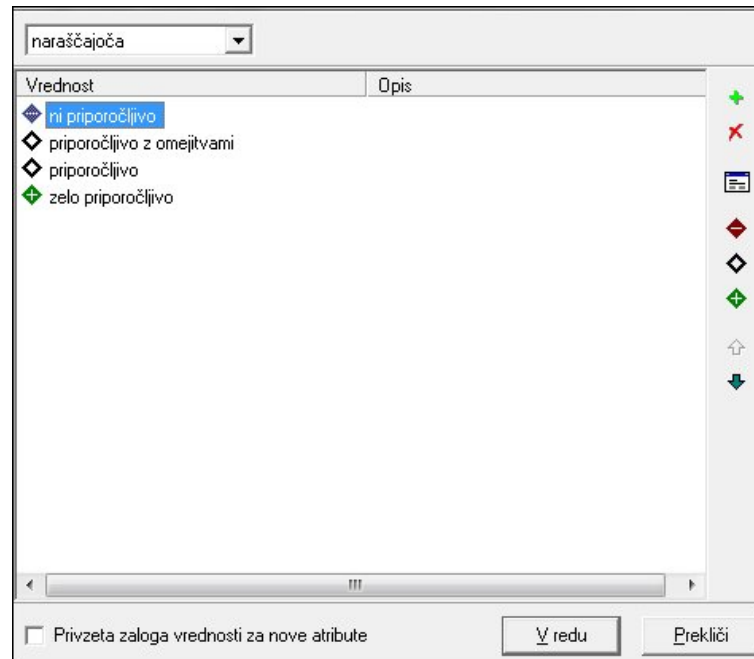
- ni priporočljivo;
- priporočljivo z omejitvami;
- priporočljivo;
- zelo priporočljivo.

Kriterij nabavne cene ocenjujemo številčno, in sicer:

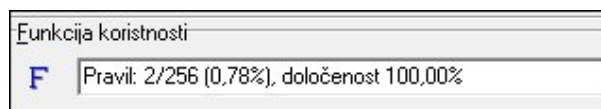
- od 25.000 € do 30.000 € (cena je previsoka (-) in ni sprejemljiva);
- med 20.000 € in 25.000 € (cena je normalna (+) in je sprejemljiva);
- do 20.000 € (cena je zelo ugodna (++) in je zelo sprejemljiva).

Funkcija koristnosti

Funkcija koristnosti je pri večparametrskem modelu odločanja sestavljena iz nižjenivojskih funkcij, ki vplivajo na višjenivojske funkcije. Seštevek vseh funkcij skupaj predstavlja končno oceno za posamezno varianto. Pri funkciji koristnosti moramo določiti odločitvena pravila s klikom na ikono **F**. Odločitvena pravila so lahko samostojno določena (po lastni presoji) ali pa z upoštevanjem urejenosti in uteži kriterijev (Slika 12.17 in 12.18).



Slika 12.16: Določanje zaloge vrednosti



Slika 12.17: Funkcija koristnosti

	Cena	Tehnične zn.	Tehnične zn.	Tehnične zn.	Izbira viliča
1	ni priporočljivo	ni priporočljiv	ni priporočljiv	ni priporočljiv	ni priporočljivo
2	ni priporočljivo	ni priporočljiv	ni priporočljiv	priporočljivo	ni priporočljivo
3	ni priporočljivo	ni priporočljiv	ni priporočljiv	priporočljivo	ni priporočljivo
4	ni priporočljivo	ni priporočljiv	ni priporočljiv	zelo priporočljivo	priporočljivo
5	ni priporočljivo	ni priporočljiv	priporočljivo	ni priporočljivo	ni priporočljivo
6	ni priporočljivo	ni priporočljiv	priporočljivo	priporočljivo	ni priporočljivo
7	ni priporočljivo	ni priporočljiv	priporočljivo	priporočljivo	priporočljivo
8	ni priporočljivo	ni priporočljiv	priporočljivo	zelo priporočljivo	priporočljivo
9	ni priporočljivo	ni priporočljiv	priporočljivo	ni priporočljivo	ni priporočljivo
10	ni priporočljivo	ni priporočljiv	priporočljivo	priporočljivo	priporočljivo
11	ni priporočljivo	ni priporočljiv	priporočljivo	priporočljivo	priporočljivo
12	ni priporočljivo	ni priporočljiv	priporočljivo	zelo priporočljivo	priporočljivo
13	ni priporočljivo	ni priporočljiv	zelo priporočljivo	ni priporočljivo	priporočljivo
14	ni priporočljivo	ni priporočljiv	zelo priporočljivo	priporočljivo	priporočljivo

Slika 12.18: Odločitvena pravila za funkcijo koristnosti






	določitev vrednosti
	dodajanje nove variante
	odstranjevanje variante
	izreži varianto
	podvoji varianto

Tabela 12.4: Orodja pri opredeljevanju variant

Variante

Variante so možnosti, med katerimi se odločamo. Za različne variante se odločimo na podlagi raziskave trga ponudbe viličarjev. Tako v ožji izbor uvrstimo tri variante:

- Viličar 1;
- Viličar 2;
- Viličar 3.

Na podlagi pridobljenih tržnih podatkov in opisa izdelkov, vse tri variante opredelimo z enakimi kriteriji:

- Cena;
- Tehnične značilnosti 1;
- Tehnične značilnosti 2;
- Tehnične značilnosti 3.

Orodja pri opredeljevanju variant v Tabeli 12.4.

Vrednotenje

Vrednotenje različnih kriterijev in variant poda zmagovalno oz. najboljšo izbiro (Slika 12.19). Z izbiro selektivne razlage opredelimo vse kriterije za določeno varianto na podlagi določenih vrednosti zalog kriterijev. Z izbiro primerjave variant lahko medsebojno primerjamo dve ali več varianti (Slika 12.20 in 12.21). Orodja pri vrednotenju kriterijev in variant v Tabeli 12.5.

Slika 12.22 prikazuje primer Poročila o primerjavi variant.





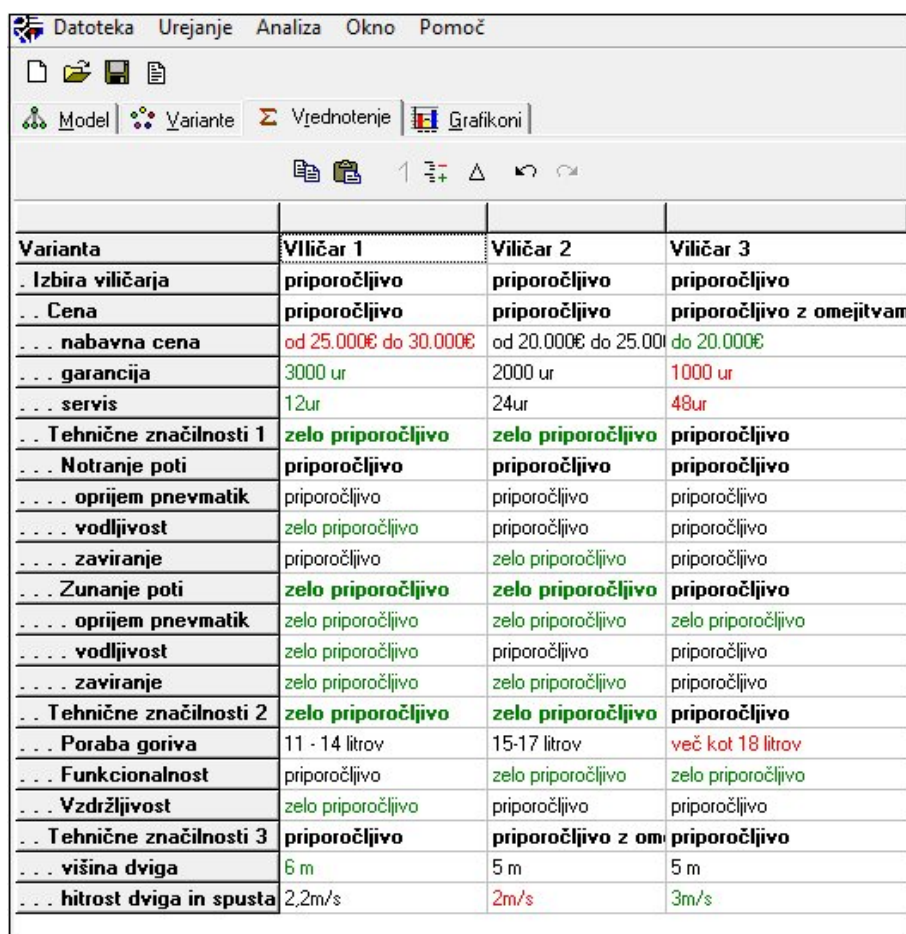
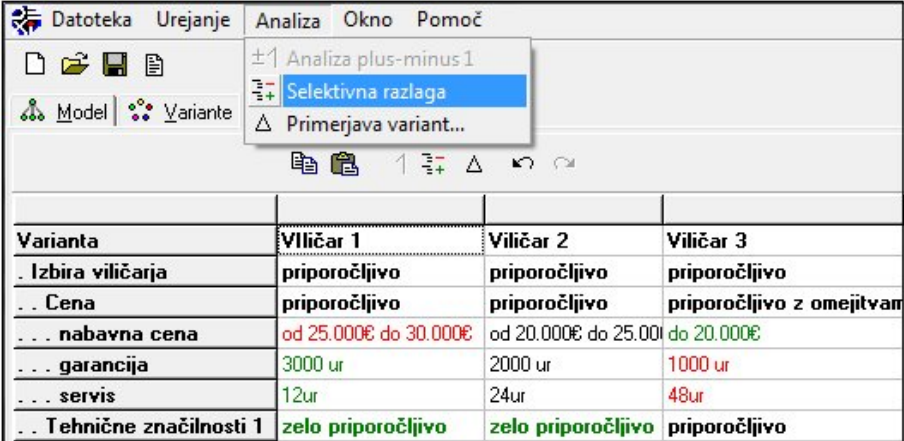
	določitev vrednosti
	selektivna razlaga
	primerjava variant
	kopiraj/prilepi

Tabela 12.5: Orodja pri vrednotenju kriterijev in variant



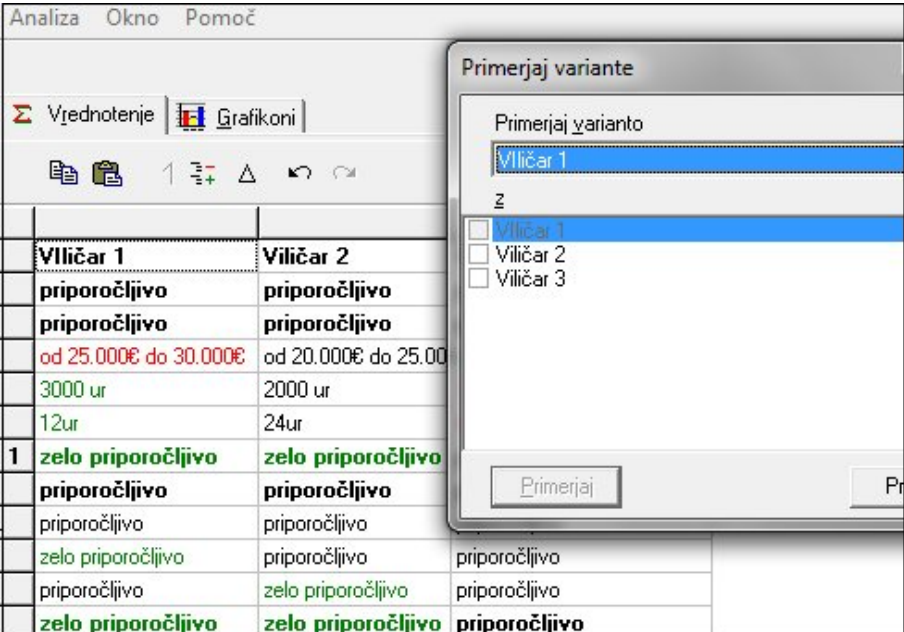
Varianta	Viličar 1	Viličar 2	Viličar 3
. Izbira viličarja	priporočljivo	priporočljivo	priporočljivo
. . . Cena	priporočljivo	priporočljivo	priporočljivo z omejitvam
. . . nabavna cena	od 25.000€ do 30.000€	od 20.000€ do 25.000€	do 20.000€
. . . garancija	3000 ur	2000 ur	1000 ur
. . . servis	12ur	24ur	48ur
. . Tehnične značilnosti 1	zelo priporočljivo	zelo priporočljivo	priporočljivo
. . . Notranje poti	priporočljivo	priporočljivo	priporočljivo
. . . . oprijem pnevmatik	priporočljivo	priporočljivo	priporočljivo
. . . . vodljivost	zelo priporočljivo	priporočljivo	priporočljivo
. . . . zaviranje	priporočljivo	zelo priporočljivo	priporočljivo
. . . Zunanje poti	zelo priporočljivo	zelo priporočljivo	priporočljivo
. . . . oprijem pnevmatik	zelo priporočljivo	zelo priporočljivo	zelo priporočljivo
. . . . vodljivost	zelo priporočljivo	priporočljivo	priporočljivo
. . . . zaviranje	zelo priporočljivo	zelo priporočljivo	priporočljivo
. . Tehnične značilnosti 2	zelo priporočljivo	zelo priporočljivo	priporočljivo
. . . Poraba goriva	11 - 14 litrov	15-17 litrov	več kot 18 litrov
. . . Funkcionalnost	priporočljivo	zelo priporočljivo	zelo priporočljivo
. . . Vzdržljivost	zelo priporočljivo	priporočljivo	priporočljivo
. . Tehnične značilnosti 3	priporočljivo	priporočljivo z om	priporočljivo
. . . višina dviga	6 m	5 m	5 m
. . . hitrost dviga in spusta	2,2m/s	2m/s	3m/s

Slika 12.19: Vrednotenje kriterijev in variant



Varianta	Viličar 1	Viličar 2	Viličar 3
Izbira viličarja	prporočljivo	prporočljivo	prporočljivo
Cena	prporočljivo	prporočljivo	prporočljivo z omejitvam
nabavna cena	od 25.000€ do 30.000€	od 20.000€ do 25.000€	do 20.000€
garancija	3000 ur	2000 ur	1000 ur
servis	12ur	24ur	48ur
Tehnične značilnosti 1	zelo prporočljivo	zelo prporočljivo	prporočljivo

Slika 12.20: Analiza in primerjava variant



Viličar 1	Viličar 2
prporočljivo	prporočljivo
prporočljivo	prporočljivo
od 25.000€ do 30.000€	od 20.000€ do 25.000€
3000 ur	2000 ur
12ur	24ur
1 zelo prporočljivo	zelo prporočljivo
prporočljivo	prporočljivo
zelo prporočljivo	prporočljivo
prporočljivo	zelo prporočljivo
zelo prporočljivo	zelo prporočljivo

Slika 12.21: Primerjava dveh variant

Kriterij	Viličar 2	Viličar 1
Izbira viličarja	priporočljivo	priporočljivo
Cena	od 20.000€ do 25.000€	od 25.000€ do 30.000€
nabavna cena		
garancija	2000 ur	3000 ur
servis	24ur	12ur
Tehnične značilnosti 1	<i>zelo priporočljivo</i>	
Notranje poti	priporočljivo	
oprijem pnevmatik	priporočljivo	
vodljivost	priporočljivo	<i>zelo priporočljivo</i>
zaviranje	<i>zelo priporočljivo</i>	priporočljivo

Slika 12.22: Poročilo o primerjavi variant

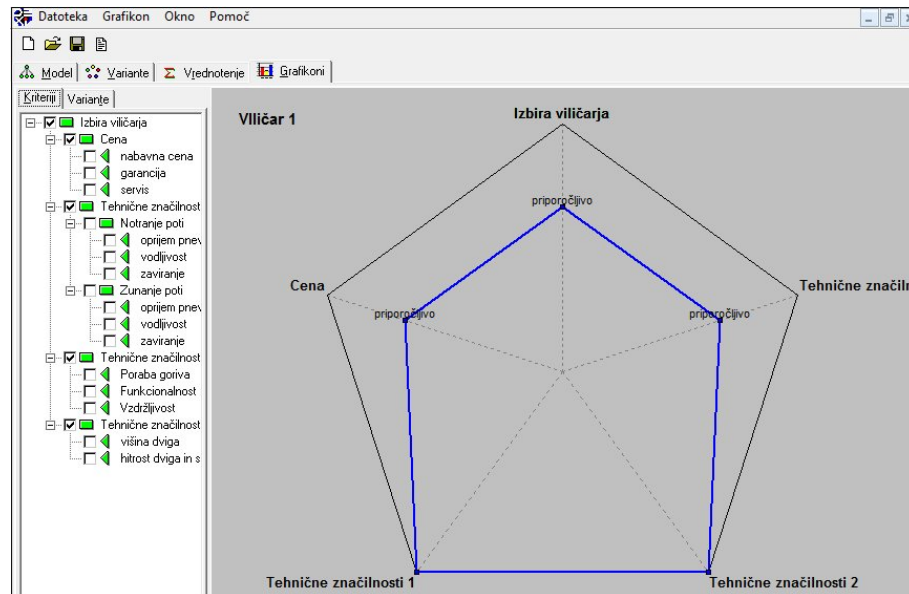
Grafikoni

Programsko orodje DEXi omogoča vizualizacijo vrednotenja atributov oz. kriterijev ter variant. S pomočjo grafičnega prikaza vrednotenja je možno prikazati in izbrati zmagovalca odločanja. V drevesu kriterijev izbiramo in določamo kriterije za primerjavo in vrednotenje (Slika 12.23).

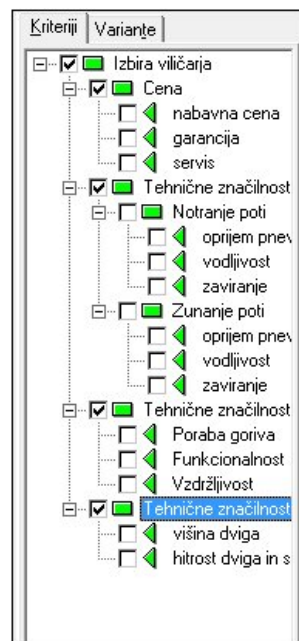
Prikaz reševanja problema

Postopek reševanja problema izbire viličarja:

1. Določiti drevesa kriterijev (primer: cena, tehnične značilnosti itd.).
2. Opredeliti zaloge vrednosti za kriterije (primer: priporočljivo, več kot 10.000 € itd.).
3. Opredeliti odločitvena pravila.
4. Določiti uteži kriterijem (primer: cena ima 45 % utežitev pomembnosti).
5. Opredeliti in določiti variante (možnosti).
6. Opredeliti variante z možnostmi zaloge vrednosti.
7. Vrednotenje kriterijev in variant.
8. Grafični prikaz vrednotenja.
9. Izdelava zaključnega poročila.



Slika 12.23: Grafični prikaz vrednotenja vrednosti parametrov



Slika 12.24: Izbira kriterijev za primerjavo

Drevo kriterijev	
Kriterij	Opis
Izbira viličarja	Izbira viličarja za potrebe interne logistike
Cena	Cena viličarja (ponudba)
nabavna cena	Nabavna cena nakupa viličarja
garancija	Delovanje in uporaba viličarja izražena v delovnih urah
servis	Servisni čas
Tehnične značilnosti 1	Lastnosti viličarjev
Notranje poti	Lastnosti manevriranja v zaprtih prostorih (notranje poti)
oprijem pnevmatik	
vodljivost	
zaviranje	
Zunanje poti	Lastnosti manevriranja viličarja na zunanjih površinah
oprijem pnevmatik	Zaviranje zimskih pnevmatik z ABS sistemom na mokri cesti
vodljivost	Vodljivost zimskih pnevmatik na mokri cesti
zaviranje	
Tehnične značilnosti 2	Lastnosti zimskih pnevmatik (povzeto po AMZS - testiranje)
Poraba goriva	Povprečna poraba uporabe viličarja v obdobju 1h
Funkcionalnost	Funkcionalnost uporabe viličarja (število funkcij)
Vzdržljivost	Pogostost vzdrževanja, redni servisi
Tehnične značilnosti 3	
višina dviga	Višina dviga teleskopa
hitrost dviga in spusta	Hitrost dviga in spusta teleskopa z 0 kg obremenitvijo

Slika 12.25: Drevo kriterijev

Zaloga vrednosti	
Kriterij	Zaloga vrednosti
Izbira viličarja	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
Cena	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
nabavna cena	od 25.000€ do 30.000€ ; od 20.000€ do 25.000€; do 20.000€
garancija	3000 ur ; 2000 ur; 1500 ur; 1000 ur
servis	12ur ; 24ur; 48ur
Tehnične značilnosti 1	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
Notranje poti	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
oprijem pnevmatik	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
vodljivost	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
zaviranje	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
Zunanje poti	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
oprijem pnevmatik	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
vodljivost	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
zaviranje	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
Tehnične značilnosti 2	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
Poraba goriva	do 10 litrov ; 11 - 14 litrov; 15-17 litrov; več kot 18 litrov
Funkcionalnost	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
Vzdržljivost	zelo priporočljivo ; priporočljivo; priporočljivo z omejitvami; ni priporočljivo
Tehnične značilnosti 3	ni priporočljivo ; priporočljivo z omejitvami; priporočljivo; zelo priporočljivo
višina dviga	4 m ; 5 m; 6 m
hitrost dviga in spusta	2m/s ; 2,2m/s; 3m/s

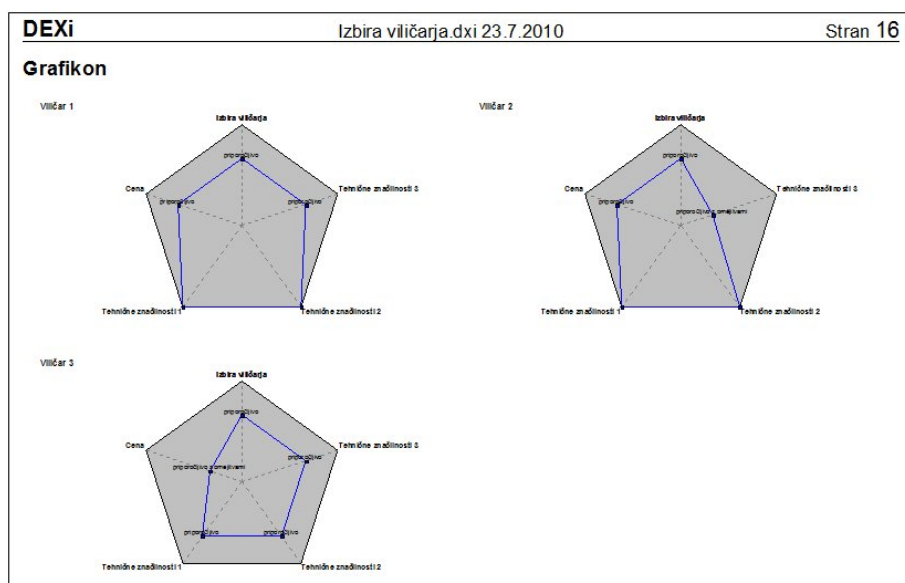
Slika 12.26: Določitev zaloge vrednosti

Tabele odločitvenih pravil		
Cena	Tehnične značilnosti 1	Tehnične značilnosti 2
45%	18%	18%
1 ni priporočljivo	ni priporočljivo	ni priporočljivo
2 ni priporočljivo	ni priporočljivo	<=priporočljivo z omejitvami
3 ni priporočljivo	ni priporočljivo	<=priporočljivo
4 ni priporočljivo	<=priporočljivo z omejitvami	ni priporočljivo
5 ni priporočljivo	<=priporočljivo z omejitvami	<=priporočljivo z omejitvami
6 ni priporočljivo	<=priporočljivo	ni priporočljivo
7 <=priporočljivo z omejitvami	ni priporočljivo	ni priporočljivo
8 ni priporočljivo	<=priporočljivo	*
9 ni priporočljivo	*	<=priporočljivo
10 <=priporočljivo z omejitvami	ni priporočljivo	<=priporočljivo
11 <=priporočljivo z omejitvami	<=priporočljivo z omejitvami	<=priporočljivo z omejitvami
12 <=priporočljivo z omejitvami	<=priporočljivo	ni priporočljivo
13 <=priporočljivo	ni priporočljivo	ni priporočljivo
14 ni priporočljivo	<=priporočljivo	>=priporočljivo z omejitvami
15 ni priporočljivo	*	priporočljivo z omejitvami:pripor
16 ni priporočljivo	*	>=priporočljivo z omejitvami
17 <=priporočljivo z omejitvami	ni priporočljivo	priporočljivo z omejitvami:pripor
18 <=priporočljivo z omejitvami	ni priporočljivo	>=priporočljivo z omejitvami
19 <=priporočljivo z omejitvami	<=priporočljivo z omejitvami	priporočljivo z omejitvami
20 <=priporočljivo z omejitvami	<=priporočljivo z omejitvami	priporočljivo z omejitvami:pripor
21 <=priporočljivo z omejitvami	<=priporočljivo	priporočljivo z omejitvami

Slika 12.27: Tabela odločitvenih pravil

Povprečne uteži				
Kriterij	Lokalne	Globalne	Lok.norm.	Glob.norm.
Izbira viličarja				
Cena	45	45	45	45
nabavna cena	40	18	37	17
garancija	24	11	30	13
servis	35	16	33	15
Tehnične značilnosti 1	18	18	18	18
Notranje poti	35	6	35	6
oprijem pnevmatik	42	3	42	3
vodljivost	42	3	42	3
zaviranje	16	1	16	1
Zunanje poti	65	12	65	12
oprijem pnevmatik	40	5	40	5
vodljivost	40	5	40	5
zaviranje	20	2	20	2
Tehnične značilnosti 2	18	18	18	18
Poraba goriva	35	6	35	6
Funkcionalnost	33	6	33	6
Vzdržljivost	32	6	32	6
Tehnične značilnosti 3	18	18	18	18
višina dviga	50	9	50	9
hitrost dviga in spusta	50	9	50	9

Slika 12.28: Povprečne uteži kriterijev



Slika 12.29: Grafični prikaz vrednotenja parametrov vseh treh variant

Z izdelavo večparametrskega odločitvenega modela ter vrednotenjem kriterijev in različnih variant, ugotovimo, da našim zahtevam najboljše ustreza **Viličar 1**, ki s svojo ceno in tehničnimi značilnostmi prevladuje med ostalimi (variantami). Grafični prikaz (Slika 12.29) jasno dokazuje, da je **Viličar 1**, glede na upoštevane kriterije, vodilni.

Kako?

Do končne ocene pridemo s pomočjo programa DEXi, v katerem natančno opredelimo in strukturiramo kriterije in variante. Izbrani kriteriji so po našem mnenju ključnega pomena, saj pri izboru viličarja igrajo najpomembnejšo vlogo njegove specifikne in cena. Vrednost oz. teža kriterija določimo glede na trenutno pomembnost (finančna kriza in upoštevanje kakovosti izdelka). Vsekakor je od teže kriterija odvisna končna odločitev, zato kriterije postavljamo in opredelimo povsem realno.

Občutljivost odločitve

V "tekmi" med najboljšimi štejejo že najmanjše malenkosti, kar opazimo tudi na danem primeru. Izenačenost variant pri kriteriju tehnične značilnosti je pri končni izbiri v ospredje postavila kriterij cene, ki bistveno vpliva na končno oceno.

Zavedati se je potrebno, da bi vsakršna sprememba utežitev kriterijev lahko bistveno vplivala na končni rezultat. V primeru spremembe cen na trgu, bi se odločitev in končna ocena med variantami še bolj zaostрила in izenačila.

Povzetek

S programskim orodjem DEXi za večparametrski odločitveni model je moč enostavno rešiti vsakdanji problem pri izbiri nakupa novega viličarja. Kateri viličar je najboljša rešitev za naše potrebe in zahteve? Enostavna razčlenitev kriterijev, opredelitev različnih variant in njihovo vrednotenje poda rešitev izbire med njimi najboljšega. S kakovostno raziskavo in primerjavo lahko uspešno nastopimo pred vodstvenim managementu za izbiro in nakup novega viličarja, katerega s pomočjo večparametrskega odločitvenega modela določimo kot najboljšo izbiro. Hitro procesiranje podatkov in odločanje je lahko v kriznih trenutkih odločitvenega pomena. S hitrim in natančnim izborom najboljše variante vsekakor prihranimo del finančnih sredstev.

Poglavje 13

GNUCASH - finančno načrtovanje



Osnovni pojmi računovodstva
Obvladovanje denarnih tokov
Primer: vpliv investicij za nakup sodobnega viličarja

13.1 Teoretično ozadje

Logistika kot dejavnost se ukvarja z upravljanjem različnih tokov (materialni, informacijski in finančni) od virov do porabnikov tako znotraj podjetij kot med podjetji. Glavna naloga je zajem fizičnega toka materiala in toka informacij od dobavitelja, preko proizvajalca in trgovca do končnega potrošnika. Cilj logistike je zagotoviti prave dobrine in storitve, na pravem mestu ob pravem času, v zadostni količini in kakovosti, z najnižjimi stroški in vplivi na okolje, skladno s sklenjeno pogodbo [60].

Celovito obvladovanje oskrbne verige, v našem primeru procesov transporta (dostave) in skladiščenja zahteva učinkovito načrtovanje in kasnejše izvajanje. Za doseganje visokega nivoja učinkovitosti oskrbne verige je med drugim potrebno upoštevati tudi finančni vidik. Finančno načrtovanje po Simonetiju [101] je v širšem pomenu:

- načrtovanje dobička;
- kratkoročno finančno načrtovanje (načrtovanje denarnih tokov);
- dolgoročno finančno načrtovanje (1 leto in več).

Eden izmed vzrok za slabo poslovanje mnogih podjetij je prav slabo finančno načrtovanje. Bistvenega pomena za obstoj in nadaljnjo uspešno poslovanje podjetja je kakovostno finančno načrtovanje, kar pomeni sproti nadzor nad dogajanjem in možnost hitrega ukrepanja ob velikih odstopanjih od načrtovanega. V primeru pozitivnega poslovanja in doseganja odličnih rezultatov podjetja, imajo od tega korist vsi poslovni procesi, med drugim tudi logistika. Dobiček omogoča investicije v nakup sodobne tehnologije in opreme, to omogoča dvig konkurenčnosti podjetja, nivoja storitev, povišanje pretočnosti materialnih tokov, učinkovitosti izvajanja aktivnosti itd. Primer: Investicija v nakup sodobnega viličarja za potrebe skladiščnih manipulacij omogoča učinkovito in uspešno obvladovanje materialnih tokov v fazi skladiščenja, kar se izkaže predvsem na področju obvladovanja stroškov (manj izgub in poškodb blaga, večja pretočnost, krajši čas manipulacij, avtomatiziranost procesov) in humanizaciji dela (lažje delo, uporaba mehanizacije). Za doseganje vseh zastavljenih ciljev, med katere uvrščamo tudi investicijo v nakup sodobnega viličarja, potrebujemo odlično finančno orodje za načrtovanje, izvajanje in kontrolo vseh denarnih tokov. Z obvladovanjem denarnih tokov je prihranek toliko bolj verjeten in kaj hitro lahko ugotovimo ali naše finančno poslovanje prenese investicijo v nakupu novega viličarja. Uporaba enostavnega računovodstva z integracijo računa za kredit za investicijo nakupa novega viličarja omogoča splošen pregled in prikaz finančnega poslovanja v manj zahtevnem okolju.

Na trgu najdemo številna odlična brezplačna oz. odprtokodna finančna orodja za obvladovanje denarnih tokov (računovodstvo), predvsem manjših podjetij in organizacij. Eden izmed njih je tudi odprtokodni in brezplačni GnuCash, ki je namenjen za vodenje osebnih računov in računov manjših podjetij [20].

13.1.1 Finančno načrtovanje

Vsakdo, ki se pri svojem delu srečuje s financami, se srečuje tudi z različnimi zahtevami, problemi in ovirami, ki izhajajo iz narave dela in okolja poslovanja. Razumljivo in pričakovano je, da želimo poiskati tem bolj enostavne rešitve, ki zagotavljajo kakovost poslovanja in storitev.

Dobrodošel pripomoček pri poslovanju malih podjetij in samostojnih podjetnikov je poznavanje osnovnih znanj in veščin računovodstva ter programskih orodij, ki omogočajo kakovostno poslovanje [12]. Vodenje računovodskih evidenc in redno oddajanje zahtevanih poročil je zgolj platforma za odličnost poslovanja.

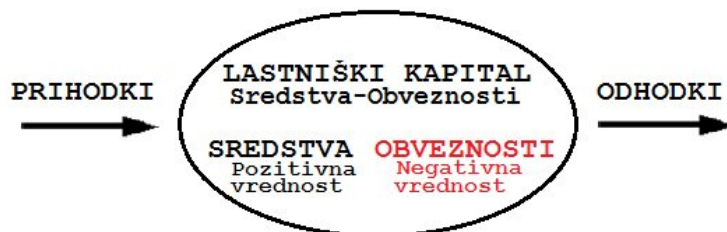
Vodenje računovodskih knjig predpisuje in terja država. Pravna podlaga za vodenje knjigovodstva izhaja iz številnih zakonov: Zakon o gospodarskih družbah, Zakon o davčnem postopku, Zakon o računovodstvu in Slovenski računovodski standardi [103].

Koncepti računovodstva

Za lažje poznavanje in uporabo finančnih programskih orodij je potrebno osnovno znanje računovodstva, ki zajema poznavanje petih osnovnih pojmov [86]:

- *Sredstva* - vse kar imamo v lasti (si lastimo);
- *Obveznosti* - vse kar smo dolžni (dolgujemo);
- *Lastniški kapital* - skupna neto vrednost;
- *Prihodek* - povečuje znesek na računu;
- *Odhodki* - znižujejo znesek na računu.

Vse finančne zadeve preprosto kategoriziramo v pet skupin. Primer: Znesek na bančnem računu so sredstva, hipoteka je obveznost, plača je prihodek, nakup viličarja je odhodek itd.



Slika 13.1: Računovodska enačba

Računovodska enačba

Lastniški kapital je definiran s sredstvi in obveznostmi. Neto vrednost izračunamo tako, da od sredstev odštejemo obveznosti.

$$\text{Sredstva} - \text{Obveznosti} = \text{Lastniški kapital}$$

Lastniški kapital lahko povišamo s prihodki in zmanjšamo z odhodki, kar nazorno prikazuje primer. Ob prejemu plačila postanemo bogatejši in ob poravnavi računa za nakupu viličarja revnejši. To opisuje naslednja enačba:

$$\text{Sredstva} - \text{Obveznosti} = \text{Lastniški kapital} + (\text{Prihodki} - \text{Odhodki})$$

Grafični prikaz razmerij med petimi osnovnimi pojmi prikazuje Slika 13.1. Neto vrednost (lastniški kapital) se poviša s prihodki in zniža z odhodki. Puščica prikazuje smer vrednosti.

Računovodska enačba je temelj dvostavnega računovodstva. Vsaka sprememba pri enem računu vpliva na ravnovesje enačbe, zato mora biti drugi račun kot proti utež za uravnavanje ravnovesja. Takšen koncept je znan kot "princip ravnovesja" in je pomemben za nadaljnjo razumevanje GnuCash in ostalih podobnih finančnih programskih orodij. Pri uporabi GnuCash vedno uporabljamo vsaj dva ali več računov, katere je potrebno uravnovesiti [86].

Dvostopno računovodstvo vključuje zapis vsake transakcije v glavni knjigi, nadalje kopiranje dela transakcije v ločenih knjigah imenovani dnevnik. V dvostopnem računovodstvu je transakcija izmenjava med najmanj dvema računoma. Ta metoda preprečuje napake in omogoča izslediti napako pri vnosu.

Pri vnosu podatkov je potrebno ločiti račun od transakcij. Koncept dvostavnega računovodstva omogoča sledljivost denarja (od kod je prišel in kam gre) [86].

Za izvedbo primera potrebujemo osnovno znanje računovodstva, poznavanje vseh osnovnih petih pojmov in njihove zakonitosti. Prikaz enostavnega vodenja računovodstva zajema manjše število transakcij različnih prihodkov in odhodkov ter vpliv investicije v nakup sodobnega viličarja.

13.2 O programskem orodju

Programsko orodje GnuCash (Slika 13.2) je odprtokodno in brezplačno orodje namenjeno vsem uporabnikom za pomoč na področju računovodstva oz. za vodenje računov [25] [56] in [59]. GnuCash uporablja GPL licenco in je del projekta GNU. Deluje na Linux, OpenBSD, FreeBSD, Solaris, Mac OS X, Unix in Microsoft Windows (z GnuCash verzijo 2.1.x od 14.04.2007 dalje) [59]. Primeren je za vodenje osebnih računov in računov v manjših podjetjih in organizacijah. Uporaben je za dvostavno knjigovodstvo v manjših podjetjih kot podpora pri vodenju prejetih in izdanih faktur [108] in [13]. Določene funkcije omogočajo, da vodimo evidenco strank (dobavitelj, kupec), zaposlenih in različnih računov - izdajanje računov, kot tudi terjatev in obveznosti. S podporo za OFX DirectConnect in HBCI lahko GnuCash komunicira tudi z banko (v kolikor podpira določene standarde). Vse končne rezultate je možno ekstrahirati, tudi poročila in grafe.

Prehod na GnuCash iz druge računovodske programske opreme je enostaven, saj omogoča uvoz dokumentov iz programov kot sta Quicken in Microsoft Money. Dostopen je v 29 tujih jezikih (slovenske različice ni) [13] in [14].

Cilji programske opreme GnuCash so [15]:

- zagotoviti odprtokodno finančno orodje;
- enostavna uporaba;
- razčlenitev aktivnosti-model-pregled-kontrolor;
- finančni motor z mnogimi funkcijami;
- modularnost, prilagodljivost;
- spletni pregledovalnik finančnih podatkov;
- prihodnost: e-denarnica, e-plačevanje računov.

Programsko orodje je namenjeno predvsem za:

The screenshot shows the GnuCash.org website. At the top, there is a navigation bar with the GnuCash logo and the text "Free Accounting Software". Below the logo, there are links for "Information", "Documentation", "Downloads", and "Developer Information". The main content area is titled "Welcome to GnuCash.org" and contains a brief description of the software, a "Feature Highlights" section listing various accounting features, and a "News" section with a prominent announcement for "GnuCash 2.3.14 (Unstable) Release - 2010-06-08". A sidebar on the right offers a "Download GnuCash 2.2.9" button with options for Linux Source, Microsoft Windows, and Mac OS X.

Slika 13.2: Spletna stran GnuCash

- osnovno računovodstvo;
- osebno računovodstvo;
- osebno e-poslovanje;
- e-nakupe in e-plačevanje računov;
- računovodstvo v manjših podjetjih.

Prednosti programskega orodja GnuCash:

- preprosta uporaba;
- interoperabilnost;
- možnost nadgradnje;
- brezplačno programsko orodje;
- široka paleta pomoči.

Slabosti programskega orodja GnuCash:

- omejitve možnosti in dodatnih funkcij, v primerjavi s plačljivimi programskimi orodji;
- okrnjena poročila (grafični prikaz);
- uvoz starih datotek.

Download GnuCash

Stable release (2.2.9)

A stable release is a version of GnuCash that is well tested and considered appropriate for every day use.

The latest stable release of GnuCash is 2.2.9. Choose the download for your operating system below.

Installers

- GnuCash 2.2.9 for [Microsoft Windows XP/Vista/7](#)
- GnuCash 2.2.9 for Mac OS X [Intel](#) | [Power PC](#) | [Readme](#)

Linux

Most linux distributions come bundled with a version of GnuCash, though it's not always the most recent version and it may not have been recommended to use the GnuCash version that comes with your distribution.

Below are ways to install GnuCash on some of the more popular distributions:

- **Fedora:** Fedora users can install GnuCash through System->Administration->Add/Remove Software (Gnome) or Applications
- **Mandriva:** Mandriva users can install GnuCash through the Software Management in the Mandriva Linux Control Center
- **Redhat/Centos:** Redhat and CentOS don't have GnuCash in their repositories by default. It can be installed however from the i configuring this additional software repository for your setup, see [Epeel's wiki page](#)
- **Ubuntu:** Ubuntu 9.10 users can install GnuCash through the Software Center (in the Applications menu). This will install GnuCash Applications menu.

Source code

- [bzip2 compressed tarball](#) for GnuCash 2.2.9
- [USA Mirror](#)
- [European Mirror](#)

Unstable/Development release (2.3.14)

Unstable (development) releases are for testing purposes only. They contain the newest features and improvements, but may also contain releases for everyday use.

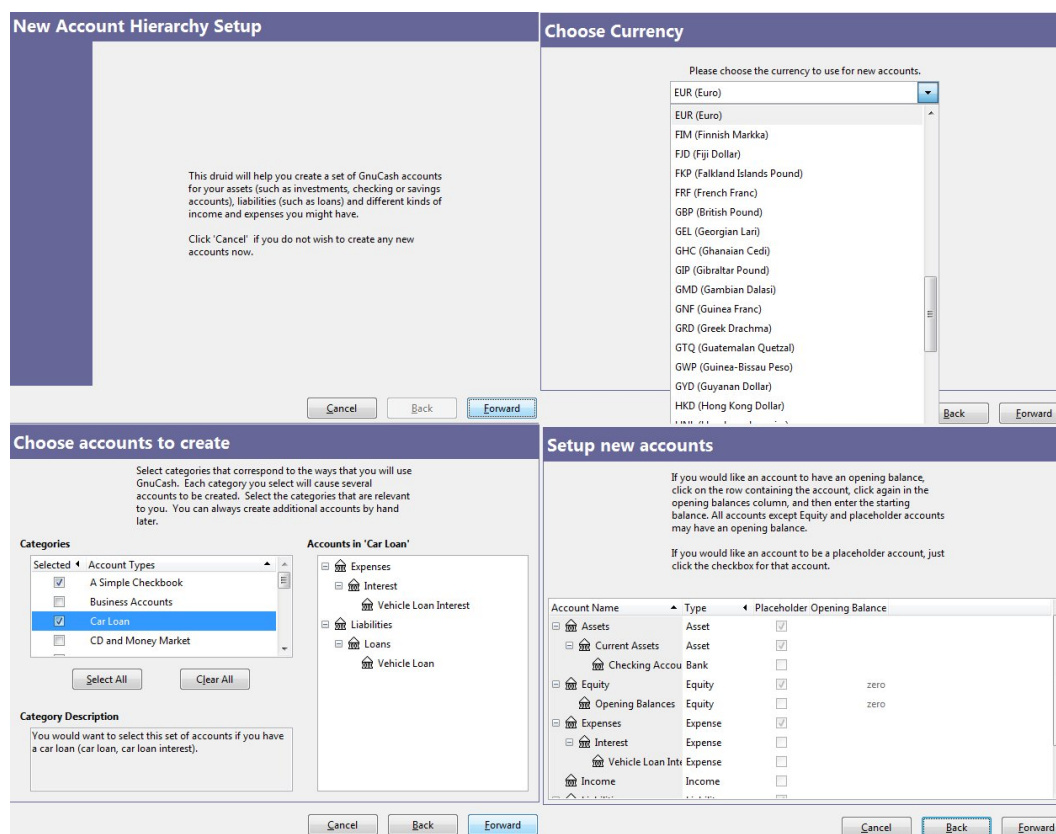
Slika 13.3: Namestitev programskega orodja GnuCash

Prenos in namestitev

Programsko orodje je dostopno na spletnem mestu GnuCash [11], kjer pod razdelkom *Download*, kot prikazuje Slika 13.3, izberemo stabilno verzijo glede na operacijski sistem, ki ga uporabljamo (Microsoft Windows, Ubuntu). Po prenosu izvedemo še namestitev programskega orodja. V kolikor uporabljamo Ubuntu operacijski sistem, je namestitev programskega orodja preprostejša, z uporabo Synaptic orodja enostavno poiščemo, prenesemo in namestimo programsko orodje GnuCash. Navodila za prenos in namestitev najdemo v uvodnem poglavju "Kako dostopati do programskih orodij?"

Nastavitve

Pred pričetkom dela s programom GnuCash s pomočjo čarovnika uredimo nastavitve za nadaljnjo uporabo (Slika 13.4). Izberemo denarno valuto in vrsto računa, ki ga želimo uporabljati. Na izbiro je več kot 30 različnih denarnih valut in 16 različnih računov, katere lahko odvisno od naših potreb in želja poljubno



Slika 13.4: Nastavitve

kombiniramo.

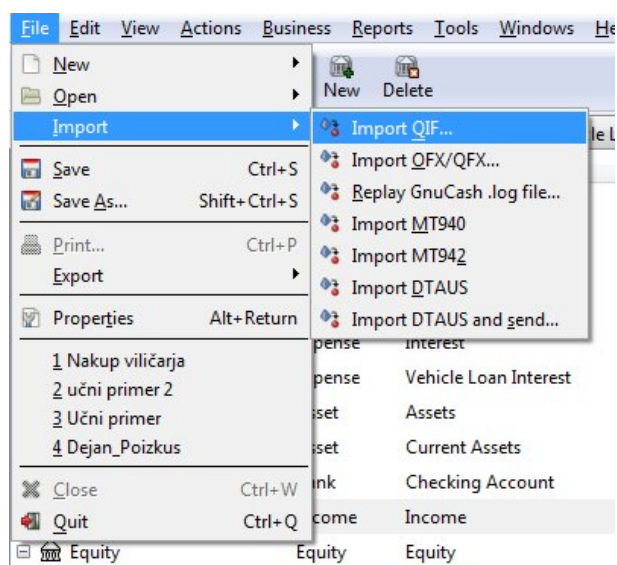
Ko kreiramo in izberemo račune, pričnemo z delom. Prvi korak je vnos podatkov oz. transakcij. Podatke je možno uvoziti ali jih samostojno kreirati pod razdelki računov (Slika 13.5).

Uvažanje podatkovnih datotek

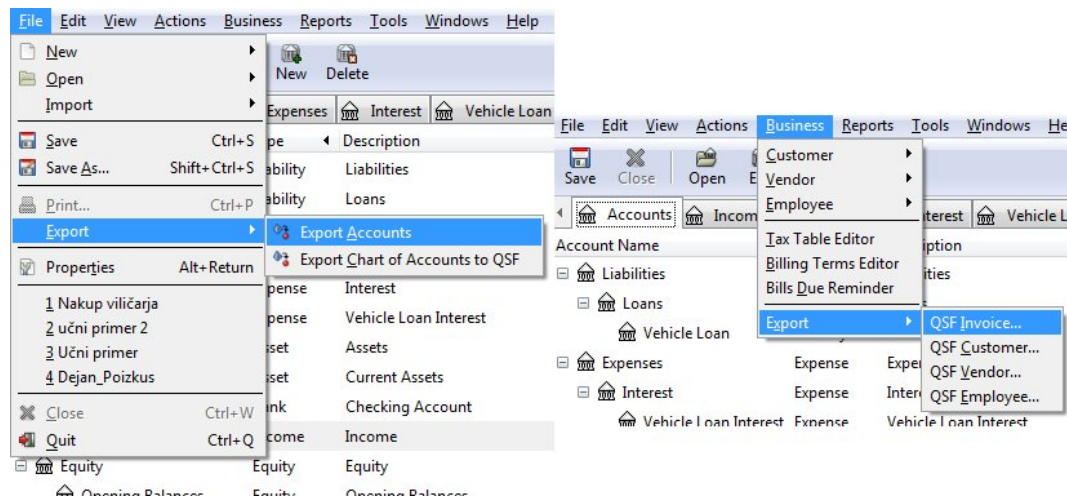
Programsko orodje GnuCash omogoča uvažanje in branje QIF (*Quicken Interchange Format*), kar omogoča večjo integracijo z drugimi plačljivimi programi in kakovostnejše delo v kolikor imamo že ustvarjene datoteke ali jih uvažamo od drugod. QIF je odprta specifikacija za branje in pisanje finančnih dokumentov. GnuCash omogoča uvoz podatkovnih datotek tipa MT942, MT940, DTAUS in OFX/QFX (naslednik QIF datotek). QIF datoteke uvažamo s klikom na *File*, nadalje *Import*, kot prikazuje Slika 13.6.

Date	Num	Description	Transfer	R	Charge	Income	Balance
10.07.2010	5001	Prihodek 1_prodaja vozil		Imbalance-EUR	n	700.000,00	700.000,00
11.07.2010	5002	Prihodek 2_prodaja vozil		Imbalance-EUR	n	70.000,00	770.000,00
11.07.2010	5003	Prihodek 3_prodaja vozil		Imbalance-EUR	n	800.000,00	1.570.000,00
14.07.2010	5004	Prihodek 4_prodaja vozil		Imbalance-EUR	n	140.000,00	1.710.000,00
16.07.2010	5005	Prihodek 5_prodaja vozil		Imbalance-EUR	n	20.000,00	1.730.000,00
17.08.2010		Description	Transfer	n	Charge	Income	Balance

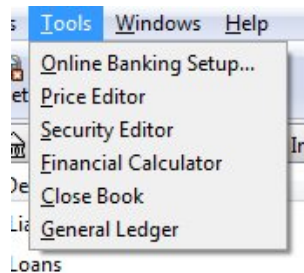
Slika 13.5: Kreiranje in vnos podatkov



Slika 13.6: Uvoz datotek



Slika 13.7: Izvoz podatkovnih datotek



Slika 13.8: Orodja – 1. del

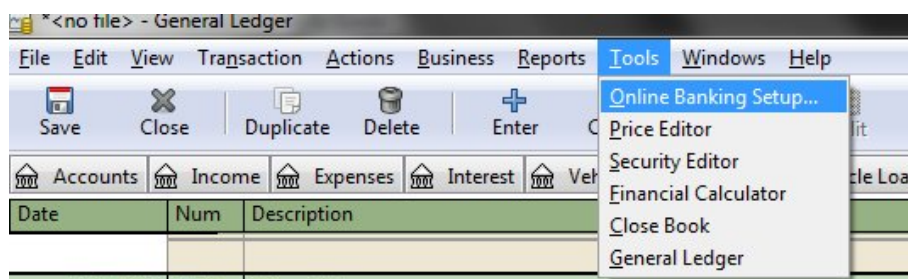
Izvažanje podatkovnih datotek

Programsko orodje GnuCash omogoča izvažanje računov in ostalih podatkovnih dokumentov (Slika 13.7). Podatkovni dokumenti so računi in dokumenti (dobavnica, stranka, zaposleni).

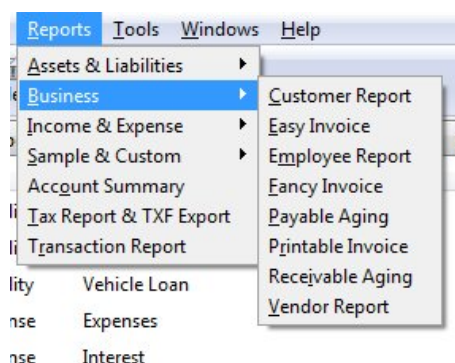
Dodatne funkcije in orodja programa GnuCash

Programsko orodje GnuCash poleg omenjenih funkcij in orodij, katere predstavimo v nadaljevanju, ponuja še različna orodja in funkcije:

1. Orodja v razdelku *Tools* v menijski vrstici: e-poslovanje, e-bančništvo (uvoz/izvoz podatkov podjetje-banka), finančni računalnik (izračun kredita, obresti in števila pologov), glavna knjiga itd. (Slika 13.8 in Slika 13.9).
2. Orodja v razdelku *Reports* v menijski vrstici: sredstva in obveznosti



Slika 13.9: Orodja – 2. del



Slika 13.10: Poročila

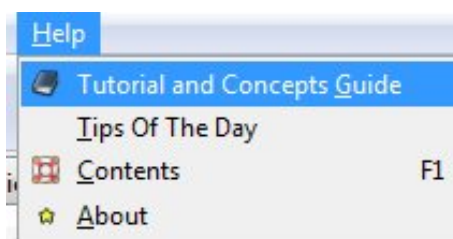
(glavna knjiga, grafi, portofolijo, bilanca stanja); poslovanje (poročilo o zaposlenih, poročilo o stranki, dobavnica); prihodki in odhodki (izkaz poslovnega izida, grafi, denarni tok); davki; transakcije (Slika 13.10).

3. Orodja v razdelku *Business* v menijski vrstici: stranka (kreiranje, urejanje, iskanje podatkov); prodajalec (kreiranje, urejanje, iskanje podatkov); zaposleni (kreiranje, urejanje, iskanje podatkov) in urejevalnik (davki, pogoji).

4. Orodja v razdelku *Help* v menijski vrstici: dodatna pojasnila in pomoč pri kreiranju računov, razložitev določenih finančnih izrazov, transakcij in osnovnih konceptov (Slika 13.11).

Dodatna pomoč za uporabo programskega orodja GnuCash:

- Uporaba GnuCash 1
- Uporaba GnuCash 2



Slika 13.11: Pomoč

Problem

Vpliv investicije v nakup sodobnega viličarja na finančno poslovanje.

Doseganje učinkovitosti in pretočnosti materialnega toka je odvisno od celotnega delovanja sistema oskrbne verige. Časovne zamude v fazi skladiščenja bistveno vplivajo na celotno pretočnost in podaljšanje procesov v oskrbni verigi. Eden izmed vzrokov je lahko zastarela tehnologija oz. uporaba zastarelih transportnih sredstev. Hitro ukrepanje in investicija v nakup sodobne tehnologije (v našem primeru sodobni in ekološko sprejemljiv viličar), ponudita uporabniku možnost optimizacije poslovanja, dosego višje pretočnosti materialnega toka in zmanjšanje stroškov (gorivo, čas in število manipulacij).

Investicija v nakup sodobnega viličarja lahko za manjša podjetja ali organizacije predstavlja nepremostljivo oviro. Tržne cene sodobnih viličarjev se gibljejo od 20.000 € dalje. Takšna investicija predstavlja kratkoročno ogromen odhodek oz. strošek, katerega dolgoročno vsekakor izkoristimo in povrnemo v obliki optimizacije poslovanja in zmanjšanja stroškov. Vsaka investicija v nakup premičnine predstavlja v prvem koraku strošek oz. odhodek, ki ga moramo pokriti z drugimi sredstvi, najpogosteje z bančnim kritje z najemom kredita ali leasinga. V drugem koraku je ta premičnina naš lastniški kapital s katerim je možno najeti dodatne bančne kredite, za katere potrebujemo kritje v obliki nepremičnin ali premičnin. Investicija v nakup sodobnega viličarja vsekakor ni strošek v njegovem celotnem življenjskem ciklu, saj nam po poplačilu obveznosti kredita ta ista premičnina ostane v lasti in nastopa v funkciji lastniškega kapitala.

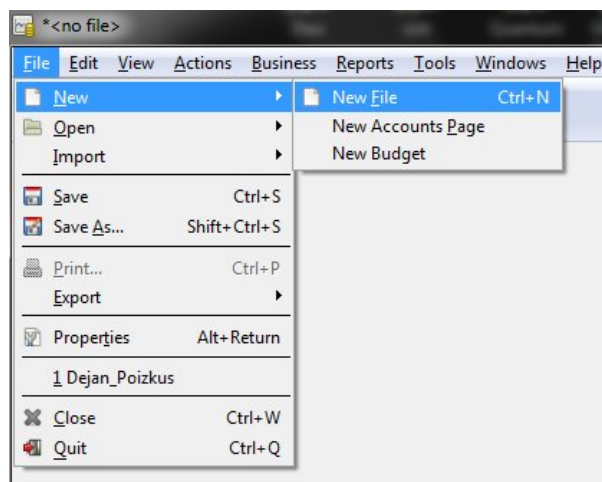
Vpliv investicije v višini 20.000 € na finančno poslovanje podjetja?

13.3 Uporaba

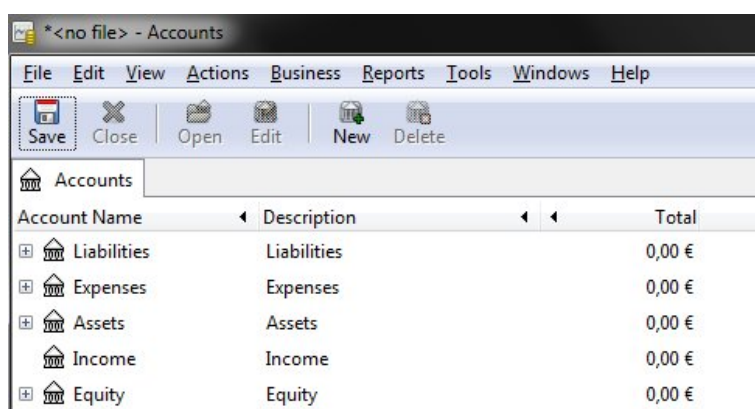
V nadaljevanju predstavimo primer poslovanja podjetja OpenStorage, njihove denarne tokove in transakcije. Poleg tega predstavimo še vpliv investicije v nakup sodobnega viličarja na poslovanje podjetja. Po namestitvi programskega orodja GnuCash in izbiri računov pričnemo z delom. Za pričetek dela je potrebno odpreti nov dokument (Slika 13.12 in Slika 13.13) in izbrati denarno valuto in vrsto računa. V našem primeru izberemo denarno valuto evro (€) in račun za enostavno računovodstvo s kombinacijo posojila za avto. Z enostavnim računovodstvom prikažemo splošne transakcije, kot npr. transakcije prihodkov in odhodkov. Račun *Car Loan* služi kot račun za posojilo za investicijo v nakup sodobnega viličarja. Račun *A Simple Chekbook* uporabimo za prikaz osnovnih transakcij prihodkov, odhodkov, lastniškega kapitala itd.



Slika 13.12: Zagon programa GnuCash



Slika 13.13: Nov dokument



Slika 13.14: Glavna stran

Accounts		Income	Expenses	Interest	Vehicle Loan Interest	Vehicle Loan	Checking Account	Current Assets	Loans	General
Date	Num	Description	Transfer	R	Charge	Income	Balance			
10.07.2010	5001	Prihodek 1_prodaja vozil		Imbalance-EUR	n	700.000,00	700.000,00			
11.07.2010	5002	Prihodek 2_prodaja vozil		Imbalance-EUR	n	70.000,00	770.000,00			
11.07.2010	5003	Prihodek 3_prodaja vozil		Imbalance-EUR	n	800.000,00	1.570.000,00			
14.07.2010	5004	Prihodek 4_prodaja vozil		Imbalance-EUR	n	140.000,00	1.710.000,00			
18.07.2010	5005	Prihodek 5_prodaja vozil		Imbalance-EUR	n	20.000,00	1.730.000,00			
17.08.2010		Description		Transfer	n	Charge	Income	Balance		

Slika 13.15: Prihodki

Po nastavitvah se v GnuCash pojavi glavna stran (Slika 13.14), kjer pričnemo z vnosom podatkov (transakcij, zneskov itd.). Podatke lahko uvozimo ali samostojno kreiramo.

Za vsak račun kreiramo ali uvozimo podatke. Podatke pridobimo z ovrednotenjem vhodnih količin, ki jih pridobimo s pomočjo simulacije dogajanja. Tako so stroški komponent ovrednoteni s ceno in vhodno količino za obdobje 5 dni. Prihodki od prodaje so ovrednoteni s proizvodno količino vozil in dobičkom od prodaje vsakega vozila.

Zaradi lažjega sledenja transakcij je potrebno natančno vnašati podatke (datum) in določiti identifikacijske številke transakcij (številka). Transakcije še opišemo in določimo vrsto transakcij (prenos in transfer računov). Prihodke sestavlja 5 različnih prihodkov od prodaje vozil v skupnem znesku 1.730.000 €, ki jih prejmemo kot nadomestilo za opravljeno dejavnost (transport, dostava, skladiščenje) v procesu oskrbne verige (Slika 13.15).

Odhodke sestavljajo plačila komponent, ki so ovrednotena z nabavno ceno in vhodno količino; mesečni stroški (voda, komunala, električna); izplačilo plač zaposlenim v oddelku skladišča in transporta; ostali stroški (režijski, vzdrževanje). Vsi odhodki znašajo 1.682.500 €. K temu prištejemo še plačilo

Date	Num	Description	Transfer	R	Expense	Rebate	Balance
14.07.2010	400	Plačilo komponent_vijaki		Imbalance-EUR	n	183.000,00	183.000,00
14.07.2010	4001	Plačilo komponent_pnevmatike		Imbalance-EUR	n	924.000,00	1.107.000,00
14.07.2010	4002	Plačilo komponent_platišča		Imbalance-EUR	n	539.000,00	1.646.000,00
14.07.2010	4005	Plačilo stroškov_mesečni		Imbalance-EUR	n	17.000,00	1.663.000,00
15.07.2010	4006	Izplačilo plač		Imbalance-EUR	n	15.000,00	1.678.000,00
16.07.2010	4007	Ostali stroški		Imbalance-EUR	n	4.500,00	1.682.500,00
17.08.2010	Num	Description		Transfer	n	Expense	Balance

Slika 13.16: Odhodki

Date	Num	Description	Transfer	R	Decrease	Increase	Balance	
20.06.2010	1001	Nakup viličarja	Equity:Opening Balances	n	20.000,00		-20.000,00	
18.08.2010	Num	Description		Transfer	n	Decrease	Increase	Balance

Slika 13.17: Najem posojila

obresti za najem posojila, ki znašajo 1.500 €. Skupni znesek odhodkov je tako enak 1.684.000 € (Slika 13.16).

Najem posojila za investicijo v nakup sodobnega viličarja opredelimo in umestimo v obveznosti, saj najem posojila predstavlja obveznost vrnitve sredstev. Znesek posojila znaša 20.000 € in v danem primeru poveča višino lastniškega kapitala (Slika 13.17).

Za sredstva, med katera spada bančni račun, določimosredstva, ki smo jih prejeli in jih hranimo na bančnem računu. Skupni znesek sredstev znaša 29.500 €. Sredstva imajo v lastniškem kapitalu pozitivni predznak, ko od njih odštejemo obveznosti, dobimo vrednost lastniškega kapitala (Slika 13.18).

Obveznosti in sredstva predstavljajo lastniški kapital, kateremu dodamo še razliko med prihodki in odhodki. Razlika med sredstvi in obveznostmi znaša v našem primeru 49.500 € (Slika 13.19).

Pregled vseh računov poda naslednjo sliko (glej Sliko 13.20):

Date	Num	Description	Transfer	R	Deposit	Withdrawal	Balance	
20.07.2010	6001	Državna pomoč	Equity:Opening Balances	n	4.500,00		4.500,00	
20.07.2010	6002	Stanje na računu	Equity:Opening Balances	n	10.000,00		14.500,00	
20.07.2010	6003	Posojilo	Equity:Opening Balances	n	15.000,00		29.500,00	
22.07.2010	Num	Description		Transfer	n	Deposit	Withdrawal	Balance

Slika 13.18: Sredstva

Vehicle Loan		Checking Account		Trial Balance	Account Summary	Hello, World	Opening Balances		Equity	Cash Flow
Date	Num	Description	Transfer	R	Decrease	Increase	Balance			
20.06.2010	1001	Nakup viličarja	Liabilities:Loans:Vehicle Loan	n		20.000,00	20.000,00			
20.07.2010	6001	Državna pomoč	ts:Current Assets:Checking Account	n	4.500,00		24.500,00			
20.07.2010	6002	Stanje na računu	ts:Current Assets:Checking Account	n		10.000,00	34.500,00			
20.07.2010	6003	Posojilo	ts:Current Assets:Checking Account	n		15.000,00	49.500,00			
18.08.2010	Num	Description	Transfer	n	Decrease	Increase	Balance			

Slika 13.19: Lastniški kapital

- višina *obveznosti*, kamor umestimo posojilo za nakup viličarja, znaša 20.000 €,
- višina *odhodka*, kamor umestimo vse odhodke oz. stroške, znaša 1.684.000 €,
- višina *sredstev*, kamor umestimo stanje zneska na bančnem računu, znaša 29.500 €,
- višina *prihodkov*, kamor umestimo vse prihodke, znaša 1.730.000 €,
- višina *lastniškega kapitala*, kjer obveznosti odštejemo od sredstev, znaša 49.500 €.

Account Name	Type	Description	Commodity	Balance	Total
Liabilities	Liability	Liabilities	Euro	0,00 €	-20.000,00 €
Loans	Liability	Loans	Euro	0,00 €	-20.000,00 €
Vehicle Loan	Liability	Vehicle Loan	Euro	-20.000,00 €	-20.000,00 €
Expenses	Expense	Expenses	Euro	0,00 €	1.684.000,00 €
Interest	Expense	Interest	Euro	1.682.500,00 €	1.684.000,00 €
Vehicle Loan Interest	Expense	Vehicle Loan Interest	Euro	1.500,00 €	1.500,00 €
Assets	Asset	Assets	Euro	0,00 €	29.500,00 €
Current Assets	Asset	Current Assets	Euro	0,00 €	29.500,00 €
Checking Account	Bank	Checking Account	Euro	29.500,00 €	29.500,00 €
Income	Income	Income	Euro	1.730.000,00 €	1.730.000,00 €
Equity	Equity	Equity	Euro	0,00 €	49.500,00 €
Opening Balances	Equity	Opening Balances	Euro	49.500,00 €	49.500,00 €
Imbalance-EUR	Bank		Euro	46.000,00 €	46.000,00 €
€ Grand Total:				Assets: 95.500,00 €	Profits: 46.000,00 €

Slika 13.20: Pregled računov

Iz slednjega sklepamo:

- podjetje posluje z dobičkom, ki znaša 46.000 €,
- višina skupnih sredstev, sestavljena iz dobička in lastniškega kapitala, znaša 95.500 €,
- investicija v nakup sodobnega viličarja (najem posojila) poviša lastniški kapital.

S programskim orodjem GnuCash lahko pripravimo in izvozimo različna poročila glede na naše potrebe in želje [24]. S klikom na *Reports* v menijski vrstici izbiramo med različnimi vrstami poročil. Za prikaz denarnih tokov izberemo v menijski vrstici razdelek *Reports*, nadalje *Income and Expense* in nadalje *Cash Flow* (Slika 13.21).

Najpomembnejše je sprotno pregledovanje in vodenje glavne knjige, kamor beležimo vse transakcije z vsemi potrebnimi podatki za določeno obdobje. Poročilo glavne knjige se nahaja v razdelku menijske vrstice *Reports*, nadalje *Assets and Liabilities* in nadalje *General Ledger* (Slika 13.22).

Cash Flow - 01.01.2010 to 31.12.2010

Selected Accounts

- [Assets](#)
- [Assets:Current Assets](#) and selected subaccounts
- [Imbalance-EUR](#)

Money into selected accounts comes from

Equity:Opening Balances	29.500,00 €
Income	1.730.000,00 €
Money In	1.759.500,00 €

Money out of selected accounts goes to

Expenses:Interest	1.682.500,00 €
Expenses:Interest:Vehicle Loan Interest	1.500,00 €
Money Out	1.684.000,00 €

Difference	75.500,00 €
-------------------	--------------------

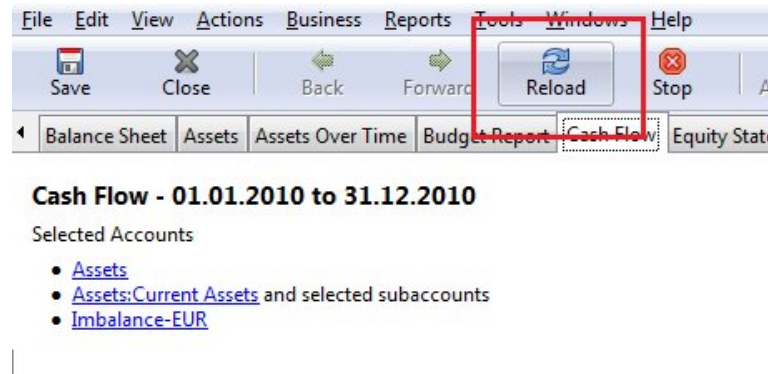
Slika 13.21: Izbira poročila denarnega toka

Transaction Report

From 01.01.2010 To 31.12.2010

Date	Description	Debit		
Opening Balances				
20.06.2010	Nakup viličarja	20.000,00 €		
Total For Opening Balances				20.000,00 €
Vehicle Loan				
20.06.2010	Nakup viličarja	20.000,00 €	20.000,00 €	
Total For Vehicle Loan				-20.000,00 €
Income				
10.07.2010	Prihodek 1_prodaja vozil	700.000,00 €	-700.000,00 €	
Total For Income				700.000,00 €
Imbalance-EUR				
10.07.2010	Prihodek 1_prodaja vozil	700.000,00 €	700.000,00 €	

Slika 13.22: Glavna knjiga



Slika 13.23: Osveževanje podatkov

Osveževanje in posodabljanje podatkov se izvaja samodejno ali s klikom na *Reload* v orodni vrstici programa. Samodejno osveževanje podatkov poskrbi za finančno natančnost pri vsakokratnem dodajanju oz. popravljanju računov (Slika 13.23).

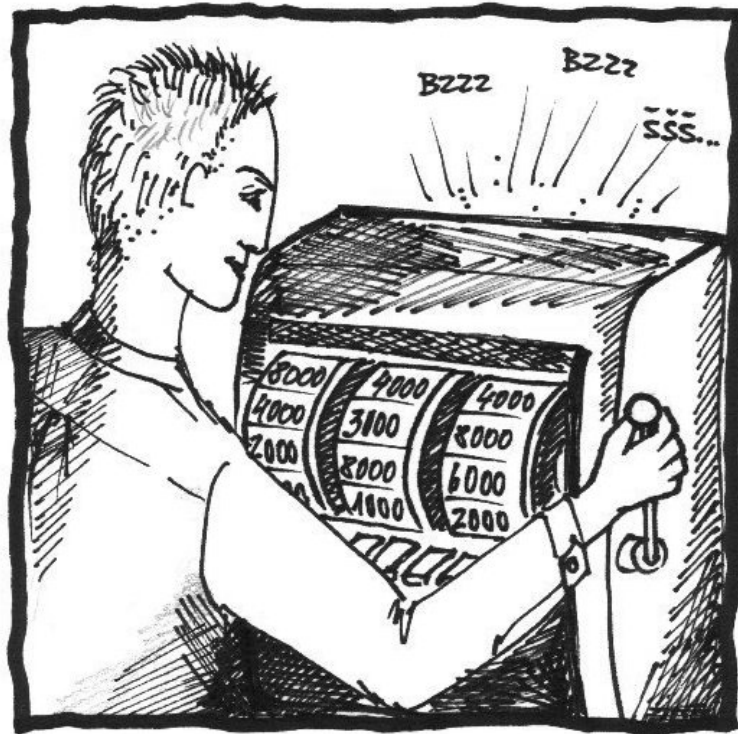
Povzetek

Poslovna učinkovitost, kamor uvrščamo vse logistične procese (transport, skladiščenje...), je v veliki meri odvisna od finančnega načrtovanja, izvajanja in kontroliranja. V številnih primerih je kapital gonilna sila razvoja in napredka in presežki izraženi v kapitalu omogočajo dolgoročni razvoj podjetja ali organizacije. Pravilno in strateško usmerjanje presežkov v razvoj in posodabljanje obstoječih sistemov in opreme zagotavljajo obstoj podjetja in povrnitev vsakega investiranega sredstva. Takšen način zahteva odlično poznavanje finančnih zakonitosti in področij (denarni tokovi, zakoni). Če k osnovnemu znanju računovodstva dodamo še poznavanje in uporabo programskega orodja, govorimo o odlični kombinaciji učinkovitega in kakovostnega vodenja računa. Z uporabo finančnega orodja GnuCash prikažemo manjši segment vodenja osnovnih računov in vpliv investicije v nakup sodobnega viličarja na finančno poslovanje. Vsekakor je investicija v prvem koraku strošek oz. odhodek za podjetje, ampak v nadaljevanju predstavlja vir lastniškega kapitala, kakršna je npr. nepremičnina. Investiranje v sodobne tehnologije ima posreden vpliv na finančno poslovanje, saj dolgoročno zmanjšuje posamezne stroške poslovanja, v našem primeru transporta in manipulacij. Programsko orodje GnuCash je priporočljivo za uporabo in vodenje računov v manjših podjetjih ali manj zahtevnih okoljih, vsekakor pa je priporočljiv za vodenje osebnega računa [50].

Pri opisu programskega orodja GnuCash smo uporabili še dodatne vire in literaturo: [7] [16].

Poglavje 14

GPSS WORLD – simulacija "dogajanja" v skladišču



Simulacijsko orodje
Simulacija vhodnega in izhodnega toka materiala
Primer: gibanje zalog materiala

14.1 Teoretično ozadje

Simuliranje resničnega življenja je že od nekdaj želja mnogih. Življenje kot tako pa je precej kompleksen sistem in kadar za reševanje podobnih kompleksnih sistemov ni dostopnih matematičnih modelov (prezahtevni matematični modeli), se za reševanje tovrstnih problemov najpogosteje uporabljajo računalniške simulacije. Informacijsko podprte simulacije realnih problemov omogočajo predstavitev obnašanja kompleksnega sistema. Njihov primarni cilj je prikaz obnašanja sistema iz resničnega življenja ali hipotetične situacije s spreminjanjem različnih parametrov. Simulacije izvajamo predvsem s podporo sodobne informacijske tehnologije.

Računalniške simulacije uporabljamo predvsem za dinamične, diskretne in stohastične procese [73]. Prav tako tudi za zvezne in hibridne procese [62]. Dinamični procesi so procesi povezani s časovnim potekom, diskretni procesi so procesi razdeljeni na večje število časovnih intervalov, stohastični procesi pa vnašajo negotovost in tveganje v obnašanje sistema. Namen simulacije je generiranje naključnih dogodkov s predhodno konkretizacijo vhodnih podatkov in parametrov, s čimer se približamo dejanskemu stanju iz resničnega sveta. Verodostojnost in točnost izhodnih podatkov zagotovimo z natančno konkretizacijo (približanje realnosti) vhodnih podatkov, na katerih temelji celotna simulacija. Uporaba simulacijskih orodij je v zadnjih letih silovito porasla, predvsem zaradi želje po kvalitetnem reševanju in prikazu problemov [73]. Simulacijska orodja so praktično uporabna v vsakem okolju, to pa zagotavlja napredek na področju informacijske tehnologije.

Področja, kjer uporabljamo simulacije so finance, medicina in druga področja med katera uvrstimo tudi logistiko, promet [73]. Obstaja več vrst simulacij, katere razlikujemo na osnovi:

- kriterijev, ki jih uporabimo za modeliranje določenega vidika realnega sveta (z drugimi besedami teorija in predpostavke, ki jih uporabimo za predstavitev delovnih procesov v podjetju, za predstavitev notranje strukture celice ali nekega naravnega sistema...);
- postopka simulacije, ki ga uporabimo za predstavitev dinamike realnega sveta.

Simulacije delimo glede na *stanje sistema* (diskretne (število programov v sistemu) in zvezne (temperatura)) in *na čas* (diskretne (dnevni tečaji delnic) in zvezne (število zahtev v čakalni vrsti)). V praksi poznamo več vrst simulacij, ki se razlikujejo predvsem na podlagi uporabe v različnih okoljih, kot npr. simulacije letov, simulacije vožnje vozila, simulacije procesov [52].

Pri računalniških simulacijah navadno uporabljajo simulacije na osnovi diskretnih dogodkov in zveznega časa. S simulacijo procesov v izbranem podjetju ponazorimo tok dogodkov in predvidimo morebitne probleme, ki se lahko pojavijo tekom simuliranja izbranega modela. Tako lahko npr. simuliramo tok vhodnega in izhodnega blaga, gibanje zaloge blaga, morebitne čakalne vrste itd.

Eden izmed prvih simulacijskih jezikov, ki omogoča simulacije sistema strežbe je GPSS (General Purpose Simulation Systems) jezik, katerega so razvili v IBM v začetku 60-tih let prejšnjega stoletja [74] [104] [73] [34]. Izgradnja sistema temelji na kopiranju specifičnih statističnih podatkov (intenziteta prometa, povprečno število čakajočih v vrsti) brez grafične ponazoritve simulacije.

Kakovost in odličnost simulacije je odvisna od simulacijskega orodja, pri čemer odločilno vlogo igrajo finance. Različna plačljiva simulacijska orodja ponujajo najsodobnejše možnosti priprave in izdelave simulacij. Za osnovne potrebe simuliranja dogodkov enostavnih primerov so uporabni že preprosti in prosto dostopni programi, kot so GPSS World, WebGPSS, SimPy itd. V nadaljevanju bomo predstavili uporabo simulacijskega orodja GPSS World.

Računalniške simulacije opisujejo operacije simulacij. Mnogo računalniških simulacij ima grafično podlago in preprosto statistično analizo in obdelavo podatkov. Pomemben del računalniški simulacij diskretnih dogodkov je generiranje naključnih kombinacij in števil, ki izhajajo iz verjetnostne porazdelitve.

Primeri programske opreme za *diskretne dogodke* so AutoMod, Arena, GPSS, Plant Simulation in Simula. Primeri programske opreme za *zvezne dogodke* so DYNAMO, SimApp, Simgua in VisSim. Primeri programske opreme za "mešane" so AnyLogic, Modelica, Saber, SeSAM Multiagent simulator and graphical modelling environment in Simulink [62].

14.2 O programskem orodju

Simulacijsko orodje GPSS World je bilo sprva namenjeno predvsem za zagotavljanje kakovostnih odgovorov na vprašanje "Kaj če?«. GPSS World ohranja transparentnost in vodljivost sistema. Zasnovan je na način zagotavljanja hitrih in zanesljivih odgovorov. V skladu s temi cilji je stiliziran in priporočljiv za uporabo, vendar samo simulacijsko orodje ne zagotavlja grafične ponazoritve procesa simulacije [73] [34].

Z uporabo GPSS World simulacijskega orodja je mogoče predvideti učinke modeliranja in izbire realnega kompleksnega sistema. Gre za celovito orodje za modeliranje, ki obsega diskretne in zvezne funkcije računalniške simulacije, z visoko stopnjo interaktivnosti in vizualizacije. Uporabniki lahko pričakujejo



Slika 14.1: Spletna stran GPSS World

hitre odgovore, brez nepotrebne predhodne priprave vizualizacije problema. Idealno je za tiste, ki potrebujejo hitre rešitve v kratkem času (Slika 14.1) [34].

Prenos in namestitvev

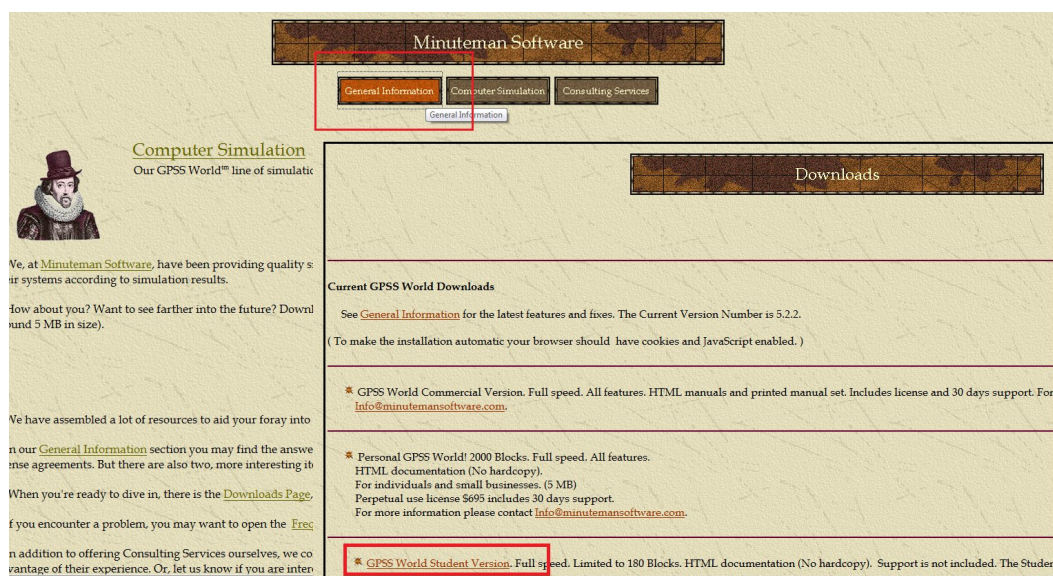
Simulacijsko orodje GPSS World je izdelano za Windows okolje, vendar programsko orodje Wine omogoča uporabo tudi v Ubuntu. Dostopen na spletni strani GPSS, kjer izbiramo med različnimi licenciranimi verzijami. Študentska verzija je brezplačna in prosto dostopna, od plačljivih verzij se razlikuje le v številu možnih blokov in hitrosti procesiranja. Izbrano študentsko verzijo namestimo s klikom na *General Information* in nadalje v razdelku *Downloads*, kjer s klikom na *GPSS World Student Version* pričnemo s prenosom programskega orodja (Slika 14.2).

Osnovno znanje za uporabo simulacijskega orodja GPSS World

Za uporabo simulacijskega orodja GPSS World priporočamo znanje iz področij: operacijskih raziskav, optimizacijskih metod, splošnih matematičnih metod, osnov programiranja.

Osnovne značilnosti simulacijskega orodja GPSS World

Simulacijsko orodje GPSS World je odlično za uporabo simuliranja izbranega modela, natančneje predvideva odziv kompleksnih sistemov. Orodje temelji na GPSS simulacijskem jeziku. Omogoča enostavne simulacije vseh vrst strežnih sistemov in mrež. GPSS World ima vgrajen program za izpis določenih statističnih podatkov (izkoriščenost strežnika, intenzivnost prometa, povprečno



Slika 14.2: Prenos programskega orodja GPSS World

število čakajočih v vrsti, povprečni strežni čas, povprečni čas čakanja v vrsti itd.) [110].

Diskretne dogodke, ki jih simuliramo v GPSS World, predstavimo z bločnim diagramom, v katerem bloki predstavljajo neke aktivnosti. V modelih srečujemo različne elemente, statične in dinamične [73]:

- dinamični elementi so: prečkanje vozil skozi križišče in stranke, ki prihajajo k frizerju in
- statični elementi: križišče, ki ga vozila prevozijo, pristajalne steze, frizer.

Dinamični elementi se imenujejo transakcije. Statični elementi so predstavljeni z bloki (izvor transakcij, domena transakcij, prostor za čakanje transakcij). Same transakcije z bločnim diagramom ne moremo prikazati, zato simulacijski projekt zahteva nekaj korakov [110]. Poleg modeliranja in zbiranja podatkov še testiranje in verificiranje, simulacije, analize in rezultiranje. GPSS World ima zmogljivost prikazati vsakega od teh korakov, pri čemer je simulacija sestavljena iz blokov in transakcij. Bloki so postaje, preko katerih potujejo transakcije. Transakcije pa predstavljajo elementarne delce, ki se gibljejo od postaje (bloka) do postaje (bloka) po v naprej določenem vrstnem redu.

Prednosti pred drugimi konkurenti (simulatorji) (povzeto po [110]):

- lahko se ga je naučiti (ne potrebujemo predznanja o programiranju);

- ni nepotrebne dela z definiranjem izpisa in
- program GPSS je najkrajši v primerjavi z drugimi jeziki.

Ima tudi nekaj slabosti:

- simulacijski čas je daljši kot pri drugih simulatorjih in
- če je problem težko rešiti v GPSS, je potrebno uporabiti drug jezik.

Transakcije	Izvršijo določeno nalogo v modelu; ko končajo svojo nalogo, zapustijo sistem in se uničijo.
Bloki	So namenjeni za opis premikov transakcij skozi sistem. Predstavlja specifično akcijo oz. dogodek, ki se pojavi v sistemu. Bloki, ki se najpogosteje uporabljajo: Generate, Terminate, Advance, Release, Size, Enter, Leave, Test, Transfer.
Resursi	So statične entitete, ki ne morejo zapustiti sistema in jih ne moremo uničiti. Uporabljajo jih transakcije. GPSS ima dva tipa resursov: Facility in Storage.
Krmilni stavki	Definirajo entitete, se uporabljajo v modelu ter upravljajo in vodijo izvršitev blokovnih stavkov v modelu.
SNA	Standardni numerični atributi so funkcije, ki zagotavljajo informacije o entitetah in izvršujejo določen tip kalkulacije.
Blokovni diagram	Diagram, v katerem so glavni deli ali funkcije prikazani s povezavami med bloki, kar predstavlja relacije in odnose med bloki.

Tabela 14.1: Osnovni elementi za simulacijo v GPSS

Vir: prirejeno po [35]

Problem 1

V izbranem primeru želimo simulirati materialni tok treh različnih komponent: pnevmatik, platišč in vijakov v oskrbni verigi, ki povezuje različne subjekte (dobavitelj, prevoznik, kupec itd.) in aktivnosti (dostava, prevzem, skladiščenje, odprema itd.). Sistematično planiranje oskrbne verige v fazi izvedbe dostave in skladiščenja potrebnih komponent zagotavlja učinkovitost in visoko stopnjo odzivnosti vseh subjektov v oskrbni verigi, kar rezultira v večjo pretočnost materialnega toka. Pretočnost materialnega toka je pomembna z dveh vidikov, ekonomskega in tehničnega. Ob predpostavljaju, da je pretočnost materiala nizka, pomeni, da imamo visoke stroške zaradi zamud, izgub itd. Upoštevanje načela odprave nepotrebnih stroškov zahteva investicijo v reorganizacijo in optimizacijo dela. Pri tem je pomemben tehnični vidik, torej kako in s čim doseči višjo pretočnost materialnega toka. Ena izmed možnosti je uporaba različnih simulacijskih orodij za izdelavo različnih scenarijev in modelov. Simuliranje materialnega toka treh različnih komponent, na vhodni strani modela. Simuliranje dostave, ki se izvrši s tremi različnimi prevoznimi sredstvi, in sicer platišča s tovornjakom, pnevmatike z vlakom in vijaki s kombijem in na izhodni strani odpreme komponent z vmesno fazo skladiščenja. Izvedemo jo s pomočjo simulacijskega orodja GPSS World in hkrati preverimo oz. izdelamo različne scenarije s spreminjanjem vhodnih podatkov (časovni interval dostave, kapacitete skladišča, količina dostavljenih komponent, manipulacijski čas itd.) z uporabo "Kaj če?" analize.

S simulacijo materialnega toka v časovnem intervalu 7200 minut oz. 5 delovnih dni (24 urni delovnik) skušamo identificirati:

- izkoriščenost skladiščnih kapacitet;
- čakalne vrste na vhodni in izhodni strani skladiščnega procesa;
- interna pretočnost materiala (skladiščenje, prevzem in odprema);
- gibanje zalog;
- optimalne naročilne količine.

S spreminjanjem vhodnih podatkov poizkusimo dobiti najoptimalnejši scenarij, to pomeni najkrajši časi (čakalnih vrst, odpreme, manipulacij), izkoriščenost skladiščnih kapacitet, enakomernost dostavljenih količin treh komponent itd. Časovne zamude pri dostavi, manipulaciji ali prezasedenosti skladiščnih kapacitet ene komponente, bodisi vijakov ali pnevmatike, povzročijo podaljšanje časa prevzema, čakalne vrste, zastoje pri odpremi itd.

Reševanje in določitev najoptimalnejšega scenarija je težaven in dolgotrajen proces, pri čemer upoštevamo številne dejavnike, ki vplivajo na izvajanje procesov in podprocesov. Z uporabo programskih orodij je to enostavno, a prihaja do razlik pri rezultatih, ki jih dobimo z analitičnim reševanjem in reševanjem s pomočjo programskih simulacijskih orodij.

Analitičen izračun časovnih intervalov prihodov dostave komponent

Izhajamo iz dejstva, da želimo v izbranem proizvodnem obdobju, ki znaša 7200 minut ali 5 dni proizvesti 770 vozil, pri čemer 1 vozilo potrebuje 4 platišča, 4 pnevmatike in 16 vijakov. Pri tem potrebujemo naslednjo količino komponent, glej Tabelo 14.2.

Naročene količine komponent je potrebno še transportirati od dobavitelja do našega skladišča. Transport izvajamo s tremi različnimi prevoznimi sredstvi. Vsako izmed njih ima določeno zmogljivost oz. kapaciteto, ki jo lahko prepelje. Kapaciteta prevoznih sredstev je predstavljena v Tabeli 14.3.

S podanimi podatki izračunamo potrebno število in časovni interval dostav v določenem obdobju. Izračun v Tabeli 14.4.

Z izračunom potrebnih intervalov dostave komponent lahko sklepamo pogodbe z našimi dobavitelji, pri čemer upoštevamo, da dobavitelj in njegov prevoznik upoštevata dogovorjene časovne roke dostave komponent, da se izognemo tveganjem zastoja proizvodnega procesa ali kopičenja zalog. Pri določitvi intervalov dostave moramo vsekakor upoštevati načelo fleksibilnosti in prožnosti dostave in dodati k dogovorjenim časovnim rokam določen odstotek standardnega odklona (+/- minute). Manjši je odstotek standardnega odklona, manjša je prožnost in fleksibilnost dostave, pri čemer obstaja večje tveganje in verjetnost, da pride do zamud pri dostavi. Nasprotno je ob večjem odstotku standardnega odklona. Kakšen naj bo ta odstotek in kako bo vplival na celoten proces oskrbne verige, je naslednji zahteven in dolgotrajen korak analitičnega reševanja. Z uporabo programskega orodja GPSS World je ta naloga enostavnejša in hitreje rešljiva.

Platišča	770 vozil x 4 kos x 5 dni = 15.400 kos
Pnevmatike	770 vozil x 4 kos x 5 dni = 15.400 kos
Vijaki	770 vozil x 16 kos x 5 dni = 61.600 kos

Tabela 14.2: Analitičen izračun časovnih intervalov

Tovornjak (platišča)	390 kos
Vlak (pnevmatike)	800 kos
Kombi (vijaki)	1.450 kos

Tabela 14.3: Kapacitete prevoznega sredstva

	Število dostav (5 dni)	Število dostav (1 dan)	Interval dostave (ura)	Interval dostave (minute)
Platišča	39,48	7,896	3,03	181,8
Pnevmatika	19,25	3,85	6,23	373,8
Vijaki	42,48	8,496	2,82	169,2

Tabela 14.4: Kapacitete prevoznega sredstva

Izračuni

Število dostav

Obdobje 5 dni

15.400 kos / 390 kos = 39,48

15.400 kos / 800 kos = 19,25

61.600 kos / 1.450 kos = 42,48

Obdobje 1 dan

39,48 / 5 = 7,896

19,25 / 5 = 3,85

42,48 / 5 = 8,496

Interval dostav

Ure

24 ur / 7,896 = 3,03 ure

24 ur / 3,85 = 6,23 ure

24 ur / 8,496 = 2,82 ure

Minute

3,03 * 60 = 181,8

6,23 * 60 = 373,8

2,82 * 60 = 169,2

14.3 Uporaba

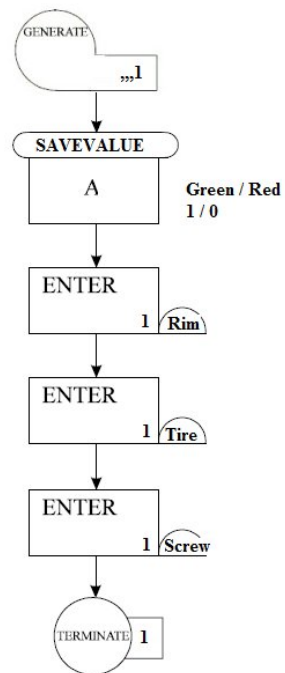
Simulacije različnih scenarijev izvedemo v simulacijskem orodju GPSS World, kjer predhodno pripravimo potreben simulacijski model oz. program (dodan v Prilogi) z vnosom izbranih podatkov in njihovo modifikacijo. Cilj simulacije je doseči vse prej omenjene predpostavke, čim krajši časi, čim večja izkoriščenost skladiščnih kapacitet itd.

Blokovni diagram

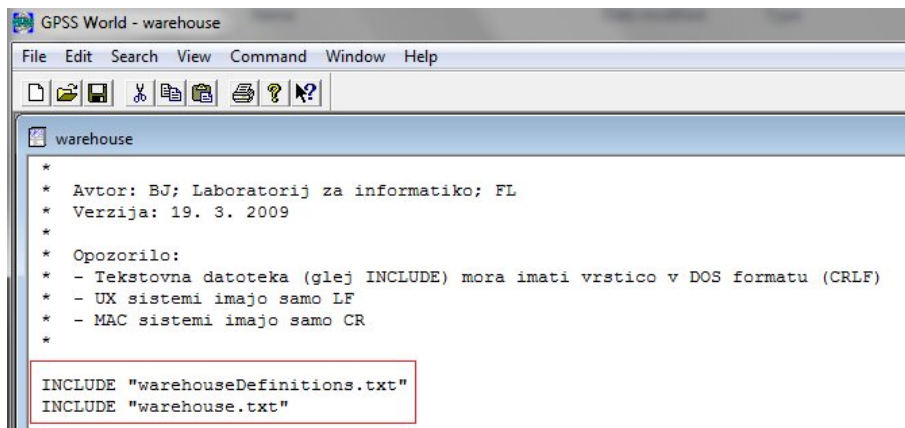
Začetek simulacije temelji na pripravi blokovnega diagrama, s katerim ponazorimo potek procesov ter njihovih medsebojnih povezav in odnosov. Blokovni diagram je hkrati tudi arhitektura simulacijskega modela. Uporaba podobnega simulacijskega orodja WebGPSS temelji na izdelavi blokovnega diagrama (Slika 14.3) [55].

Tekstovna datoteka

Simulacijski model in vhodne podatke predhodno pripravimo v obliki tekstovne datoteke v beležnici (*Notepad*) in jih kasneje kličemo v GPSS World. Tekstovna datoteka s simulacijskim modelom mora vsebovati s smiselno urejene bloke in transakcije (bistveni element GPSS so transakcije, ki v modelu izvršijo določeno nalogo in lahko imajo enega ali več atributov), tekstovna datoteka z vhodnimi



Slika 14.3: Primer blokovnega diagrama za izbrano simulacijo



```
GPSS World - warehouse
File Edit Search View Command Window Help
warehouse
*
* Avtor: BJ; Laboratorij za informatiko; FL
* Verzija: 19. 3. 2009
*
* Opozorilo:
* - Tekstovna datoteka (glej INCLUDE) mora imati vrstico v DOS formatu (CRLF)
* - UX sistemi imajo samo LF
* - MAC sistemi imajo samo CR
*
INCLUDE "warehouseDefinitions.txt"
INCLUDE "warehouse.txt"
```

Slika 14.4: Klicanje tekstovnih datotek v GPSS

podatki pa vhodne podatke kot so skladiščna kapaciteta, časovni intervali. GPSS kliče vse izbrane tekstovne datoteke in izvede simulacijo na podlagi podanega simulacijskega modela in podatkov (Slika 14.4). GPSS omogoča izpis izhodnih podatkov v obliki poročila, pridobljene podatke pa lahko poljubno urejamo in uporabljamo za nadaljnje raziskave, analize in grafične ponazoritve.

Izdelava simulacijskega modela

Simulacijski model je sestavljen iz 14 osnovnih blokov in stavkov, ki skupaj tvorijo kombinacijo celotne simulacije iz 56 blokov. Simulacija je sestavljena iz simulacijskega modela in vhodnih podatkov. Oba dela simulacije sta kreirana v tekstovni datoteki in sta "klicana" v začetni model, kreiran v programskem orodju GPSS World. Za simulacijski model uporabimo bloke in stavke, ki so obrazloženi v Tabeli 14.5 (prirejeno po [35]).

GENERATE	Blok GENERATE kreira transakcijo in jih vnaša v model.
SAVEVALUE	Blok SAVEVALUE shranjuje vrednosti.
ENTER	Blok ENTER simulira vstop transakcije v "storage".
TERMINATE	Blok TERMINATE uniči transakcije, ko zapustijo model.
QUEUE	Blok QUEUE omogoča vstop transakcije v vrsto za statistiko.
TEST	Blok TEST dovoljuje in prepoveduje nadaljevanje poti transakcije preko bloka.
ADVANCE	Blok ADVANCE pomeni čas zadrževanja transakcije, uporabljamo ga za simulacijo zakasnitve transakcije.
DEPART	Blok DEPART pomeni, da je transakcija zapustila vrsto.
TABULATE	S TABLE izvršimo izračun tabele z vstopom transakcije v blok TABULATE.
TABLE	S stavkom TABLE definiramo tabelo.
LEAVE	Blok LEAVE simulira izstop transakcije iz "storage".
START	START je krmilni stavek, ki ga uporabimo za inicializacijo simulacije.
STORAGE	STORAGE je definicijski blok, kateremu določimo kapaciteto.
EQU	Številčno izražena vrednost.

Tabela 14.5: Bloki in stavki

SIMULACIJSKI MODEL (del)

=====

```

GENERATE      ,,,1
SAVEVALUE    TrafficLight,Green
ENTER        StoRim,StartNoRim
ENTER        StoTire,StartNoTire
ENTER        StoScrew,StartNoScrew

```

```

TERMINATE

```

```

*** Rim section begin *****

```

```

*
```

```

* Input to warehouse :: rims

```

```

*
```

```

GENERATE      InRimTimeMean,InRimTimeRange
QUEUE        QueueInWarehouse,1
TEST GE      R$StoRim,InCapVehRim
TEST E       X$TrafficLight,Green
SAVEVALUE    TrafficLight,Red
ENTER        StoRim,InCapVehRim
ADVANCE      InVehManipulRim,InVehManipulRimRange
SAVEVALUE    TrafficLight,Green
DEPART       QueueInWarehouse,1

```

```

TERMINATE

```

```

*
```

```

* Output from warehouse :: rims

```

```

*
```

```

GENERATE      OutRimTimeMean,OutRimTimeRange
TABULATE     TableRim
TEST GE      S$StoRim,OutCapVehRim,RimStorageEmpty
LEAVE        StoRim,OutCapVehRim

```

```

TERMINATE

```

```

RimStorageEmpty TERMINATE

```

```

*
```

```

*
```

```

*** Rim section end *****

```

```

*** Tire section begin *****

```

```

*
```

```

* Input to warehouse :: tire

```

```

*
```

```

GENERATE      InTireTimeMean,InTireTimeRange
QUEUE        QueueInWarehouse,1
TEST GE      R$StoTire,InCapVehTire

```

```

TEST E           X$TrafficLight,Green
SAVEVALUE       TrafficLight,Red
ENTER           StoTire,InCapVehTire
ADVANCE         InVehManipulTire,InVehManipulTireRange
SAVEVALUE       TrafficLight,Green
DEPART          QueueInWarehouse,1

TERMINATE

*
*   Output from warehouse :: tire
*
GENERATE         OutTireTimeMean,OutTireTimeRange
TABULATE        TableTire
TEST GE         S$StoTire,OutCapVehTire,TireStorageEmpty
LEAVE           StoTire,OutCapVehTire

TERMINATE
TireStorageEmpty TERMINATE
*
*
*** Tire section end *****

```

Obrazložitev modela

Problem distribucije in skladiščenja treh različnih komponent oblikujemo v simulacijski model. Simulacijski model generira transakcije v obdobju 7200 minut, le te potujejo po vnaprej izbrani poti, kar pomeni, da obiščejo vse bloke, ki jih uporabimo v našem simulacijskem modelu.

Simulacija se prične s kreiranjem začetnega stanja zalog v skladiščih in proste čakalne vrste. Sledi generiranje transakcij prihodov dostavnih prevoznih sredstev v določenih časovnih intervalih. V danem primeru je transakcija prihod dostavnega vozila, ki se postavi v čakalno vrsto za razkladanje komponent. Pri tem preverimo čakalno vrsto, ali je kakšno prevozno sredstvo pred njim in stanje zaloge v skladišču, saj mora v primeru zasedenosti skladiščnih kapacitet prevozno sredstvo počakati na zmanjšanje skladiščnih enot (odprema) in posledično na razkladanje. Po razkladanju sledi manipulacija in skladiščenje komponent. Odpremni časi in odpreme količine so določeni s časovnimi intervali. Viličar ob prihodu v skladišče preveri razpoložljivost skladiščnih enot. V kolikor ni zadosti skladiščnih enot, čaka na dostavo novih enot. Simulacijski model je enak za vse tri komponente, katere medsebojno vplivajo na čakalno vrsto pri dostavi komponent.

Porazdelitev gibanja zalog zapisujemo v tabelo, s katero lahko ugotovimo pogostost praznega ali polnega skladišča. K tem podatkom lahko dodamo še finančni vidik in finančno izkoriščenost skladišča.

OBRAZLOŽITEV SIMULACIJSKEGA MODELA

=====

Začetno stanje v skladišču

GENERATE - generiramo transakcije začetnega stanja v skladišču, limit je 1
 SAVEVALUE - shranimo vrednosti, če je »Green« je 1, »Red« je 0
 ENTER - začetno stanje v skladišču platišč (»storage«)
 ENTER - začetno stanje v skladišču pnevmatik (»storage«)
 ENTER - začetno stanje v skladišču vijakov (»storage«)
 TERMINATE - transakcija zapusti model

Skladišče platišč

Prihod

GENERATE - generiramo transakcije prihodov v skladišče platišč po časovnih intervalih
 QUEUE - vstop transakcije v čakalno vrsto
 TEST GE - preverimo skladiščne kapacitete ali vhodne količine presegajo skladiščne kapacitete
 TEST E - preverimo, če skladiščne kapacitete dovoljujejo skladiščenje, če dovoljujejo, se transakcija nadaljuje
 SAVEVALUE - če skladiščne kapacitete ne dovoljujejo skladiščenja (so polno zasedene), gre transakcija v čakalno vrsto in stoji. Shranimo vrednosti
 ENTER - dobavna količina vstopi v skladišče (transakcija vstopi v »storage«)
 ADVANCE - trajanje transakcije oz. manipulacij s platišči
 SAVEVALUE - shranimo vrednosti
 DEPART - transakcija zapusti čakalno vrsto in sprosti mesto za naslednjo transakcijo

Odhod

GENERATE - generiranje transakcije odhodov platišč iz skladišča v proizvodnjo po časovnih intervalih
 TABULATE - zapis transakcije v tabelo
 TEST GE - preverimo stanje skladiščnih kapacitet
 LEAVE - izstop transakcije iz »storage«

TERMINATE - transakcija zapusti model
 TERMINATE - če je »storage« prazen, transakcija čaka

Skladišče pnevmatik

Prihod

GENERATE - generiramo transakcije prihodov pnevmatik v skladišče po časovnih intervalih
 QUEUE - vstop transakcije v čakalno vrsto
 TEST GE - preverimo skladiščne kapacitete ali vhodne količine presegajo skladiščne kapacitete
 TEST E - če skladiščne kapacitete dovoljujejo skladiščenje, se transakcija nadaljuje
 SAVEVALUE - preverimo, če skladiščne kapacitete dovoljuje skladiščenje, če dovoljujejo, se transakcija nadaljuje
 ENTER - dobavna količina vstopi v skladišče (transakcija vstopi v »storage«)
 ADVANCE - trajanje transakcije oz. manipulacij s platišči
 SAVEVALUE - shranimo vrednosti
 DEPART - transakcija zapusti čakalno vrsto in sprosti mesto za naslednjo transakcijo

Odhod

GENERATE - generiranje transakcije odhodov pnevmatik iz skladišča v proizvodnjo po časovnih intervalih
 TABULATE - zapis transakcije v tabelo
 TEST GE - preverimo stanje skladiščnih kapacitet
 LEAVE - izstop transakcije iz »storage«
 TERMINATE - transakcija uspešno zapusti model
 TERMINATE - če je »storage« prazen, transakcija čaka

Skladišče vijakov

Prihod

GENERATE - generiramo transakcije prihodov vijakov v skladišče po časovnih intervalih
 QUEUE - vstop transakcije v čakalno vrsto
 TEST - preverimo skladiščne kapacitete ali vhodne količine presegajo skladiščne kapacitete
 TEST E - preverimo, če skladiščne kapacitete dovoljuje skladiščenje, če dovoljujejo, se transakcija nadaljuje
 SAVEVALUE - če skladiščne kapacitete ne dovoljujejo skladiščenja (so polno zasedene), gre transakcija v čakalno vrsto in stoji. Shranimo vrednosti.
 ENTER - dobavna količina vstopi v skladišče (transakcija vstopi

v »storage«)

ADVANCE - trajanje transakcije oz. manipulacij s platišči
 SAVEVALUE - shranimo vrednosti
 DEPART - transakcija zapusti čakalno vrsto in sprosti mesto za naslednjo transakcijo

Odhod

GENERATE - generiranje transakcije odhodov vijakov iz skladišča v proizvodnjo po časovnih intervalih
 TABULATE - zapis transakcije v tabelo
 TEST GE - preverimo stanje skladiščnih kapacitet
 LEAVE - izstop transakcije iz »storage«
 TERMINATE - transakcija zapusti model
 TERMINATE - če je »storage« prazen, transakcija čaka

Simulacija

GENERATE - generiranje transakcij za določeno obdobje trajanja simulacije
 TERMINATE - zmanjšanje števca transakcij za 1
 START - začetek simulacije

Vhodni podatki

Izbrane vhodne podatke opredelimo na osnovi lastnih izkušenj in izračunov trajanja manipulacij, kapacitet prevoznih sredstev itd. Izbrani podatki so za nas najoptimalnejši in se dokaj približajo dejanskemu stanju realnih procesov. Vhodne podatke je potrebno vnesti in urediti v izbrani tekstovni datoteki, ki vsebuje podatke za model in kasnejšo simulacijo (Slika 14.5). Le te lahko poljubno spreminjamo. Za trajanje simulacije določimo čas 7200 minut. Izhodne podatke porazdelitve gibanja zalog zapisujemo v tabelo z lastnostmi [19,19,100] za platišča in pnevmatike in [79,79,100] za vijake.

VHODNI PODATKI

=====

* Time is in minutes

StoRim STORAGE 800 ; Warehouse storage capacity for rims
 StoTire STORAGE 800 ; Warehouse storage capacity for tires
 StoScrew STORAGE 2600 ; Warehouse storage capacity for screws

```

; Frequency distribution table for rims, tires and screws
TableRim          TABLE      S$StoRim,19,19,100
TableTire         TABLE      S$StoTire,19,19,100
TableScrew        TABLE      S$StoScrew,79,79,100

; Initial number of items in the warehouse (at the start time of simulation)
StartNoRim        EQU         200
StartNoTire       EQU         200
StartNoScrew      EQU         800

; Mean arrival time for rims, tires and screws
InRimTimeMean    EQU         180
InTireTimeMean   EQU         360
InScrewTimeMean  EQU         170

; Arrival_time = (InXXTimeMean-InXXTimeRange..InXXTimeMean+InXXTimeRange)
; XX = {Rim, Tire, Screw}
InRimTimeRange   EQU         10
InTireTimeRange  EQU         20
InScrewTimeRange EQU         10

; Capacity of one input vehicle carrying rims, tires and screws
InCapVehRim      EQU         390
InCapVehTire     EQU         800
InCapVehScrew    EQU         1450

; Vehicle upload manipulation time for rims, tires and screws
InVehManipulRim  EQU         6
InVehManipulTire EQU         6
InVehManipulScrew EQU         6

; Mean depart time for rims, tires and screws
OutRimTimeMean   EQU         8
OutTireTimeMean  EQU         8
OutScrewTimeMean EQU         8

; Departure_time = (OutXXTimeMean-OutXXTimeRange..OutXXTimeMean+OutXXTimeRange)
; XX = {Rim, Tire, Screw}
OutRimTimeRange  EQU         1
OutTireTimeRange EQU         1
OutScrewTimeRange EQU         1

; Capacity of one input vehicle carrying rims, tires and screws
OutCapVehRim     EQU         20
OutCapVehTire    EQU         20
OutCapVehScrew   EQU         80

; unloading manipulation time for rims, tires and screws

```



```

InVehManipulRim      EQU      6
InVehManipulTire     EQU      6
InVehManipulScrew    EQU      6

; Unloading_time = unloading manipulation time for rims, tires and screws
InVehManipulRimRange EQU      0
InVehManipulTireRange EQU      0
InVehManipulScrewRange EQU      0

Green                 EQU      1      ; Boolean value for green is 1
Red                   EQU      0      ; Boolean value for red is 0

SimDur                EQU      7200   ; Simulation duration in minutes

```

Simulacijski model

Pred zagonom simulacije je potrebno urediti tekstovne datoteke z vhodnimi podatki, katere kasneje kličemo v GPSS. V GPSS napišemo program za klicanje ustreznih tekstovnih datotek. Simulacijo izvršimo v simulacijskem orodju GPSS World s klikom na *Command in Create Simulation* (Slika 14.6, Slika 14.7 in Slika 14.8).

Analiza izhodnih podatkov

Iskanje najboljšega scenarija, torej čim višja izkoriščenost skladiščnih kapacitet, enakomerne dostavne količine, čim krajše čakalne vrste, čim višja pretočnost itd. se zaključí s primerjavo večjega števila različnih scenarijev. Simulacija na podlagi upoštevanje izbranih vhodnih podatkov poda rezultate, katerih pomen je obrazložen v 14.6:

1. Informacije o simulaciji (Tabela 14.6) in Slika 14.9)

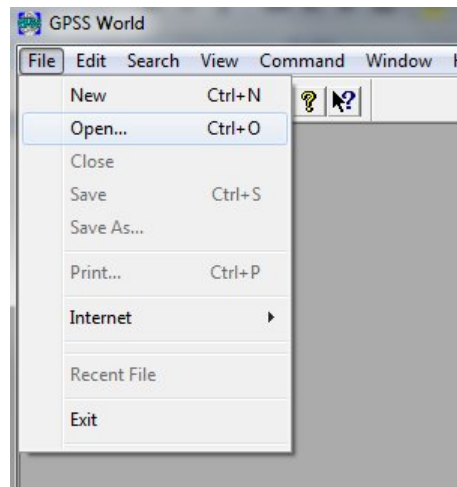
V danem primeru se je simulacija pričela ob vrednosti 0 in zaključila ob vrednosti 7200 – trajala je natanko 7200 minut. Simulacija se izvrši čez 56 blokov, ki smo jih uredili v tekstovni datoteki, katero kličemo v simulacijskem modelu. V simulacijskem modelu uporabimo vrednost *Storages 3*, kar pomeni, da imamo 3 simulacijska "skladišča", ki prikazujejo v našem primeru skladišča treh različnih komponent.

2. Imenovanje in vrednosti blokov (Tabela 14.7)

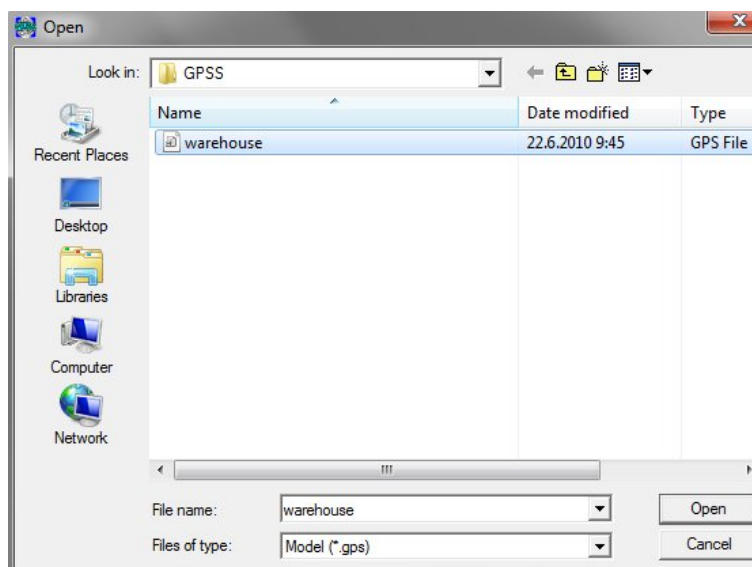
Vsakemu izbranemu imenu, določenemu v tekstovni datoteki, GPSS World določi ime izraženo v številski vrednosti, saj simulacijsko orodje GPSS World operira samo s številčno izraženimi vrednostmi in ne z besedami (Slika 14.10).

Osnovni podatki				
PODJETJE OpenStorage				
DOSTAVA				
Časovni interval dostave			7200 minut = 120 ur = 5 dni	
Frekvenca dostave			Urejena z naključno porazdelitvijo	
Komponente	Sredstvo	PRIHODI	Interval	Odstopanje +/-
Platišča	Tovornjak		180 min	10 min
Pnevmatika	Vlak		360 min	20 min
Vijak	Kombi		170 min	10 min
ODPREMA				
Komponente	Količina	ODHODI	Interval	Odstopanje +/-
Platišča	20 kosov		8 min	1 min
Pnevmatika	20 kosov		8 min	1 min
Vijak	80 kosov		8 min	1 min
<i>Opomba: skupaj sestavlja 5 avtomobilov</i>				
MANIPULACIJE				
Praznjenje viličarja v proizvodnji			Polnjenje viličarja pri prevzemu	
	Trajanje	Odstopanje +/-	Trajanje	
Platišča	6 min	0 min	6 min	
Pnevmatika	6 min	0 min	6 min	
Vijak	6 min	0 min	6 min	
Skladiščne kapacitete		max	Začetno stanje ob simulaciji	
Platišča		800 kosov	200 kosov	
Pnevmatika		800 kosov	200 kosov	
Vijak		2600 kosov	800 kosov	
KAPACITETA prevoznega sredstva	Optimalno	Investicija	Cena viličarja	
Tovornjak	390 kosov/platišča	Nakup viličarja	20.000 €	
Vlak	800 kosov/pnevmatika			
Kombi	1450 kosov/vijakov			

Slika 14.5: Osnovni podatki



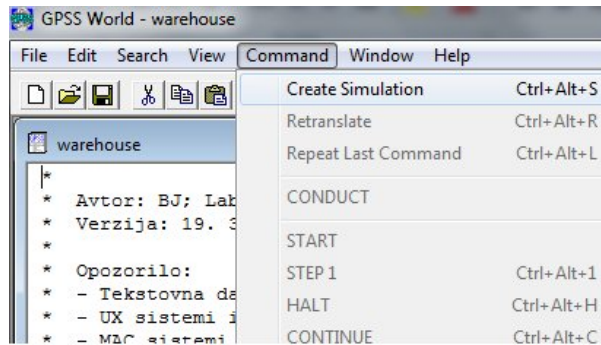
Slika 14.6: Kako odpremo datoteko



Slika 14.7: Zagon datoteke v GPSS

START – END TIME	trajanje simulacije
BLOCKS	število blokov
STORAGES	število skladišč

Tabela 14.6: Informacije o simulaciji



Slika 14.8: Zagon simulacije

```

START TIME          END TIME  BLOCKS  FACILITIES  STORAGES
      0.000          7200.000    56         0           3
  
```

Slika 14.9: Transakcije

NAME	izbrana imena v simulacijskem modelu
VALUE	številčno izražena vrednost izbranega imena

Tabela 14.7: Imenovanje in vrednosti blokov

```

STARTNOSCREW          800.000
STARTNOTIRE           200.000
STORIM                10000.000
STOSCREW              10002.000
  
```

Slika 14.10: Določitev številčno izražene vrednosti imena za operiranje transakcij v izbrani simulaciji

Številčno izraženo ime se začne s številom 10.000 in dalje. V kolikor smo sami določili vrednosti, le te ostanejo nespremenjene.

DOLOČITEV VREDNOSTI

=====

NAME	VALUE
GREEN	1.000
INCAPVEHRIM	390.000
INCAPVEHSCREW	1450.000
INCAPVEHTIRE	800.000
INRIMTIMEAN	180.000
INRIMTIMERANGE	10.000
INSCREWTIMEAN	170.000
INSCREWTIMERANGE	10.000
INTIRETIMEAN	360.000
INTIRETIMERANGE	20.000
INVEHMANIPULRIM	6.000
INVEHMANIPULRIMRANGE	0
INVEHMANIPULSCREW	6.000
INVEHMANIPULSCREWRANGE	0
INVEHMANIPULTIRE	6.000
INVEHMANIPULTIRERANGE	0
OUTCAPVEHRIM	20.000
OUTCAPVEHSCREW	80.000
OUTCAPVEHTIRE	20.000
OUTRIMTIMEAN	8.000
OUTRIMTIMERANGE	1.000
OUTSCREWTIMEAN	8.000
OUTSCREWTIMERANGE	1.000
OUTTIRETIMEAN	8.000
OUTTIRETIMERANGE	1.000
QUEUEINWAREHOUSE	10037.000
RED	0
RIMSTORAGEEMPTY	22.000
SCREWSTORAGEEMPTY	54.000
SIMDUR	7200.000
STARTNORIM	200.000
STARTNOSCREW	800.000
STARTNOTIRE	200.000
STORIM	10000.000
STOSCREW	10002.000
STOTIRE	10001.000
TABLERIM	10003.000
TABLESCREW	10005.000
TABLETIRE	10004.000
TIRESTORAGEEMPTY	38.000

TRAFFICLIGHT

10036.000

3. Potek simulacije

Simuliranje izbranega modela se izvrši z bloki in stavki, ki transakcije "vodijo" skozi izbrani simulacijski model. V praksi to pomeni, da ima neka transakcija svoj začetek in konec. V našem primeru je začetek kreiranje prihodov dostavnih prevoznih sredstev v skladišče, kjer se postavijo v čakalno vrsto za razkladanje komponent. Po razkladanju sledi manipulacija komponent v izbrano skladišče, odvisno od komponente, kjer sledi po skladiščnem procesu odprema komponent v proizvodno linijo. Pri odpremi preverimo kolikokrat se zgodi, da imamo na zalogi premajhno število skladiščnih enot, saj v tem primeru pride do zastoja odpreme in čakanja na dostavo novih komponent.

POTEK SIMULACIJE

```
=====
```

LABEL	LOC	BLOCK TYPE	ENTRY COUNT	CURRENT	COUNT	RETRY
	1	GENERATE	1		0	0
	2	SAVEVALUE	1		0	0
	3	ENTER	1		0	0
	4	ENTER	1		0	0
	5	ENTER	1		0	0
	6	TERMINATE	1		0	0
	7	GENERATE	39		0	0
	8	QUEUE	39		0	0
	9	TEST	39		0	0
	10	TEST	39		0	0
	11	SAVEVALUE	39		0	0
	12	ENTER	39		0	0
	13	ADVANCE	39		0	0
	14	SAVEVALUE	39		0	0
	15	DEPART	39		0	0
	16	TERMINATE	39		0	0
	17	GENERATE	900		0	0
	18	TABULATE	900		0	0
	19	TEST	900		0	0
	20	LEAVE	769		0	0
	21	TERMINATE	769		0	0
RIMSTORAGEEMPTY	22	TERMINATE	131		0	0
	23	GENERATE	19		0	0
	24	QUEUE	19		0	0
	25	TEST	19		0	0
	26	TEST	19		0	0

	27	SAVEVALUE	19	0	0
	28	ENTER	19	0	0
	29	ADVANCE	19	0	0
	30	SAVEVALUE	19	0	0
	31	DEPART	19	0	0
	32	TERMINATE	19	0	0
	33	GENERATE	901	0	0
	34	TABULATE	901	0	0
	35	TEST	901	0	0
	36	LEAVE	770	0	0
	37	TERMINATE	770	0	0
TIRESTORAGEEMPTY	38	TERMINATE	131	0	0
	39	GENERATE	42	0	0
	40	QUEUE	42	0	0
	41	TEST	42	0	0
	42	TEST	42	0	0
	43	SAVEVALUE	42	0	0
	44	ENTER	42	0	0
	45	ADVANCE	42	0	0
	46	SAVEVALUE	42	0	0
	47	DEPART	42	0	0
	48	TERMINATE	42	0	0
	49	GENERATE	902	0	0
	50	TABULATE	902	0	0
	51	TEST	902	0	0
	52	LEAVE	759	0	0
	53	TERMINATE	759	0	0
SCREWSTORAGEEMPTY	54	TERMINATE	143	0	0
	55	GENERATE	1	0	0
	56	TERMINATE	1	0	0

Platišča	Št.
Število dostav	39
Število prihodov viličarjev v skladišče	900
Število odpreme brez čakanja na skladiščne enote	769
Število odpreme s čakanjem na skladiščne enote (zastoj)	131

Tabela 14.8: Analiza poteka simulacije

Pnevmatike	Št.
Število dostav	19
Število prihodov viličarjev v skladišče	901
Število odpreme brez čakanja na skladiščne enote	770
Število odpreme s čakanjem na skladiščne enote (zastoj)	131

Tabela 14.9: Analiza poteka simulacije

4. Analiza poteka simulacije (Tabela 14.8, Tabela 14.9, Tabela 14.10 in Tabela 14.11)

Rezultati razkrivajo slabost celotnega procesa, saj se v vseh treh primerih dogodi zastoj pri odpremi, zaradi premajhnih količin skladiščnih enot. Če bi strošek zastoja (čakanje viličarja, zastoj proizvodnje itd.) ovrednotili s 50 €, bi strošek vseh zastojev znašal 20.250 € (405 zastojev x 50 € strošek zastoja). Odprava zastojev pri odpremi ni tako enostavna naloga, saj je potrebno upoštevati integracijo celotne verige in procesov, kateri integralno vplivajo na celoten sistem. V kolikor bi povečali skladiščne prostore in neposredno skladiščne kapacitete, kar zahteva določene finančne investicije, bi prišlo do pojava povečanja neizkoriščenosti skladiščnih kapacitet in nesmotnosti

Vijaki	Št.
Število dostav	42
Število prihodov viličarjev v skladišče	902
Število odpreme brez čakanja na skladiščne enote	759
Število odpreme s čakanjem na skladiščne enote (zastoj)	143

Tabela 14.10: Analiza poteka simulacije

Skupaj	Št.
Število dostav	101
Število prihodov viličarjev v skladišče	2703
Število odpreme brez čakanja na skladiščne enote	2298
Število odpreme s čakanjem na skladiščne enote (zastoj)	405

Tabela 14.11: Analiza poteka simulacije

MAX	maksimalno število čakajočih
CONT.	trenutno število čakajočih v čakalni vrsti ob zaključku simulacije
ENTRY	število vhodov
ENTRY (0)	število vhodov brez čakanja
AVE. CONT.	povprečno število čakajočih v čakalni vrsti
AVE. TIME	povprečni čas čakanja v čakalni vrsti
AVE. (-0)	povprečni čas transakcij brez čakanja v vrsti

Tabela 14.12: Čakalne vrste

povečanja skladiščnih kapacitet. V kolikor pa zmanjšamo odpreme količine, vplivamo na proizvodni proces in upočasnitev le te, kar zopet vpliva na finančno poslovanje in posledično izgubo tržišča.

4. Čakalne vrste (Tabela 14.12)

V primeru s simulacijo 100-tih vhodov vozil dobimo čakalno vrsto v skladišču, katere maksimalno število čakajočih vozil znaša 2. Postrežb brez čakanja oz. nultih čakalnih vrst vozil ni, kar pripišemo dejstvu, da je potrebno vsako prevozno sredstvo razložiti. Ob zaključku simulacije ni nobenega vozila v čakalni vrsti pri prevzemu v skladišču. Povprečni čas čakanja vozil v čakalni vrsti znaša 6.1 minute. V povprečju v čakalni vrsti stoji 0,086 vozil. Povprečni čas transakcij v nulti čakalni vrsti enak 6.1 minute, saj nimamo vhodov brez

```

QUEUE          MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME  AVE. (-0)  RETRY
QUEUEINWAREHOUSE  2    0   100     0    0.086    6.162    6.162    0

```

Slika 14.11: Čakalne vrste

Naziv	
CAP	kapaciteta skladišča
REM	število neuporabljenih skladiščnih enot ob koncu simulacije
MIN	minimalno število izdelkov v skladišču
MAX	maksimalno število izdelkov v skladišču
ENTRIES	število izdelkov, ki so prišli v skladišče
AVL	skladiščni prostor na voljo (1 na voljo / 2 ni na voljo)
AVE.C.	povprečni čas skladiščenja
UTIL.	izkoriščenost skladiščnih kapacitet

Tabela 14.13: Skladišče

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
STORIM	800	770	0	400	15410	1	172.232	0.215	0	0
STOTIRE	800	800	0	800	15400	1	339.069	0.424	0	0
STOSCREW	2600	1620	0	1520	61700	1	638.023	0.245	0	0

Slika 14.12: Proces skladiščenja

čakanja na postrežbo. V takšnem primeru je povprečni čas čakanja v vrsti enak povprečnemu času transakcij oziroma "čakanja" v nultih čakalnih vrstah (Slika 14.11).

5. Skladišče (Tabela 14.13)

Za naš simulacijski model določimo 3 skladišča *Storages*. Vsako izmed njih predstavlja ločeno skladišče za določeno komponento. Kapacitete skladišč so različne, le te določimo v tekstovni datoteki za vhodne podatke simulacije (Slika 14.12).

Skladišče platišč

Za skladišče platišč določimo kapaciteto 800 kosov. Simulacija prikaže, da je v skladišču maksimalno 400 kosov platišč in minimalno 0 kosov. Ob zaključku simulacije je neizkoriščenih 770 skladiščnih enot oziroma 30 skladiščnih enot na zalogi. V celoti v skladišče prejmemo 15.410 kosov platišč, v povprečju to znaša 172,23 kosov. Izkoriščenost skladiščnih kapacitet znaša 21,5 %. Na podlagi tega lahko sklepamo, da je neizkoriščenost skladiščnih kapacitet precejšna, saj imamo ob maksimalnem številu kosov v skladišču le 50 % izkoriščenost skladiščnih kapacitet ali 400 prostih skladiščnih enot. Možen vzrok za takšno stanje je višja pretočnost, ki je rezultat optimalne razporeditve prihodov, količin,

manipulacijskega časa itd. (glej Sliko 14.12).

Skladišče pnevmatik

Za skladišče pnevmatik določimo kapaciteto 800 kosov. Simulacija pokaže, da je v skladišču maksimalno število pnevmatik 800 kosov in minimalno 0. Ob zaključku simulacije je neizkoriščenih 800 skladiščnih enot oz. skladiščnih enot ni na zalogi. V celoti v skladišče prejmemo 15.400 kosov pnevmatik, kar v povprečju znaša 339,06 kosov. Izkoriščenost skladiščnih kapacitet je 42,4 %. Na podlagi tega sklepamo, da je skladišče nekajkrat v celoti polno in izkoriščeno. Glede na povprečje skladiščnih enot pa lahko trdimo, da je neizkoriščenost skladiščnih kapacitet precejšnja. Prav tako je skladišče nekajkrat prazno, kar priča podatek, da je skladišče ob koncu simulacije prazno. Vzrok za takšno stanje je višja pretočnost, ki je rezultat optimalne razporeditve prihodov, količin in manipulacijskega časa (glej Sliko 14.12).

Skladišče vijakov

Za skladišče vijakov določimo kapaciteto 2.600 kosov. Simulacija pokaže, da je možno skladiščiti maksimalno 1.520 kosov in minimalno 0 kosov vijakov. Ob zaključku simulacije je neizkoriščenih 1.620 skladiščnih enot oz. 980 skladiščnih enot na zalogi. V celoti prejmemo v skladišče 61.700 kosov vijakov, v povprečju to znaša 638,02 kosov. Izkoriščenost skladiščnih kapacitet je 24,5 %. Na podlagi tega sklepamo, da je neizkoriščenost skladiščnih kapacitet precejšnja, saj je ob maksimalnem številu kosov le 58,4 % izkoriščenost skladiščnih kapacitet oz. 1.080 prostih skladiščnih enot. Možen vzrok za takšno stanje je višja pretočnost, ki je rezultat optimalne razporeditve prihodov, količin in manipulacijskega časa.

Ob vsem tem lahko zmanjšamo skladiščne kapacitete za skladiščenje platišč in vijakov na maksimalno število skladiščenih kosov z upoštevanjem določene stopnje rezerve, da ne vplivamo na spremembe v obnašanju celotnega modela. V primeru skladiščenja pnevmatik lahko razmislimo o povečanju kapacitet, predvsem zaradi upoštevanja dejstva varnostnih zalog in izključitve tveganja zastoja proizvodne linije. V skladišče platišč je v času 7.200 minut prispelo 15.410 kosov platišč, kar pomeni zagotoviti proizvodnje 3.852 vozil v enakem obdobju, kar na dnevni ravni predstavlja zagotovitev proizvodnje 770 vozil (glej Sliko 14.12).

Računski prikaz simulacije skladišča platišč

Št. prihodov:

Trajanje simulacije:

15.410 kosov 7200 minut = 120 ur = 5 dni

Št. vozil, ki jih oskrbimo (SKUPAJ):\
15.410 platišč / 4 platišča (komplet) = 3.852 vozil

Št. vozil, ki jih oskrbimo (DAN): Št. vozil, ki jih oskrbimo (URA):
3.852 vozil / 5 dni = 770 vozil 770 vozil / 24 ur = 32 vozil

6. Gibanje zalog (Tabela 14.14)

Prikaz porazdelitve oz. gibanje zalog omogočimo z blokom *Tabulate*, ki zapisuje vrednosti v tabelo (frekvenco in komulativo gibanja zaloge). Vrednost zalog se zapiše vsakič, ko prispe viličar po komponente za odpremo v proizvodno linijo.

V našem primeru je povprečno 179,32 kosov platišč s standardnim odklonom 126,33 kosov. Najpogosteje, kar 131-krat je v skladišču količina platišč med 0 in 19, kar pomeni, da se prav tolikokrat dogodi zastoj, zaradi premajhnih skladiščnih količin. Imamo še povprečno 347,05 kosov pnevmatik s standardnim odklonom 257,96 kosov in 676,93 kosov vijakov s standardnim odklonom 475,6 kosov.

Podatki porazdelitve gibanja skladiščnih kapacitet razkrivajo tudi zanimivost, da se v vseh treh primerih, torej skladiščih vseh treh komponent, največkrat pojavi zasedenost skladiščnih kapacitet med 0 in 19 kosi, kar pomeni zastoj. V primeru skladišča pnevmatik se dogodi, da imamo kar 19-krat maksimalno izkoriščene skladiščne kapacitete, kar pomeni zastoj pri dostavi, saj je dostavljene komponente nemogoče uskladiščiti ob maksimalno izkoriščenih skladiščnih kapacitetah.

PORAZDELITEV GIBANJA ZALOG

=====

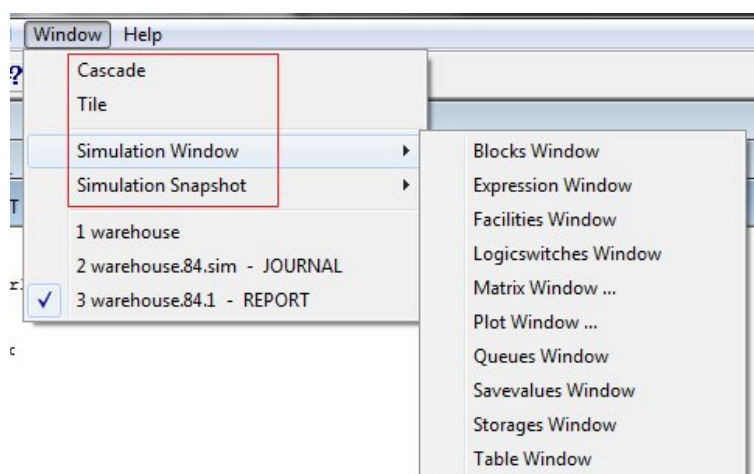
TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
TABLERIM	179.322	126.339		0		
			– –	19.000	131	14.56
			19.000 –	38.000	39	18.89
			38.000 –	57.000	40	23.33
			57.000 –	76.000	40	27.78
			76.000 –	95.000	40	32.22
			95.000 –	114.000	40	36.67
			114.000 –	133.000	40	41.11
			133.000 –	152.000	40	45.56

			152.000	-	171.000		40	50.00
			171.000	-	190.000		40	54.44
			190.000	-	209.000		20	56.67
			209.000	-	228.000		39	61.00
			228.000	-	247.000		39	65.33
			247.000	-	266.000		39	69.67
			266.000	-	285.000		39	74.00
			285.000	-	304.000		39	78.33
			304.000	-	323.000		39	82.67
			323.000	-	342.000		39	87.00
			342.000	-	361.000		39	91.33
			361.000	-	380.000		39	95.67
			380.000	-	399.000		20	97.89
			399.000	-	418.000		19	100.00
TABLETIRE	347.059	257.968				0		
			-	-	19.000		131	14.54
			19.000	-	38.000		20	16.76
			38.000	-	57.000		20	18.98
			57.000	-	76.000		20	21.20
			76.000	-	95.000		20	23.42
			95.000	-	114.000		20	25.64
			114.000	-	133.000		20	27.86
			133.000	-	152.000		20	30.08
			152.000	-	171.000		20	32.30
			171.000	-	190.000		20	34.52
			190.000	-	209.000		20	36.74
			209.000	-	228.000		19	38.85
			228.000	-	247.000		19	40.95
			247.000	-	266.000		19	43.06
			266.000	-	285.000		19	45.17
			285.000	-	304.000		19	47.28
			304.000	-	323.000		19	49.39
			323.000	-	342.000		19	51.50
			342.000	-	361.000		19	53.61
			361.000	-	380.000		19	55.72
			380.000	-	399.000		0	55.72
			399.000	-	418.000		19	57.82
			418.000	-	437.000		19	59.93
			437.000	-	456.000		19	62.04
			456.000	-	475.000		19	64.15
			475.000	-	494.000		19	66.26
			494.000	-	513.000		19	68.37
			513.000	-	532.000		19	70.48
			532.000	-	551.000		19	72.59
			551.000	-	570.000		19	74.69
			570.000	-	589.000		19	76.80
			589.000	-	608.000		19	78.91
			608.000	-	627.000		19	81.02
			627.000	-	646.000		19	83.13

			646.000	-	665.000		19	85.24
			665.000	-	684.000		19	87.35
			684.000	-	703.000		19	89.46
			703.000	-	722.000		19	91.56
			722.000	-	741.000		19	93.67
			741.000	-	760.000		19	95.78
			760.000	-	779.000		0	95.78
			779.000	-	798.000		19	97.89
			798.000	-	817.000		19	100.00
TABLESCREW	676.929	475.600				0		
			-	-	79.000		143	15.85
			79.000	-	158.000		42	20.51
			158.000	-	237.000		42	25.17
			237.000	-	316.000		42	29.82
			316.000	-	395.000		42	34.48
			395.000	-	474.000		42	39.14
			474.000	-	553.000		42	43.79
			553.000	-	632.000		42	48.45
			632.000	-	711.000		42	53.10
			711.000	-	790.000		42	57.76
			790.000	-	869.000		37	61.86
			869.000	-	948.000		41	66.41
			948.000	-	1027.000		41	70.95
			1027.000	-	1106.000		42	75.61
			1106.000	-	1185.000		42	80.27
			1185.000	-	1264.000		42	84.92
			1264.000	-	1343.000		42	89.58
			1343.000	-	1422.000		42	94.24
			1422.000	-	1501.000		42	98.89
			1501.000	-	1580.000		10	100.00

MEAN	povprečje števila izdelkov v skladišču
STD.DEV.	standardni odklon povprečja izdelkov v skladišču
RANGE	interval skladiščenih izdelkov
FREQUENCY	frekvenca pojava določenega intervala skladiščenih izdelkov
CUM.%	kumulativa frekvenca

Tabela 14.14: Gibanje zalog



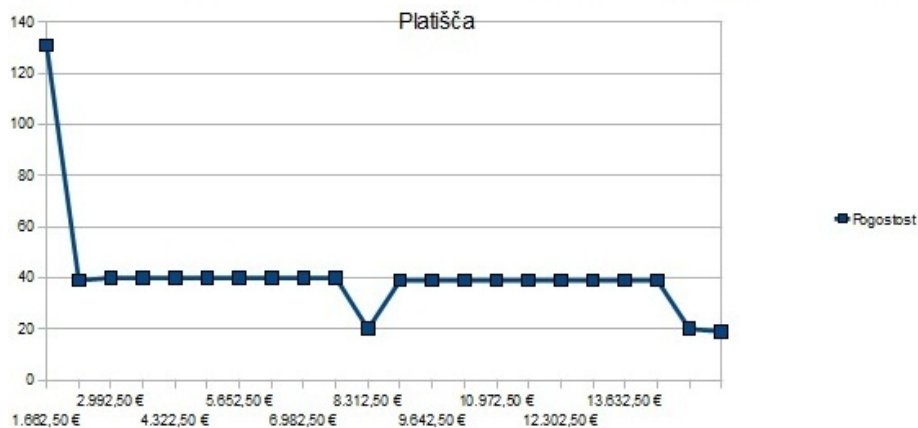
Slika 14.13: Izbira pogleda poročila

S spreminjanjem vhodnih podatkov, lahko tabeli poljubno določimo intervale. V kolikor želimo zelo natančno izvedeti, kako se gibajo zaloge in njihov finančni vidik, v vhodnih podatkih nastavimo za tabelo naslednje intervale: [1,1,1000], in sicer v vrstici **TableRim TABLE S\$StoRim,1,1,1000**.

Izhodne podatke je možno izvoziti v kopiji teksta izbrane tekstovne datoteke (.txt) ali poročila (.gpr) za kasnejše dodatne analize in grafične ponazoritve (v našem primeru porazdelitev gibanja zalog v skladiščnih prostorih in stroškovni vidik izkoriščenosti skladiščnih kapacitet). Poročilo simulacije prikažemo v različnih oknih s klikom na *Window* v orodni vrsti, kjer tudi izbiramo med različnimi prikazi poročila (Slika 14.13).

Za grafičen prikaz porazdelitve gibanja zalog in stroškov zalog platišč (Slika 14.14) uporabimo OO (OpenOffice.org) Preglednico, kamor prenesemo in uredimo dobljene podatke. S pomočjo grafičnega prikaza je tudi razumevanje rezultatov enostavnejše in preglednejše.

Tabela Platišča						
Interval		Pogostost	Strošek zastoja €	Strošek skladiščenja €		Povprečje stroškov skladiščenja
0	19	131	65.500,00 €			
19	38	39		665,00 €	1.330,00 €	997,50 €
38	57	40		1.330,00 €	1.995,00 €	1.662,50 €
57	76	40		1.995,00 €	2.660,00 €	2.327,50 €
76	95	40		2.660,00 €	3.325,00 €	2.992,50 €
95	114	40		3.325,00 €	3.990,00 €	3.657,50 €
114	133	40		3.990,00 €	4.655,00 €	4.322,50 €
133	152	40		4.655,00 €	5.320,00 €	4.987,50 €
152	171	40		5.320,00 €	5.985,00 €	5.652,50 €
171	190	40		5.985,00 €	6.650,00 €	6.317,50 €
190	209	20		6.650,00 €	7.315,00 €	6.982,50 €
209	228	39		7.315,00 €	7.980,00 €	7.647,50 €
228	247	39		7.980,00 €	8.645,00 €	8.312,50 €
247	266	39		8.645,00 €	9.310,00 €	8.977,50 €
266	285	39		9.310,00 €	9.975,00 €	9.642,50 €
285	304	39		9.975,00 €	10.640,00 €	10.307,50 €
304	323	39		10.640,00 €	11.305,00 €	10.972,50 €
323	342	39		11.305,00 €	11.970,00 €	11.637,50 €
342	361	39		11.970,00 €	12.635,00 €	12.302,50 €
361	380	39		12.635,00 €	13.300,00 €	12.967,50 €
380	399	20		13.300,00 €	13.965,00 €	13.632,50 €
399	418	19		13.965,00 €	14.630,00 €	14.297,50 €



Slika 14.14: Grafični prikaz stroškov skladiščenja platišč

Problem 2**Vpliv investicije v nakup sodobnega viličarja na finančno poslovanje.**

S preizkušanjem različnih kombinacij in spreminjanjem vhodnih podatkov v obstoječem simulacijskem modelu dobimo najboljšo varianto, ki ustreza izbranim parametrom in željam. To pomeni, da dobimo najkrajše čakalne vrste, čim višjo pretočnost na vhodni in izhodni strani skladiščnega procesa in čim višjo izkoriščenost skladiščnih kapacitet.

Malce obrnimo zgodbo. Ne iščemo najboljše variante s spreminjanjem vseh vhodnih podatkov, temveč na podlagi potreb proizvodnje poiščemo najoptimalnejšo nabavno količino (dostavo), kar pomeni najoptimalnejše dobavne količine in časovne intervale prihodov dostave treh različnih komponent s tremi različnimi transportnimi sredstvi.

Da se izognemo zastojem v skladišču pri odpremi komponent v proizvodni proces, zaradi premajhnih količin skladiščnih enot, moramo poiskati optimalno dostavo. Na podlagi predvidene proizvodnje količin naročimo optimalne količine treh komponent z upoštevanjem intervalov prihodov dostave. Ureditev dostave je pomembna predvsem zaradi pogodbenih obveznosti, ki jih morata upoštevati obe stranki, tako podjetje (skladišče) kot dobavitelj (s sodelovanjem prevoznika). Obe stranki morata izpolniti v pogodbi določene obveznosti - skladišče je dolžno dostavljene komponente v določenem roku plačati, dobavitelj pa je dolžan naročene komponente v dogovorjenem roku dostaviti (ura dostave, naročene količine, nepoškodovano komponente). Ob neupoštevanju dogovorjenih dostavnih časov dobavitelja (zamude pri dostavi), lahko naše podjetje zahteva povrnitev nastale škode, zaradi zastoja proizvodnega procesa, ki je posledica zamude pri dostavi naročenih količin. Takšna tveganja je potrebno vedno pogodbeno zavarovati. Da bi izključili takšna in drugačna tveganja, je v interesu obeh strank poiskati optimalne dostavne pogoje.

V danem primeru spreminjamo samo vhodne podatke za intervale prihodov treh različnih prevoznih sredstev in naročene količine. Tudi optimalne naroče količine imajo omejitve. Naročene količine je potrebno enkratno dostaviti z enim prevoznim sredstvom in zmogljivostjo slednjega, saj lahko vsako prevozno sredstvo pripelje maksimalno število kosov določenega izdelka, kar je v našem primeru enako optimalni količini.

Zgradba modela temelji na fiksiranju podatkov (odpremni časi in količine, manipulacijski čas, kapacitete prevoznih sredstev (maksimalna izkoriščenost), skladiščne kapacitete in začetno stanje v skladišču) in spreminjanju podatkov (intervali prihodov)(glej Sliko 14.15). Sestava pogodbe in plačilo za opravljeno storitev je odvisna od določitve prihodov (vnaprej dogovorjeni intervale prihodov z dodatkom plačila zamud pri dostavi).

Osnovni podatki				
PODJETJE OpenStorage				
DOSTAVA				
Časovni interval dostave		7200 minut = 120 ur = 5 dni		
Frekvenca dostave		Urejena z naključno porazdelitvijo		
Komponente	Sredstvo	PRIHODI	Interval	Odstopanje +/-
Platišča	Tovornjak		180 min	10 min
Pnevmatika	Vlak		360 min	20 min
Vijak	Kombi		170 min	10 min
ODPREMA				
Komponente	Količina	ODHODI	Interval	Odstopanje +/-
Platišča	20 kosov		8 min	1 min
Pnevmatika	20 kosov		8 min	1 min
Vijak	80 kosov		8 min	1 min
<i>Opomba: skupaj sestavlja 5 avtomobilov</i>				
MANIPULACIJE				
Praznjenje viličarja v proizvodnji			Polnjenje viličarja pri prevzemu	
	Trajanje	Odstopanje +/-	Trajanje	
Platišča	6 min	0 min	6 min	
Pnevmatika	6 min	0 min	6 min	
Vijak	6 min	0 min	6 min	
Skladiščne kapacitete		max	Začetno stanje ob simulaciji	
Platišča		800 kosov	200 kosov	
Pnevmatika		800 kosov	200 kosov	
Vijak		2600 kosov	800 kosov	
KAPACITETA prevoznega sredstva	Optimalno	Investicija	Cena viličarja	
Tovornjak	390 kosov/platišča	Nakup viličarja	20.000 €	
Vlak	800 kosov/pnevmatika			
Kombi	1450 kosov/vijakov			

Slika 14.15: Osnovni podatki

Variabilni podatki v Tabeli 14.15

V nadaljevanju simuliramo dve različni situaciji, ki se lahko pojavita v procesu transporta komponent od dobavitelja do našega skladišča. Simuliranje dveh različnih situacij izvedemo s spreminjanjem variabilnih podatkov, pri čemer ostalih podatkov ne spremenimo. Primerjamo jih z obstoječim modelom (najoptimalnejši model), modeliranih v prvem delu. Situacije:

- situacija 1: sprememba standardnega odklona intervala dostave pnevmatik z vlakom in
- situacija 2: sprememba intervalnega časa dostave platišč s tovornjakom.

Kaj se zgodi, če povečamo standardni odklon intervala dostave? V nadaljevanju preverimo tri pomembne dejavnike, ki vplivajo na izvajanje

PRIHOD	interval	stand. odklon +/-
Platišča	180 min	10 min
Pnevmatike	360 min	20 min
Vijaki	170 min	10 min

Tabela 14.15: Variabilni podatki

Spremembe	Vpliv na
Čakalne vrste	povečanje čakalne vrste, zastoji, število vhodov itd.
Vhodne količine komponent	enakomernost vhodnih količin
Porazdelitev gibanja zalog	stroškovni vidik izkoriščenosti skladiščnih kapacitet

Tabela 14.16: Zanima nas

proizvodnega procesa in stroškovno analizo posledic, ki se dogodijo v danih razmerah. Pogledamo stanje vhodnih količin izbrane komponente, ki jo transportiramo z vlakom, gibanje zalog (porazdelitev) in čakalne vrste. Vsi trije dejavniki vplivajo na osnovno funkcijo skladiščenja, to je zagotavljanje nemotenega proizvodnega procesa in izločitev nihanja dobave komponent.

Zanima nas (Tabela 14.16):

Podatki (Tabela 14.17):

Situacija 1: Sprememba standardnega odklona intervala dostave pnevmatik z vlakom

Velika časovna nihanja v železniškem transportu (transporta pnevmatik z vlakom), nas prisilijo, da povečamo standardni odklon intervala dostave za 10

Platišča	35 €
Pnevmatike	60 €
Vijaki	3 €
Avtomobil	3.000 €
Zastoj proizvodnje	50 €

Tabela 14.17: Podatki

PRIHOD	interval	stand. odklon +/-
Platišča	180 min	10 min
Pnevmatike	360 min	30 min
Vijaki	170 min	10 min

Tabela 14.18: Sprememba variabilnih podatkov

```

InRimTimeMean      EQU      180      ; Mean arrival time for rims
InTireTimeMean     EQU      360      ; Mean arrival time for tires
InScrewTimeMean    EQU      170      ; Mean arrival time for screw
InRimTimeRange     EQU      10      ; Arrival_time
InTireTimeRange    EQU      30      ; Arrival_time
InScrewTimeRange   EQU      10      ; Arrival_time

```

Slika 14.16: Sprememba standardnega odklona za dostavo pnevmatik

minut (iz 20 na 30 minut), kar pomeni, da dobavitelj dostavlja pnevmatike vsakih 360 minut z upoštevanjem standardnega odklona intervala +/- 30 minut. S tem zagotovimo večjo fleksibilnost in prožnost dostave komponent ter izločimo pogodbeno tveganja pri dostavnih zamudah. V kolikor bi standardni odklon zmanjšali, na primer na 10 minut, bi to pomenilo povečanje tveganja zamude pri dostavi, saj so v železniškem transportu, zaradi organizacije dela in slabe infrastrukture velika časovna nihanja in zamude (glej Sliko 14.16).

Sprememba variabilnih podatkov (Situacija 1) (Tabela 14.18)

V tekstovni datoteki za vhodne podatke spremenimo standardni odklon intervala dostave pnevmatik. Povečamo ga iz +/- 20 minut na +/- 30 minut. Simulacijo ponovno poženemo in preverimo spremembe.

Analiza izhodnih podatkov

Novo ugotovitve simulacije primerjamo z obstoječim modelom simulacije:

1. Čakalne vrste

S povečanjem standardnega odklona za 10 minut, so spremembe pri čakalnih vrstah v primerjavi z obstoječim modelom malenkostne in zanemarljive. Število čakajočih v vrsti se ne spremeni, prav tako se ne spremeni čas čakajočih v vrsti in povprečno število čakajočih, spremeni se samo število vhodov, in sicer za 1 % (Slika 14.17).

2. Vhodne količine komponent

Vhodne količine treh komponent razkrivajo neenakomernost in pojav kopičenja zalog v skladišču pnevmatik. Vhodne količine platišč in vijakov so v

QUEUE	MAX	CONT.	ENTRY	ENTRY (0)	AVE.CONT.	AVE.TIME	AVE. (-0)	RETRY
QUEUEINWAREHOUSE	2	0	101	0	0.085	6.064	6.064	0

Slika 14.17: Čakalne vrste za Situacijo 1

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
STORIM	800	790	0	400	15410	1	171.559	0.214	0	0
STOTIRE	800	40	0	800	16200	1	339.240	0.424	0	0
STOSCREW	2600	2100	0	1520	61700	1	647.489	0.249	0	0

Slika 14.18: Vhodne količine komponent za Situacijo 1

primerjavi z obstoječim modelom ostale nespremenjene in zato ne potrebujemo modifikacije proizvodnega procesa, saj zagotavljamo zadostno količino vseh komponent za proizvodnjo 770 vozil. Ostaja pa kar 800 kosov neizkoriščenih zalog pnevmatik, nadaljevanje takšnega trenda pomeni zasičenost skladiščnih kapacitet in zastoj pri uskladiščenju komponent, kar prikazuje Slika 14.18. Skladišče pnevmatik v obstoječem stanju kar 19-krat doseže maksimalno izkoriščenost skladiščnih kapacitet, z dodatnim kopičenjem zalog bi to številko samo še povečevali.

V kolikor se nadaljuje trend kopičenja zalog, te predstavljalo za izbrano obdobje strošek v višini 48.000 €, kar predstavlja 4,9 % vrednosti dobavljene količine komponent oziroma 5,2 % vrednosti potrebnih količin komponent za izbrano proizvodno količino.

Izračuni

Število dodatnih zalog x vrednost komponente
 = strošek dodatnih zalog
 800 kosov x 60 € = 48.000 €

Strošek dodatnih zalog / vrednost dobavljene količine x 100
 = delež dodatnih zalog na dobavljene količine
 48.000 € / 16.200 kosov x 60 € x 100 = 4,9 \%

Strošek dodatnih zalog / vrednost potrebnih količin x 100 =
 48.000 € / 15.400 x 60 € x 100 = 5,2 \%

Strošek kopičenja zalog je za podjetje nepotreben in nepričakovan strošek, ki ga je potrebno v najkrajšem možnem času odpraviti ali vsaj zmanjšati, z upoštevanjem vidika zmanjšanja tveganja morebitnega zastoja proizvodne linije.

3. Porazdelitev gibanja zalog

S povečanjem standardnega odklona za 10 minut, so spremembe pri porazdelitvi gibanja zalog malenkostne, prav tako so malenkostni tudi stroški. Pri odpremi zmanjšamo število zastojev v skladišču platišč za 1, v skladišču vijakov za 10, v skladišču pnevmatik pa jih povečamo za 1.

Omenimo lahko tudi minimalno povečanje maksimalne izkoriščenosti skladiščnih kapacitet za pnevmatike.

PORAZDELITEV GIBANJA ZALOG ZA SITUACIJO 1

=====

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
TABLERIM	179.344	126.311		0		
			- - 19.000		130	14.44
			19.000 - 38.000		40	18.89
			38.000 - 57.000		40	23.33
			57.000 - 76.000		40	27.78
			76.000 - 95.000		40	32.22
			95.000 - 114.000		40	36.67
			114.000 - 133.000		40	41.11
			133.000 - 152.000		40	45.56
			152.000 - 171.000		40	50.00
			171.000 - 190.000		40	54.44
			190.000 - 209.000		20	56.67
			209.000 - 228.000		39	61.00
			228.000 - 247.000		39	65.33
			247.000 - 266.000		39	69.67
			266.000 - 285.000		39	74.00
			285.000 - 304.000		39	78.33
			304.000 - 323.000		39	82.67
			323.000 - 342.000		39	87.00
			342.000 - 361.000		39	91.33
			361.000 - 380.000		39	95.67
			380.000 - 399.000		20	97.89
			399.000 - 418.000		19	100.00
TABLETIRE	347.655	258.639		0		
			- - 19.000		132	14.60
			19.000 - 38.000		20	16.81
			38.000 - 57.000		20	19.03
			57.000 - 76.000		20	21.24
			76.000 - 95.000		20	23.45

			95.000	-	114.000		20	25.66
			114.000	-	133.000		20	27.88
			133.000	-	152.000		20	30.09
			152.000	-	171.000		20	32.30
			171.000	-	190.000		20	34.51
			190.000	-	209.000		20	36.73
			209.000	-	228.000		19	38.83
			228.000	-	247.000		19	40.93
			247.000	-	266.000		19	43.03
			266.000	-	285.000		19	45.13
			285.000	-	304.000		19	47.23
			304.000	-	323.000		19	49.34
			323.000	-	342.000		19	51.44
			342.000	-	361.000		19	53.54
			361.000	-	380.000		19	55.64
			380.000	-	399.000		0	55.64
			399.000	-	418.000		19	57.74
			418.000	-	437.000		19	59.85
			437.000	-	456.000		19	61.95
			456.000	-	475.000		19	64.05
			475.000	-	494.000		19	66.15
			494.000	-	513.000		19	68.25
			513.000	-	532.000		19	70.35
			532.000	-	551.000		19	72.46
			551.000	-	570.000		19	74.56
			570.000	-	589.000		19	76.66
			589.000	-	608.000		19	78.76
			608.000	-	627.000		19	80.86
			627.000	-	646.000		19	82.96
			646.000	-	665.000		19	85.07
			665.000	-	684.000		19	87.17
			684.000	-	703.000		19	89.27
			703.000	-	722.000		19	91.37
			722.000	-	741.000		19	93.47
			741.000	-	760.000		19	95.58
			760.000	-	779.000		0	95.58
			779.000	-	798.000		20	97.79
			798.000	-	817.000		20	100.00
TABLESCREW	684.621	472.159				0		
			-	-	79.000		133	14.81
			79.000	-	158.000		42	19.49
			158.000	-	237.000		42	24.16
			237.000	-	316.000		42	28.84
			316.000	-	395.000		42	33.52
			395.000	-	474.000		42	38.20
			474.000	-	553.000		42	42.87
			553.000	-	632.000		43	47.66
			632.000	-	711.000		43	52.45
			711.000	-	790.000		43	57.24

790.000	-	869.000	38	61.47
869.000	-	948.000	42	66.15
948.000	-	1027.000	42	70.82
1027.000	-	1106.000	42	75.50
1106.000	-	1185.000	42	80.18
1185.000	-	1264.000	42	84.86
1264.000	-	1343.000	42	89.53
1343.000	-	1422.000	42	94.21
1422.000	-	1501.000	42	98.89
1501.000	-	1580.000	10	100.00

PRIHOD	interval	stand. odklon +/-
Platišča	240 min	10 min
Pnevmatike	360 min	20 min
Vijaki	170 min	10 min

Tabela 14.19: Sprememba variabilnih podatkov

Stroškovni vidik izkoriščenosti skladiščnih kapacitet izdelamo v OOo (Preglednica), kamor kopiramo in nato uredimo podatke porazdelitve gibanja zalog. Vsako komponento in zastoj opredelimo s ceno (Slika 14.19).

Stroškovna analiza obeh modelov, obstoječega modela in Situacije 1, pokaže, da so razlike pri porazdelitve zalog malenkostne, stroški, ki nastanejo v obeh modelih, pa se razlikujejo za 47.500 €. Kljub temu, da smo z 10 minutno spremembo standardnega odklona intervala dostave pnevmatik dosegli zmanjšanje števila zastojev in z njimi povezanimi stroški, smo na drug strani povišali stroške dodatnih zalog v višini 48.000 € (Slika 14.20).

Situacija 2: Sprememba intervalnega časa dostave platišč s tovornjakom

V podjetju se odločimo, da spremenimo intervalni čas dostave platišč, in sicer iz dosedanjih 180 minut na 240 minut, kar pomeni, da povečamo dostavni čas za 60 minut (Slika 14.21). V najboljšem primeru imamo lahko tako namesto 8 dostav samo 6 dostav na dan. Kaj se zgodi v primeru spremembe intervalnega časa dostave platišč? Kako se obnaša celotna veriga?

Sprememba variabilnih podatkov (Situacija 2) (Tabela 14.19)

Analiza izhodnih podatkov

Simulacija poda naslednje ugotovitve (v primerjavi z obstoječim modelom simulacije):

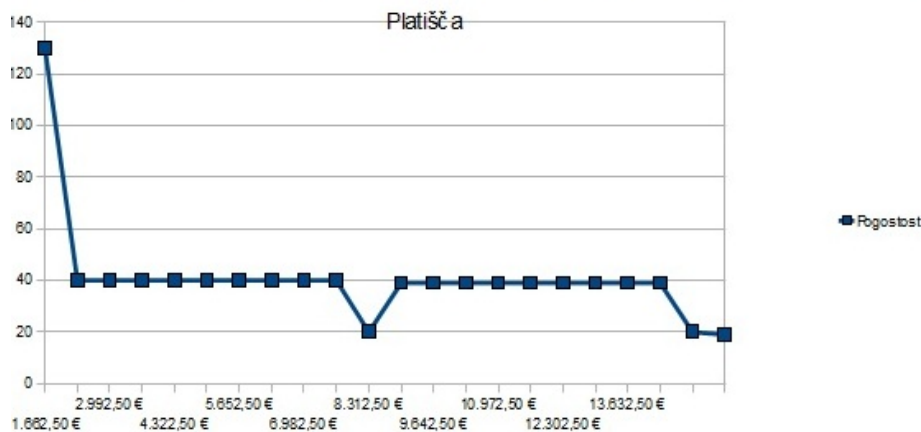
1. Čakalne vrste

S povečanjem intervalnega časa iz 180 na 240 minut, so spremembe precejšnje. V primerjavi z obstoječim modelom se število vhodov zmanjša za 10 % oz. 10. V povprečju je še manjše število čakajočih. Povprečni čas čakajočih se bistveno ne spremeni, prav tako se ne spremeni tudi maksimalno število čakajočih v vrsti za razkladanje (Slika 14.22).

2. Vhodne količine komponent

Vhodne količine treh komponent razkrivajo neenakomernost in primanjkljaj zalog v skladišču platišč (Slika 14.23). Zaradi tega se pojavi zastoj proizvodne

Tabela Platišča			Strošek zastoja €	Strošek skladiščenja €		Povprečje stroškov skladiščenja
Interval	Pogostost	6.500,00 €				
0	19	130				
19	38	40		665,00 €	1.330,00 €	997,50 €
38	57	40		1.330,00 €	1.995,00 €	1.662,50 €
57	76	40		1.995,00 €	2.660,00 €	2.327,50 €
76	95	40		2.660,00 €	3.325,00 €	2.992,50 €
95	114	40		3.325,00 €	3.990,00 €	3.657,50 €
114	133	40		3.990,00 €	4.655,00 €	4.322,50 €
133	152	40		4.655,00 €	5.320,00 €	4.987,50 €
152	171	40		5.320,00 €	5.985,00 €	5.652,50 €
171	190	40		5.985,00 €	6.650,00 €	6.317,50 €
190	209	20		6.650,00 €	7.315,00 €	6.982,50 €
209	228	39		7.315,00 €	7.980,00 €	7.647,50 €
228	247	39		7.980,00 €	8.645,00 €	8.312,50 €
247	266	39		8.645,00 €	9.310,00 €	8.977,50 €
266	285	39		9.310,00 €	9.975,00 €	9.642,50 €
285	304	39		9.975,00 €	10.640,00 €	10.307,50 €
304	323	39		10.640,00 €	11.305,00 €	10.972,50 €
323	342	39		11.305,00 €	11.970,00 €	11.637,50 €
342	361	39		11.970,00 €	12.635,00 €	12.302,50 €
361	380	39		12.635,00 €	13.300,00 €	12.967,50 €
380	399	20		13.300,00 €	13.965,00 €	13.632,50 €
399	418	19		13.965,00 €	14.630,00 €	14.297,50 €



Slika 14.19: Stroški skladiščenja platišč za Situacijo 1

Komponenta	Obstoječ model			Situacija 1		
	Interval	Pogost.	Povp. strošek	Interval	Pogost.	Povp. strošek
Platišča	Zastoj	131	50 €	Zastoj	130	50 €
	190-209	20	6.982 €	190-209	20	6.982 €
	399-418	19	14.297 €	399-418	19	14.297 €
Pnevmatike	Zastoj	131	50 €	Zastoj	132	50 €
	399-418	19	24.510 €	399-418	19	24.510 €
	798-817	19	48.450 €	798-817	20	48.450 €
Vijaki	Zastoj	143	50 €	Zastoj	133	50 €
	790-869	37	2.488 €	790-869	38	2.488 €
	1501-1580	10	4.621 €	1501-1580	10	4.621 €
<hr/>						
Obstoječi model	Zastoj	405	50 €	20.250 €		
	Zaloge	0 €	0 €	0 €		
			Skupaj	20.250 €		
<hr/>						
Situacija 1	Zastoj	395	50 €	19.750 €		
	Zaloge	800	60 €	48.000 €		
			Skupaj	67.750 €		

Slika 14.20: Spremembe pri intervalih porazdelitve gibanja zalog – Situacija 1

```

InRimTimeMean    EQU      240 + 60 min    ; Mean arrival
InTireTimeMean   EQU      360                ; Mean arrival
InScrewTimeMean  EQU      170                ; Mean arrival

InRimTimeRange   EQU      10                ; Arrival_time
InTireTimeRange  EQU      20                ; Arrival_time
InScrewTimeRange EQU      10                ; Arrival_time

```

Slika 14.21: Sprememba intervalnega časa za dostavo platišč

```

QUEUE           MAX CONT. ENTRY ENTRY(0) AVE.CONT. AVE.TIME  AVE.(-0) RETRY
QUEUEINWAREHOUSE 2     0     90     0     0.077     6.125     6.125  0

```

Slika 14.22: Čakalne vrste za Situacijo 2

STORAGE	CAP.	REM.	MIN.	MAX.	ENTRIES	AVL.	AVE.C.	UTIL.	RETRY	DELAY
STORIM	800	790	0	400	11510	1	128.630	0.161	0	0
STOTIRE	800	640	0	800	15400	1	340.722	0.426	0	0
STOSCREW	2600	1700	0	1520	61700	1	641.669	0.247	0	0

Slika 14.23: Vhodne količine komponent za Situacijo 2

linije, ki predstavlja (za podjetje) precejšnje stroške. Strošek zastoja proizvodne linije lahko opredelimo z izgubo čistega dobička na proizvedeno vozilo, v danem primeru znaša 3.000 €. Z danimi vhodnimi količinami zagotovimo proizvodnjo v obsegu 575 vozil na dan, kar je 195 vozil manj oz. 25 %. Izguba čistega dobička zaradi zmanjšanja proizvodnje znaša kar 585.000 €.

Izračuni

Zmanjšano število vozil x vrednost izgube čistega dobička na vozilo
= celotna vrednost izgube
 $195 \text{ vozil} \times 3.000 \text{ €} = 585.000 \text{ €}$

Obstoječe število proizvedenih vozil / število proizvedenih vozil v Situaciji 2 x 100
= delež zmanjšanja proizvodnje vozil
 $770 \text{ vozil} / 575 \text{ vozil} \times 100 = 25 \%$

Število dodatnih pnevmatik x vrednost komponente = strošek dodatnih zalog
 $3.900 \text{ kosov} \times 35 \text{ €} = 136.500 \text{ €}$

Število dodatnih vijakov x vrednost komponente = strošek dodatnih zalog
 $15.560 \text{ kosov} \times 3 \text{ €} = 46.980 \text{ €}$

V primeru premajhnih količin ene komponente zmanjšamo in prilagodimo proizvodno kapaciteto na spodnjo mejo oziroma na najnižjo vhodno količino (vhodna količina platišč). Ob tem se pojavita dva scenarija:

- izguba tržišča in dobička, zaradi zmanjšanja proizvodnih količin in nezadostitve tržišča in
- kopičenje zalog drugih dveh komponent, v primeru, ko proizvodna količina zadosti potrebam tržišča.

Potrebno je analizirati oba scenarija. V 1. scenariju vrednost izgube dobička, zaradi nezadostitve tržišča v izbranem obdobju, znaša 585.000 €. V 2. scenariju strošek kopičenja zalog drugih dveh komponent znaša 136.500 € za pnevmatike in 46.980 € za vijake. Skupaj torej 183.480 €. Tem vrednostim moramo prišteti še stroške zastojev, ki nastanejo zaradi nezadostnih količin ob prispetju viličarja.

3. Porazdelitev gibanja zalog

S povečanjem intervala dostave platišč za 60 minut so spremembe pri porazdelitvi gibanja zalog precejšnje, kar se pozna tudi pri vhodnih količinah iste komponente. Zmanjšanje vhodne količine platišč neposredno vpliva na spremembo porazdelitve gibanja zalog. Dogodi se zastoj pri odpremi platišč (329-krat), kar predstavlja strošek v višini 16.450€. Prav tako povzroči 3 zastoje več pri odpremi pnevmatik. Pri odpremi vijakov se število zastojev zmanjša za 1. Porazdelitve gibanja zalog platišč se bistveno spreminja glede na obstoječe stanje, predvsem na račun spremenjenih vhodnih količin.

PORAZDELITEV GIBANJA ZALOG ZA SITUACIJO 2

=====

TABLE	MEAN	STD.DEV.	RANGE	RETRY	FREQUENCY	CUM.%
TABLERIM	134.314	132.790		0		
			- -		329	36.39
			19.000 -		30	39.71
			38.000 -		30	43.03
			57.000 -		30	46.35
			76.000 -		30	49.67
			95.000 -		30	52.99
			114.000 -		30	56.31
			133.000 -		30	59.62
			152.000 -		30	62.94
			171.000 -		30	66.26
			190.000 -		15	67.92
			209.000 -		29	71.13
			228.000 -		29	74.34
			247.000 -		29	77.54
			266.000 -		29	80.75
			285.000 -		29	83.96
			304.000 -		29	87.17
			323.000 -		29	90.38
			342.000 -		29	93.58
			361.000 -		29	96.79
			380.000 -		15	98.45
			399.000 -		14	100.00
TABLETIRE	348.192	258.287		0		
			- -		134	14.96
			19.000 -		19	17.08
			38.000 -		19	19.20
			57.000 -		19	21.32
			76.000 -		19	23.44
			95.000 -		19	25.56
			114.000 -		19	27.68
			133.000 -		19	29.80
			152.000 -		19	31.92
			171.000 -		20	34.15
			190.000 -		20	36.38
			209.000 -		19	38.50
			228.000 -		19	40.63
			247.000 -		19	42.75
			266.000 -		19	44.87
			285.000 -		19	46.99
			304.000 -		19	49.11
			323.000 -		19	51.23
			342.000 -		19	53.35
			361.000 -		19	55.47
			380.000 -		0	55.47

			399.000	-	418.000	19	57.59
			418.000	-	437.000	19	59.71
			437.000	-	456.000	19	61.83
			456.000	-	475.000	19	63.95
			475.000	-	494.000	19	66.07
			494.000	-	513.000	19	68.19
			513.000	-	532.000	19	70.31
			532.000	-	551.000	19	72.43
			551.000	-	570.000	19	74.55
			570.000	-	589.000	19	76.67
			589.000	-	608.000	19	78.79
			608.000	-	627.000	19	80.92
			627.000	-	646.000	19	83.04
			646.000	-	665.000	19	85.16
			665.000	-	684.000	19	87.28
			684.000	-	703.000	19	89.40
			703.000	-	722.000	19	91.52
			722.000	-	741.000	19	93.64
			741.000	-	760.000	19	95.76
			760.000	-	779.000	0	95.76
			779.000	-	798.000	19	97.88
			798.000	-	817.000	19	100.00
TABLESCREW	678.137	475.006				0	
			-	-	79.000	142	15.74
			79.000	-	158.000	42	20.40
			158.000	-	237.000	42	25.06
			237.000	-	316.000	42	29.71
			316.000	-	395.000	42	34.37
			395.000	-	474.000	42	39.02
			474.000	-	553.000	42	43.68
			553.000	-	632.000	42	48.34
			632.000	-	711.000	42	52.99
			711.000	-	790.000	42	57.65
			790.000	-	869.000	37	61.75
			869.000	-	948.000	41	66.30
			948.000	-	1027.000	42	70.95
			1027.000	-	1106.000	42	75.61
			1106.000	-	1185.000	42	80.27
			1185.000	-	1264.000	42	84.92
			1264.000	-	1343.000	42	89.58
			1343.000	-	1422.000	42	94.24
			1422.000	-	1501.000	42	98.89
			1501.000	-	1580.000	10	100.00

Stroškovna analiza obeh modelov, obstoječega modela in Situacije 2 pokaže da so razlike pri porazdelitve zalog precejšnje. Stroški, ki nastanejo v obeh

modelih, se v 1. scenariju razlikuje za 595.000€. V 2. scenariju pa za 146.500 €. V 1.scenariju je višina stroškov odvisna od izgube dobička in zastojev pri odpremi, torej zmanjšanja proizvodne količine in posledično manjše prodaje ob nespremenjenih potrebah tržišča. V 2. scenariju pa je višina stroškov odvisna od prekomernih zalog oz. od kopičenja zalog in zastojev pri odpremi. Poleg tega je 2. scenarij pogojen z zadovoljitvijo trga - če proizvedena količina vozil zadovolji potrebe tržišča, predstavlja kopičenje zalog dodaten strošek. Vse to prikazuje Slika 14.25.

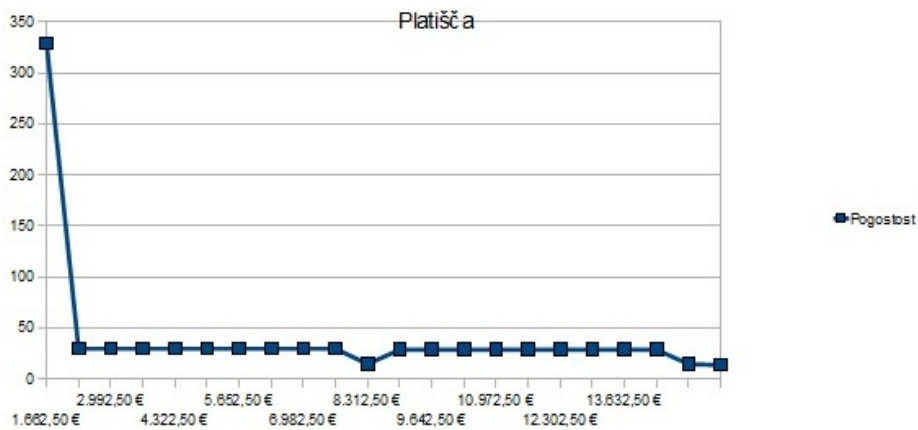
Povzetek

S simulacijskim orodjem GPSS World simuliramo materialni tok dostave, skladiščenja in odpreme treh različnih komponent, ki jih dostavimo s tremi različnimi prevoznimi sredstvi. Vsaka komponenta se skladišči ločeno, v lastnem skladišču. Optimiranje časov prihodov, odpreme, manipulacij, količin itd. vpliva na višanje pretočnosti materialnega toka skozi celotno oskrbno verigo, kar vpliva na manjše čakalne vrste, višje količine dostavljenih komponent, nizko izkoriščenost skladiščnih kapacitet itd. Tako lahko s simuliranjem različnih scenarijev (različni vhodni podatki) poiščemo najboljši scenarij, ki ustreza določenim kriterijem in analiziramo bodoče obnašanje sistema ob predpostavki "kaj če". V primeru spremembe enega od vhodnih podatkov (čas prihoda prevoznega sredstva, količina dostavljene komponente, manipulacijski čas) lahko v "trenutku« analiziramo kaj se zgodi in ob tem identificiramo morebitne scenarije kot so daljše čakalne vrste in manjše dostavljene količine določene komponente.

V nadaljevanju s spreminjanjem časov prihodov dostavnih vozil prikažemo, kako vplivajo zamude pri dostavi komponent na celotni nadaljnji proces. Kaj se zgodi, ko zamudi dostava pnevmatik z vlakom? Kakšne stroške imamo pri zastoju proizvodne linije in kolikšni so stroški zalog v primeru polnega skladišča?

Simulacijsko orodje GPSS World je zelo praktično in uporabno za simuliranje dogodkov. Z "kaj če" analizo in simulacijo dogajanja identificiramo dogodke, katere želimo izključiti, saj za nas predstavljajo določeno tveganje (zastoji, stroški zalog itd.).

Tabela Platišča					
Interval	Pogostost	Strošek zastoja €	Strošek skladiščenja €	Povprečje stroškov skladiščenja	
0	19	329			
19	38	30	164.500,00 €		
38	57	30		665,00 €	1.330,00 €
57	76	30		1.330,00 €	1.995,00 €
76	95	30		1.995,00 €	2.660,00 €
95	114	30		2.660,00 €	3.325,00 €
114	133	30		3.325,00 €	3.990,00 €
133	152	30		3.990,00 €	4.655,00 €
152	171	30		4.655,00 €	5.320,00 €
171	190	30		5.320,00 €	5.985,00 €
190	209	15		5.985,00 €	6.650,00 €
209	228	29		6.650,00 €	7.315,00 €
228	247	29		7.315,00 €	7.980,00 €
247	266	29		7.980,00 €	8.645,00 €
266	285	29		8.645,00 €	9.310,00 €
285	304	29		9.310,00 €	9.975,00 €
304	323	29		9.975,00 €	10.640,00 €
323	342	29		10.640,00 €	11.305,00 €
342	361	29		11.305,00 €	11.970,00 €
361	380	29		11.970,00 €	12.635,00 €
380	399	15		12.635,00 €	13.300,00 €
399	418	14		13.300,00 €	13.965,00 €
				13.965,00 €	14.630,00 €



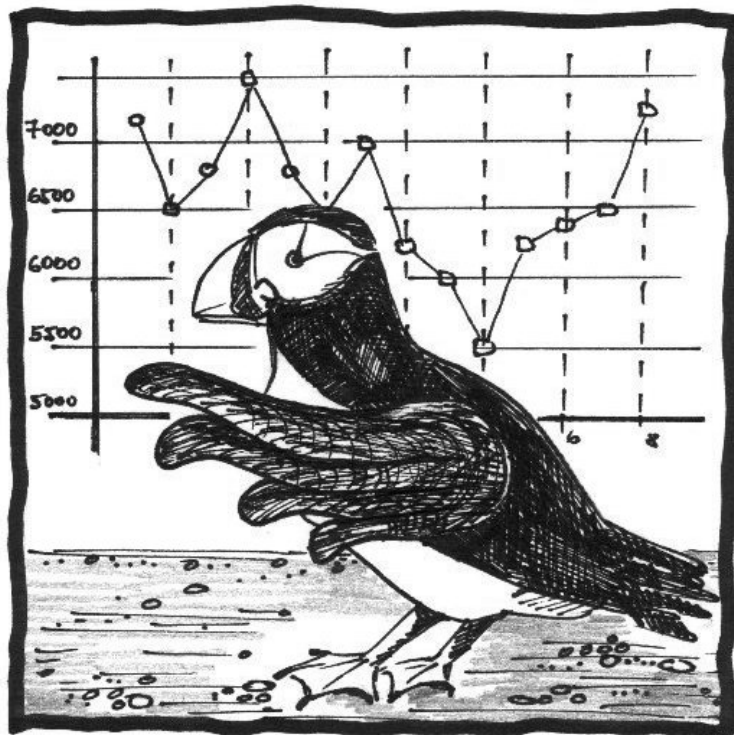
Slika 14.24: Strošek skladiščenja platišč za Situacijo 2

Komponenta	Obstoječ model			Situacija 2		
	Interval	Pogost.	Povp. strošek	Interval	Pogost.	Povp. strošek
Platišča	Zastoj	131	50 €	Zastoj	329	50 €
	190-209	20	6.982 €	190-209	15	6.982 €
	399-418	19	14.297 €	399-418	14	14.297 €
Pnevmatike	Zastoj	131	50 €	Zastoj	134	50 €
	399-418	19	24.510 €	399-418	19	24.510 €
	798-817	19	48.450 €	798-817	19	48.450 €
Vijaki	Zastoj	143	50 €	Zastoj	142	50 €
	790-869	37	2.488 €	790-869	37	2.488 €
	1501-1580	10	4.621 €	1501-1580	10	4.621 €
Obstoječi model	Zastoj	405	500 €	20.250 €		
	Zaloge	0 €	0 €	0 €		
			Skupaj	20.250 €		
Situacija 2	Dva scenarija					
	Zastoj		605	50 €	30.250 €	
Scenarij 1	Izpad proizvodnje		195	3.000 €	585.000 €	
Scenarij 2	Zaloge		Obe komponente		136.500 €	
Scenarij 1			Skupaj		615.250 €	
Scenarij 2			Skupaj		166.750 €	

Slika 14.25: Spremembe pri intervalih porazdelitve gibanja zalog – Situacija 2

Poglavje 15

SCILAB - numerično reševanje in analiziranje podatkov



Numerično reševanje problemov s programiranjem
Kvantitativne metode napovedovanja
Primer: napoved prodaje vozil

15.1 Teoretično ozadje

15.1.1 Napovedovanje povpraševanja

Z napovedovanjem ocenjujemo verjetne razvoje v prihodnosti. Planiranje ne oznanja samo verjetnega, ampak na osnovi verjetnega postavlja tisto, kar je zaželeno. Pogosto se s pojmom "napovedovanje povpraševanja" pojavljata dva pojma, in sicer napovedovanje in predvidevanje. Pri čemer naj bi pojem "napovedovanje" ustrezal angleškemu izrazu *forecasting*, "predvidevanje" pa izrazu *prognosis*. Po pregledu različne literature ugotavljamo, da avtorji skoraj povsod govorijo o obeh izrazih, kot o enem samem.

Glede na časovni horizont za katerega napovedujemo povpraševanja ločimo dolgoročno, srednjeročno in kratkoročno napovedovanje. Za dolgoročno napovedovanje je značilno, da poteka daljše časovno obdobje in se praviloma izvaja za skupine proizvodov. Na podlagi tega napovedovanja se odločamo o fiksnih zmogljivostih. Srednjeročno napovedovanje je napovedovanje, ki zavzema obdobje nekje od 6 do 18 mesecev. Na podlagi tega poskušamo ugotoviti predvsem sezonska nihanja na ravni skupin proizvodov, katerim nato prilagajamo proizvodnjo. Kratkoročno napovedovanje se izvaja za obdobje od nekaj tednov pa do nekaj mesecev in služi (predvsem) kot osnova operativnemu planiranju.

Glede na dani problem se odločimo za vrsto in temu primerno metodo napovedovanja. Metode v osnovi delimo na dve skupini, in sicer: kvalitativne in kvantitativne. Za kvalitativne metode je značilno, da so običajno subjektivne narave. Kvantitativne metode napovedovanja povpraševanja temeljijo na podatkih iz preteklosti in statističnih orodjih. Med kvantitativne metode uvrščamo dva ključna pristopa. Prvi pristop so t.i. ekstrapolacijske metode, ki predpostavljajo, da je gibanje nekega pojava prvenstveno funkcija časa. Drugi pristop, imenovan vzročne metode temelji na predpostavki, da je gibanje nekega pojava v korelaciji z mnogimi zunanjimi faktorji, izmed katerih je eden lahko tudi čas. Kvalitativne metode so uporabne predvsem za dolgoročne napovedi ali pa za napovedi, ko nimamo na voljo dovolj podatkov o preteklem gibanju pojavov. V to vrsto uvrščamo Delfi metodo, ki jo uporabljamo za dolgoročno napovedovanje, oceno menagementa osebja, oceno prodajnega osebja, anketiranje, tržne raziskave ipd.

15.1.2 Kvantitativne metode napovedovanja

V izbranem primeru se osredotočimo na kvantitativne metode napovedovanja, ki kot osnovo napovedi, uporabljajo matematične modele, ki so zasnovani na podatkih iz preteklosti. Te metode temeljijo na predpostavki, da gibanja v

prihodnosti lahko predvidimo na podlagi gibanj v preteklosti. Govorimo o treh najpogosteje uporabljenih metodah, katere v nadaljevanju prikažemo na konkretnem primeru:

- Regresija meri odvisnost dveh slučajnih spremenljivk – kakšen vpliv ima ena na drugo. Linearna regresija je metoda, s katero ob znani spremenljivki X napovemo, koliko bo Y . Y je odvisna spremenljivka, X pa neodvisna (imenovana tudi prediktor).
- Holtova linearna metoda se imenuje po statistiku Holtu. Metoda uporablja dva faktorja glajenja - Alfa, ki predstavlja eksponentno glajenje in Beta, ki predstavlja trend v opazovanem obdobju. Obe konstanti se nahajata v razponu med 0 in 1. Praksa je pokazala, da mora biti vrednost Beta manjša od Alfe, Beta v razponu od 0,005 in 0,2 ter Alfa med 0,02 in 0,5.
- Brownova metoda eksponentnega glajenja je metoda pri kateri se upoštevajo vsi podatki, ki so na voljo, pri čemer se starejšim podatkom dodeli nižja utež. Gre za enokoračno metodo napovedovanja, kjer tvorimo napoved za en korak naprej v časovnem horizontu. Uporablja se v primerih, ko pri meritvah povpraševanja ni zaznati kakšnega spreminjanja trenda, niti ne cikličnih in sezonskih nihanj.

15.2 O programskem orodju

Scilab je programsko orodje za računanje numeričnih funkcij. Uporabno je za računanje z različnimi oblikami matematičnih zapisov in struktur z vsemi znanimi oblikami števil (z matrikami, polinomi, objekti ipd.). Z danim orodjem analiziramo linearne in nelinearne dinamične sisteme. Programsko orodje uporabimo kot pripomoček za numerično optimizacijo, kot sta npr. linearno in nelinearno programiranje. Razvili so ga na inštitutu INRIA v Franciji in je prosto odprto orodje. Prepozna funkcije kot podatkovne objekte, z njimi računa ali jih pretvori v druge podatkovne objekte. Funkcijo, ki je definirana znotraj Scilaba, lahko uporabljamo tudi kot vhodni ali izhodni argument druge funkcije. Podpira znakovne nize podatkovnega tipa, kar v posebnih pogojih omogoča sprotno tvorjenje funkcij.

Nekatere izmed možnosti Scilab so še: 2D in 3D grafika, animacija; polinomi, racionalne funkcije; linearna algebra, razpršene matrike; simulacija; statistika in mnoga druga področja. Deluje na številnih operacijskih sistemih (Unix, Linux, Windows ipd.). Vsebuje izvorno kodo, on-line pomoč in angleški uporabniški vodnik.



Slika 15.1: Uradna spletna stran Scilab

Prenos in namestitvev

Programsko orodje Scilab namestimo na različne načine, odvisno od operacijskega sistema, ki ga uporabljamo. V okolju Windows programsko orodje namestimo iz uradne spletne strani Scilab [46] in ga nato brezplačno uporabljamo na lastnem računalniku, pri čemer moramo upoštevati avtorske pravice.

V operacijskem okolju Ubuntu programsko orodje namestimo po dveh različnih poteh, ki smo jih predhodno že opisali. Najenostavneje to naredimo z uporabo Synaptic orodja, kjer programsko orodje enostavno poiščemo, prenesemo in namestimo. Nameščen program nato odpremo v menijski vrstici v razdelku *Programiranje* (glej Sliko 15.2).

Program zaženemo z dvoklikom na ikono programa, pri čemer se odpreta dve okni. V konzolnem oknu (v nadaljevanju osnovno okno) se izvaja pomožni program Scilab, ki mora biti ves čas dostopen. V drugem oknu se prikaže vnosna vrstica, pred katero je zapisan simbol → (glej Sliko 15.3).

Osnove za delo s programskim orodjem so predstavljene v *Introduction* v razdelku *Scilab Demonstration*. Ob zagonu programskega orodja se prikaže osnovno okno z zapisom menijske in orodne vrstice (glej Sliko 15.4).

Menijska vrstica je sestavljena iz 6 razdelkov. Razdelek *Datoteka (File)* omogoča zagon skriptne datoteke (Execute), odpretje že shranjene datoteke (*Open a file*), nalozitev shranjene spremenljivke (*Load environment*), shranitev



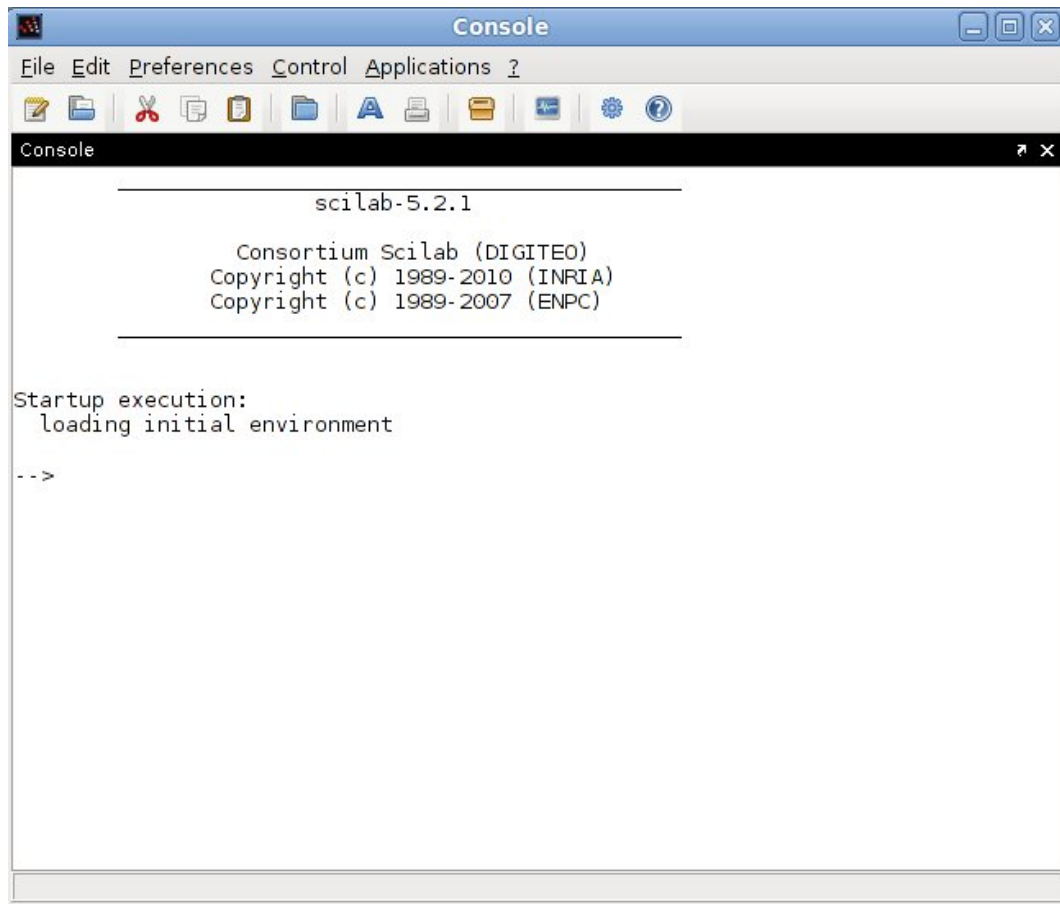
Slika 15.2: Odpiranje programskega orodja v Ubuntu

spremenljivke (*Save environment*), spremembo delovne mape (*Change current directory*), izpis delovne mape (*Display current directory*), tiskanje zapisanega programa (*Print*) in izhod iz programa (*Quit*). Razdelek *Uredi (Edit)* omogoča urejanje standardnih možnosti: rezanje (*Cut*), kopiranje (*Copy*), lepljenje podatkov iz drugih dokumentov (*Paste*) ipd. Razdelek *Preferences* ponuja oblikovne možnosti - poljubno lahko nastavimo barvo ozadja (*Colors*), velikost in vrsto pisave (*Font*). V primeru, ko imamo v okencu določene podatke, ki jih ne potrebujemo več, jih enostavno pobrišemo s *Clear History* (glej Sliko 15.5).

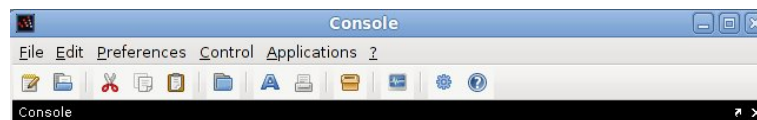
Razdelek *Kontrola (Control)* vsebuje določena kontrolna orodja in ponuja možnosti, kot so *Resume* (omogoča izvajanje programa po premoru), *Abort* (preneha z izvajanjem trenutnega programa) in *Interrupt* (prekine izvajanje trenutnega programa). Razdelek *Aplikacija (Application)* je pomemben predvsem zaradi možnosti *Editor*, s katero odpremo novo okno za pisanje programa, ki ga nato shranimo in izvozimo v osnovno programsko okno (glej Sliko 15.6).

Slika 15.7 prikazuje okno, ki je namenjeno zapisu programa, za izvoz v osnovno okno.

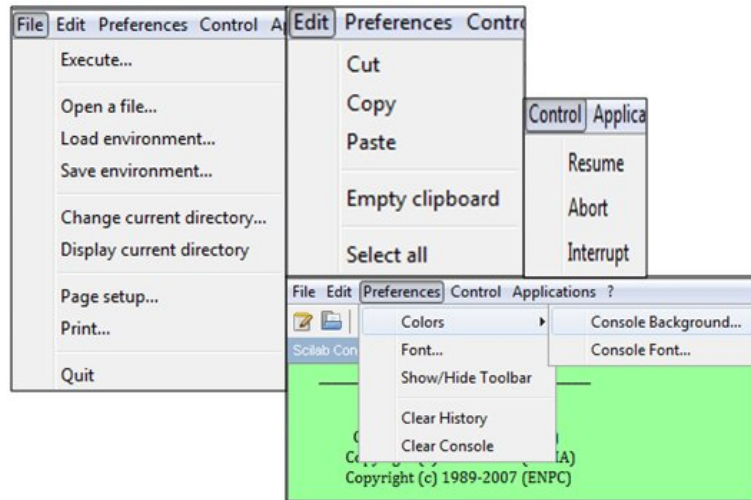
Razdelek *Xcos* uporabljamo za risanje blokovnih shem. Na voljo je večje število blokovnih shem ter prazno okno, namenjeno oblikovanju le teh (glej Sliko 15.8).



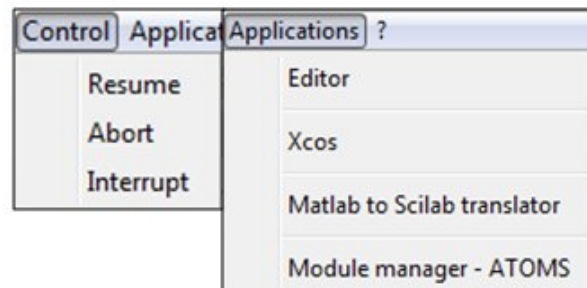
Slika 15.3: Zagon programa



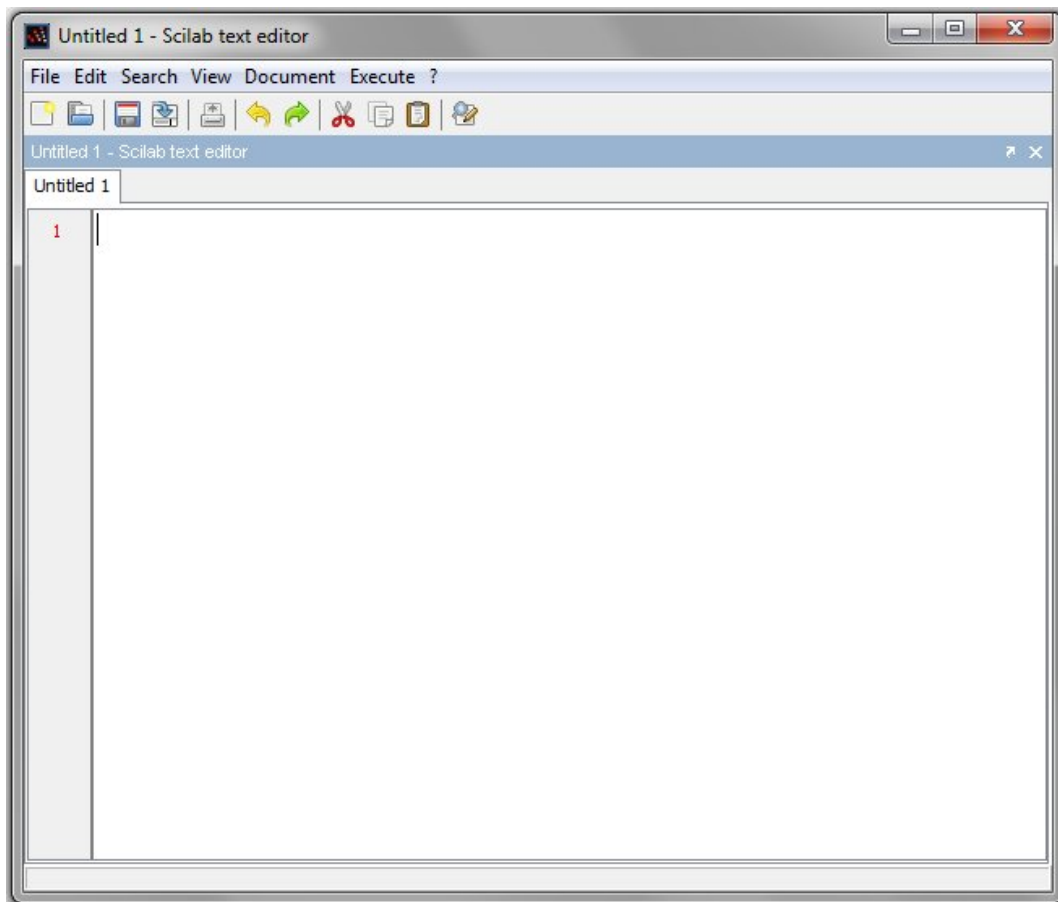
Slika 15.4: Menijska in orodna vrstica v Scilabu



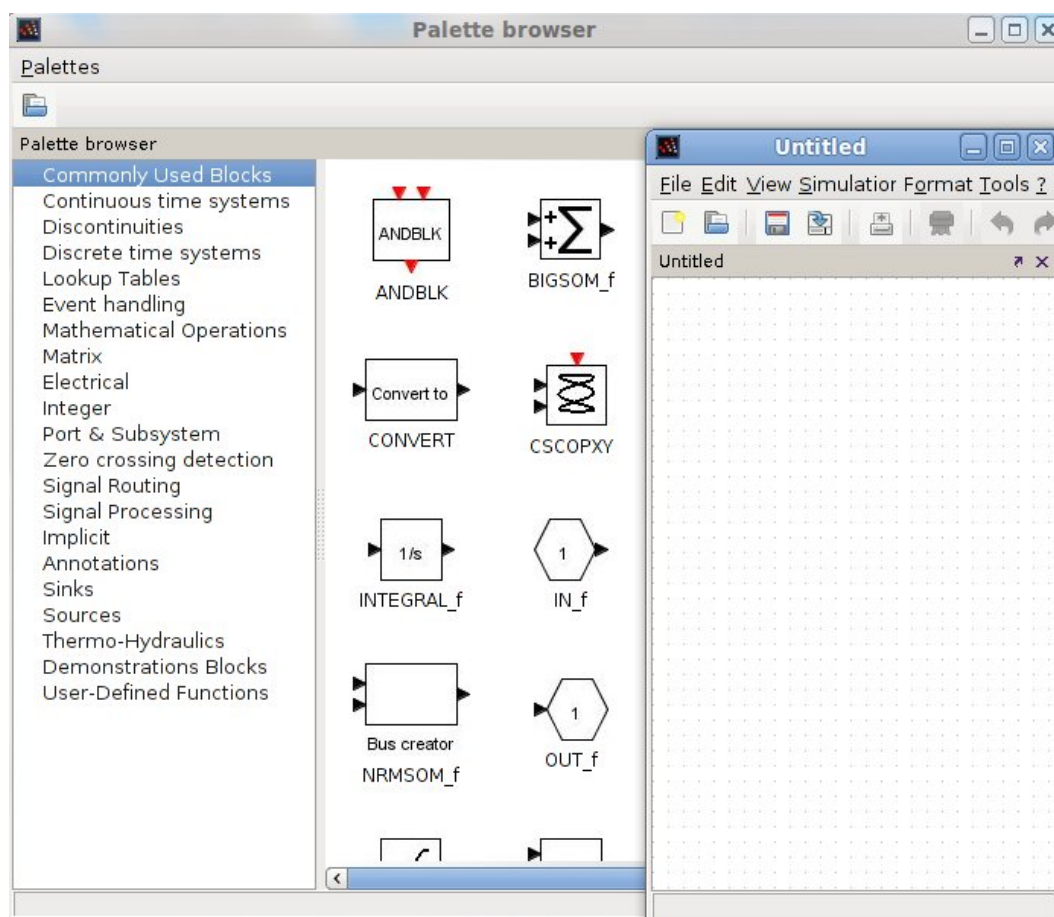
Slika 15.5: Menijska vrstica – 1. del



Slika 15.6: Menijska vrstica – 2. del



Slika 15.7: Okno za zapis programa



Slika 15.8: Razdelek Xcos

Ena izmed značilnih vrst blokov so bloki linearnih gradnikov, ki predstavljajo gradnike za opis linearnih dinamičnih sistemov. Nekateri izmed pomembnejših blokov linearnih gradnikov:

- *Sum* (seštevalnik) - blok, ki na izhodu da vsoto vseh vhodov. Bloku lahko določimo poljubno število vhodov, njihove uteži in njihov predznak. Poleg osnovnega bloka pravokotne oblike obstaja tudi inačica okrogle oblike. Pri slednji ne moremo spreminjati predznaka in uteži vhodov.
- *Integrator* - izhod tega gradnika je integral vhoda.
- *Numerator/Denominator* ($num(s)/den(s)$) - blok za predstavitev prenosne funkcije.
- *Gain* (ojačevalnik) uporabimo kadar moramo funkcijo množiti s konstanto. Na izhodu tako dobimo vhod z ojačenjem, ki ga določimo.

Uporabniki lahko v programskem orodju uporabljamo razdelek *Pomoč (?)*, ki omogoča pogled v opis posameznih blokov in ostalih funkcij s podrobnejšim opisom. Razdelek *?* omogoča pomoč pri uporabi programa (*Scilab Help*). *Scilab Demonstration* nudi možnost ogleda že izdelanih primerov programa. Dostopamo lahko tudi do njihove spletne strani (*Web links*), (*About Scilab*) pove več o programu. Preverimo lahko ali smo pravilno postavili blok, je potrebna kakršnakoli sprememba ipd. (glej Sliko 15.9).

Slika 15.10 prikazuje menijsko vrstico z vsemi vsebovanimi razdelki.

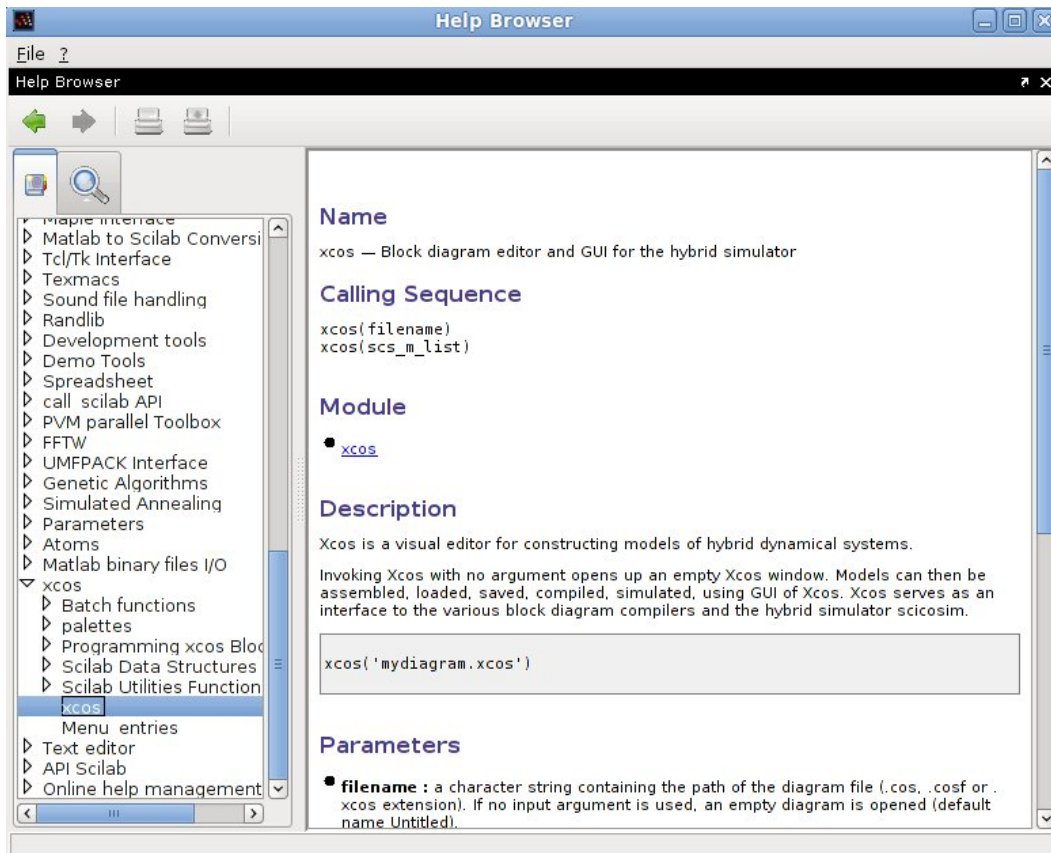
Scilab Demonstrations prikazuje nekaj izmed že zapisanih programov z rešitvami. Le-te lahko preoblikujemo glede na izbran problem in jih smiselno uporabimo (glej Sliko 15.11).

Orodno vrstico sestavlja 12 ikon. Glej Tabelo 15.1 in Sliko 15.12.

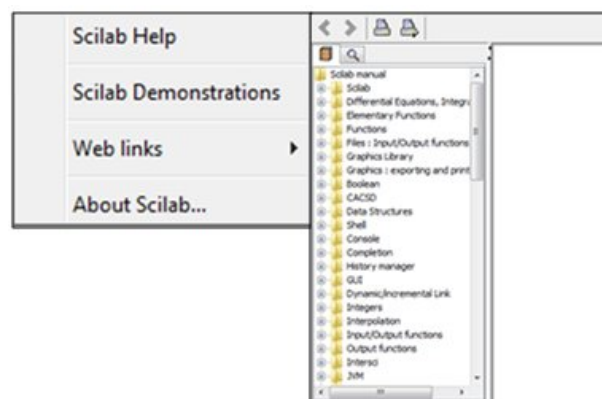
Scilab prepozna veliko podatkovnih tipov. Skalarni objekti so konstante, polinomi, znakovni nizi in racionalna števila.

Spremenljivke in osnovne računske operacije

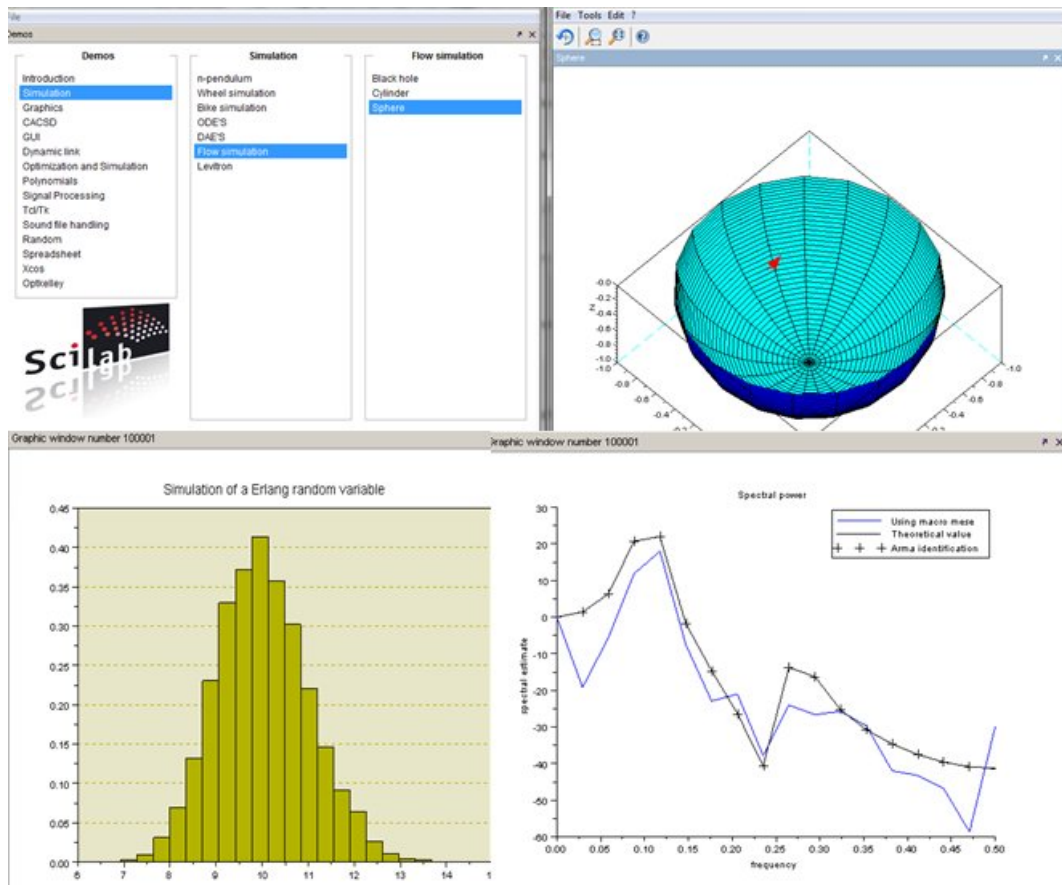
Programsko orodje je občutljivo na velike in male črke - b in B sta dve različni spremenljivki. V kolikor želimo izpis neke spremenljivke v matrični obliki, vpišemo ime pod katerim je shranjena. Na začetku spremenljivkama b in B določimo vrednosti ter za vsakim zapisom navedemo znak ;, ki pomeni neizpis rezultata. Če znaka ni, se rezultati izpišejo. S simbolom // označimo komentarje.



Slika 15.9: Razdelek Pomoč



Slika 15.10: Menijska vrstica – 3. del



Slika 15.11: Demo programi



Slika 15.12: Orodna vrstica

Ikone orodne vrstice	
1	Odpiranje okenca za zapis programa
2	Shranjevanje dokumenta
3, 4, 5	Orodja za rezanje, kopiranje in lepljenje dokumentov
6	Spreminjanje delovne mape
7	Spreminjanje oblike in velikosti pisave
8	Tiskanje dokumenta
9	Namestitev posameznih modulov
10	Xcos za risanje blokovnih shem
11	Demo programi
12	Pomoč

Tabela 15.1: Ikone orodne vrstice

Matrike

Za vpisovanje matrik ali vektorjev upoštevamo nekaj osnovnih pravil. V vrstico matrike ali vektorja vpišemo število z vmesnim presledkom ali vejico. V novo vrstico matrike ali vektorja se pomaknemo s podpičjem ter celoten zapis matrike obdelamo z oglatimi oklepaji. Ko definiramo matriko, jo Scilab shrani pod oznako, s katero smo jo poimenovali in jo z vpisom oznake kadarkoli prikličemo - v danem primeru velika tiskana črka D.

Zapis spremenljivk in matrik

```
-->a=1
a =
  1.

-->A=2
A =
  2.

-->a+A
ans =
  3.

-->D=[16 3 2 1
-->5 10 11 8
-->9 6 7 12
-->4 15 15 1]
```

D =

16.	3.	2.	1.
5.	10.	11.	8.
9.	6.	7.	12.
4.	15.	15.	1.

Funkcije

Scilab ima veliko število vgrajenih funkcij: *srqt*, *sin*, *exp* ipd. Nekatere izmed vgrajenih funkcij so osnovne (*sum*, *prod*, *sqrt*, *diag*, *cos*, *max*, *round*, *sign*, *fft*), urejevalne (*sort*, *gsort*, *find*), za posebne matrike (*zeros*, *eye*, *ones*, *matrix*, *empty*), linearna algebra (*det*, *inv*, *gr* . . .), nizi znakov (*string*, *part*, *evstr* . . .), diferencialne enačbe (*ode*, *dassl*, *dassrt*, *odedc*) itd.

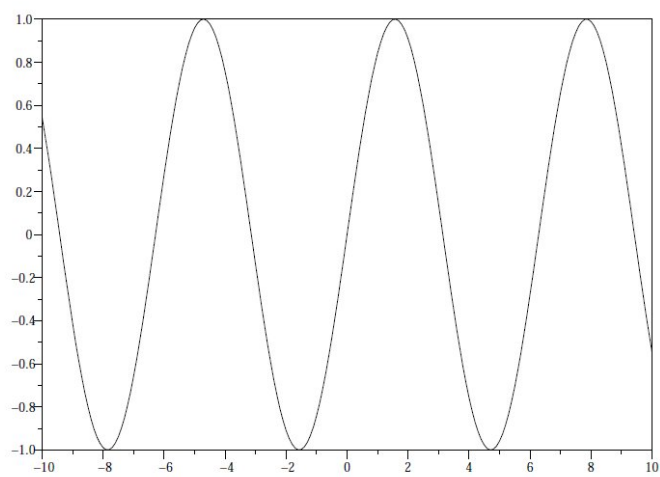
Grafi

S Scilab izrišemo številne grafe matrik, vektorjev in drugih funkcij v 2D in 3D, na različne načine in v obliki animiranega filma. Odprtih imamo lahko več grafičnih oken hkrati, vendar je v vsakem trenutku lahko aktivno le eno okno 15.13.

Programiranje v Scilab

Pogosto se v praksi srečujemo s problemi, ki jih ne moremo razrešiti na enostaven način, saj je predhodno potrebno zapisati določen računalniški program oz. model. Za izdelavo modela v programskem orodju Scilab je potrebna pripravljenost za raziskovanje. S pomočjo uporabe dodatnih priročnikov danega programa je programiranje veliko enostavnejše.


```
-->y=sin(x);  
-->plot2d(x,y)
```



Slika 15.13: Graf funkcije narisana v Scilab

Leto	Nakup novih osebnih vozil	Leto	Nakup novih osebnih vozil
1999	77.217	2005	60.531
2000	55.120	2006	59.885
2001	52.316	2007	67.791
2002	50.859	2008	71.037
2003	59.881	2009	57.391
2004	62.697		

Tabela 15.2: Statistični podatki prodaje v obdobju 1999-2009

Problem

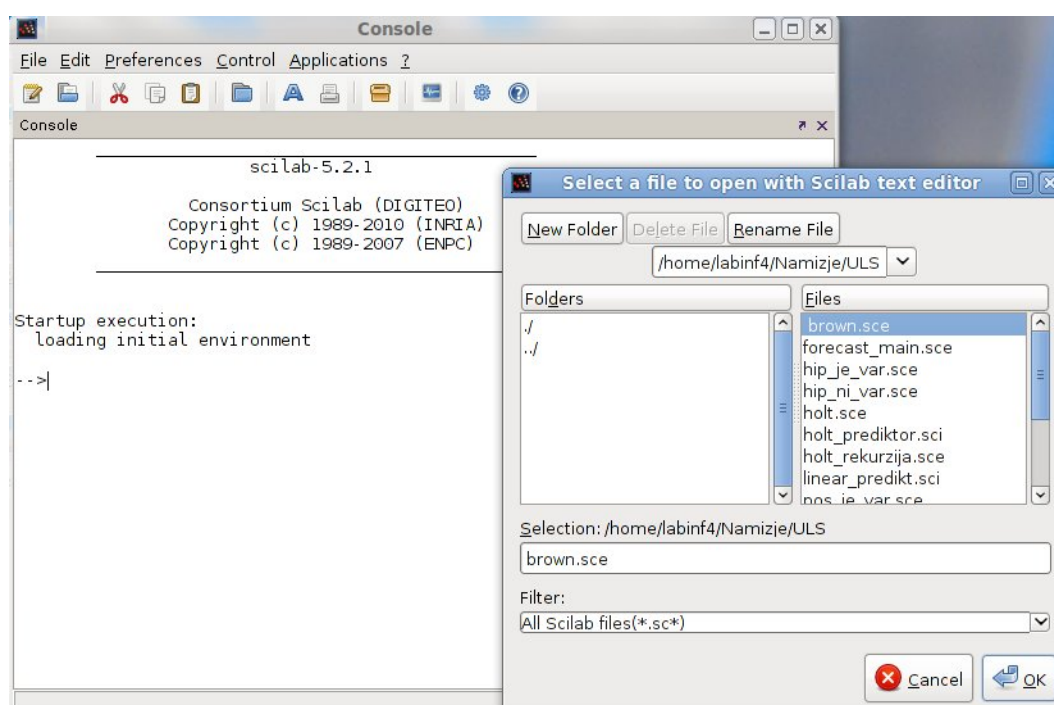
Podjetje OpenStorage se ukvarja s skladiščenjem treh vrst komponent (pnevmatik, platišč in vijakov), namenjenih za izdelavo osebnega avtomobila. Na podlagi zbranih podatkov o nakupu v preteklih 11 letih izvedemo napoved prodaje avtomobilov za obdobje prihodnjih 5 let. Ob zapisu danega programa in dobljenih rezultatih bomo preverimo ali je izbrana metoda pravilna oz. ali jo je za nadaljnjo delo potrebno spremeniti. Statistični podatki prodaje v obdobju 1999-2009 so podane v Tabeli 15.2.

Problem poskušamo rešiti na podlagi izdelanega modela linearne regresije, Holt in Brown metode. Dani program, že zapisan v programskem orodju Scilab pridobimo iz učnega gradiva dr. Dejana Dragana [77], ki program uporablja, kot študijsko gradivo na fakulteti.

15.3 Uporaba

Za razrešitev danega problema je v prvi fazi potreben zapis modela v Scilab. Prikažemo problem in njegovo rešitev na že izdelanih programih z Holt metodo (*holt.sce*), Brown metodo (*brown.sce*) in linearno regresijo (*regresija.sce*). Program je prikazan v Prilogi [77]. Matematičen model lahko zapišemo tudi sami, vendar je potrebno vložiti precej časa in truda. Na takšen način izvedemo marsikatero optimizacijo in napovedovanje. Zapisani programi delujejo znotraj skupnega programa imenovanega *forecast_main*, ki ga dopolnimo z izbranimi podatki. V Scilab odpremo osnovno okno in v razdelku *File* izberemo možnost *Open a file*. Izberemo program npr. z Brown metodo in ga s klikom na razdelek *Open* odpremo v novem oknu (glej Sliko 15.14).

Ko je program odprt, ga je potrebno izvoziti v osnovno okno, kar storimo



Slika 15.14: Odpiranje datoteke

tako, da v meniju *Execute* izberemo možnost *Excute File Into Scilab*. Postopek izvedemo še za vse ostale tri datoteke.

Zapis programa

```
//
// brown.m
//
//

function brown(d,alfa)

fakt = 5;

N = length(d);

t = [1:1:N];

// izhodi prediktor-filtra (projekcija prihodnosti za en korak naprej -
// napoved kolicine za cas t+1 ob casu t):
// (relevantno je k = 2:N)

p = [];
p(1) = d(1)
p(2) = d(1)

for k=3:N
    p(k) = alfa*d(k-1) + (1-alfa)*p(k-1)
end;
.....
```

Pozorni moramo biti predvsem pri datoteki *forecast_main*, kjer v program vstavimo podatke. V izbranem primeru vstavimo podatke namišljenega podjetja, ki se ukvarja s skladiščenjem komponent za izdelavo avtomobilov. Podatki prikazujejo število prodanih vozil v letih od 1999 do 2009. Ko izvozimo vse datoteke v osnovno okno zapišemo pogoj *forecast_main* ter pritisnemo *Enter*.

Klicanje programa

```
-->exec('C:\Users\Lab4\Desktop\ULS\brown.sce', -1)

-->exec('C:\Users\Lab4\Desktop\ULS\holt.sce', -1)
```

```
-->exec('C:\Users\Lab4\Desktop\ULS\regresija.sce', -1)
-->exec('C:\Users\Lab4\Desktop\ULS\forecast_main.sce', -1)
-->forecast_main
```

Program vpraša kako naj izvede povpraševanje: ročno (vpis naših podatkov), po "defaultu" (sam izbere vrednosti). Odločimo se za ročno povpraševanje. Ko program vpraša po podatkih, zapišemo podatke, ki so na voljo (prodaja novih avtomobilov od leta 1999 do 2009). Program vpraša še po metodi po kateri se bo izvedlo povpraševanje. Izberemo regresijsko metodo (lahko pa se odločimo za katero izmed preostalih) in določimo za koliko let izvajamo predikcijo. V danem primeru izberemo vrednost 5, ki pomeni predikcijo za naslednjih 5 let.

```
Povprasevanje rocno (1), default(2), nakljucno (3)1
povprasevanje d= ? npr. [7 9 12 14 12 30 8 12 14 15 25 40][77217 55120 52316
50859 59881 62697 60531 59885 67791 71037 57391]
regresija, mnk(1)/brown(2)/holt(3)1
Za koliko let predikcija =5
```

Ob vnosu izbranih podatkov za prikaz rezultatov, pritisnemo tipko *Enter* (glej Sliko 15.15).

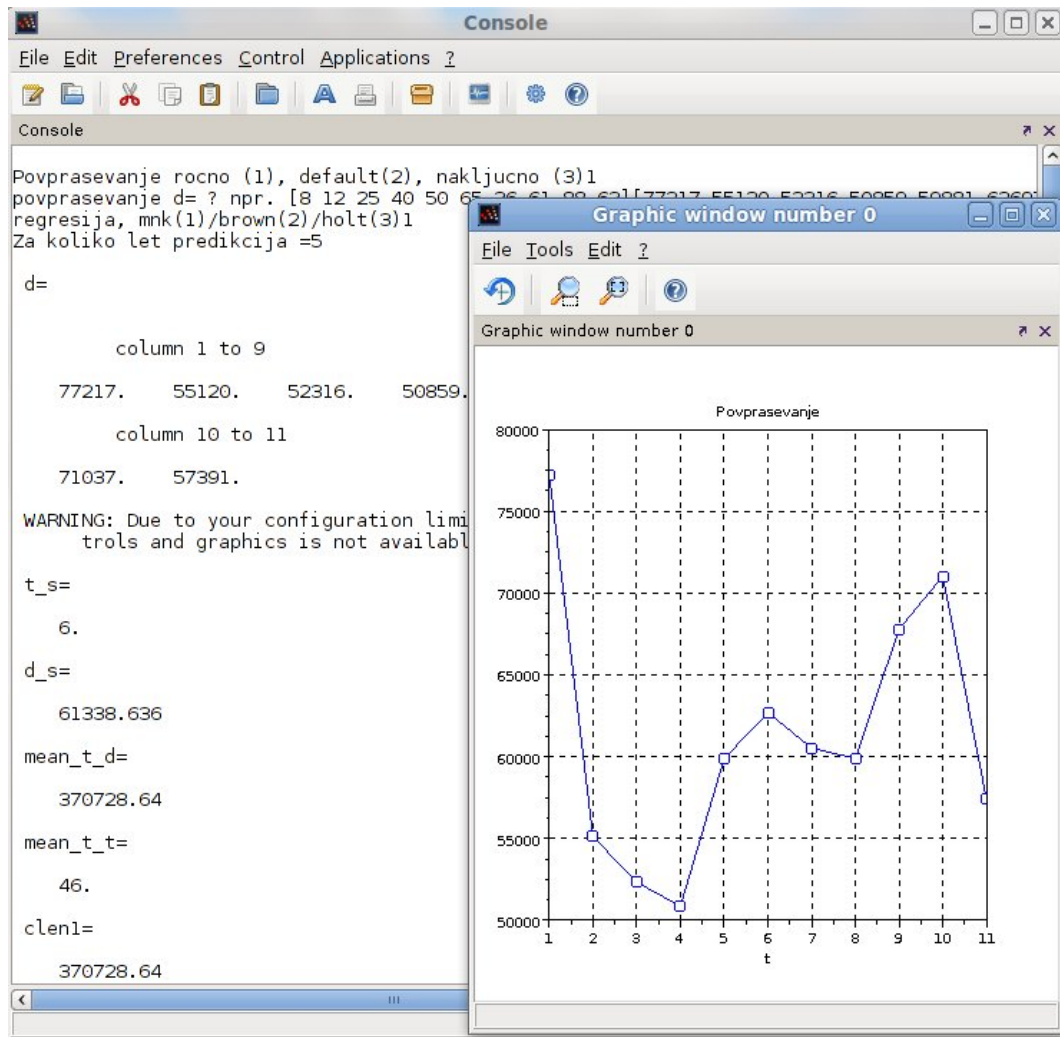
Program vpraša ali nadaljuje z izpisom rezultatov, za potrditev pritisnemo na tipko *Y*.

```
d_s=
61338.636

mean_t_d=
370728.64

mean_t_t=
46.

c1en1=      370728.64
c1en2=      368031.82
```



Slika 15.15: Izpis rezultatov

Razlaga rezultatov:

d - število podatkov = 77217 55120 52316 50859 59881 62697 60531 59885 67791
71037 57391

t_s - povprečen čas = 6

$$\bar{t} = \frac{1}{N} \sum t$$

d_s - povprečna vrednost = 61338,636

$$\bar{d} = \frac{1}{N} \sum d(t)$$

mean_t_d = 370728,64 - če bi računali analitično je to vrednost

$$\frac{1}{N} \sum t^2 \cdot d(t)$$

mean_t_t = 46 - če bi računali analitično je to vrednost

$$\frac{1}{N} \sum t^2$$

clen1 = 370728,64

clen2 = 368031,82 - če bi računali analitično je to vrednost

$$\bar{t} \cdot \bar{d}$$

clen3 = mean_t_t = 46

clen4 = 36 - če bi računali analitično je to vrednost

$$\bar{t}^2$$

a = 269,68182 - če bi računali analitično je to vrednost parametra

$$\hat{a} = \frac{\frac{1}{N} \sum t \cdot d(t) - \bar{t} \cdot \bar{d}}{\frac{1}{N} \sum t^2 - \bar{t}^2}$$

b = 59729,545 - če bi računali analitično je to vrednost parametra

$$\hat{b} = \bar{d} - \hat{a} \cdot \bar{t}$$

d_o - ocena modela za prvih 11 let = 59990,27 60259,909 60529,591... 61878

e - izračunani pogreški za prvih 1 let = 17226,773 -5139,9091 -8213,5909...
-5296,0455

e_sr = 1,984D-12

VAR - varianca = 63203478

$$VAR(e) = \frac{1}{N-1} \sum [e(t) - \bar{e}]^2$$

stde - standardni odklon = 7950,0615

$$STD(e) = \sqrt{VAR(e)}$$

d_o_pred - napoved povpraševanja za naslednjih 5 let = 62956,727 63226,
409... 64035,455

d_o_pred_zg - največje predvideno povpraševanje = 78856,85 79126,532...
79935,578 (če upoštevamo 2-kratno standardno deviacijo)

d_o_pred_sp - najmanjše predvideno povpraševanje = 47056,604
47326,286...
48135,332 (če upoštevamo 2-kratno standardno deviacijo)

Rezultati (Scilab izpis)

```

clen3=
  46.
clen4=
  36.
a=
  269.68182
b=
  59720.545

d_o=

```

```

column 1 to 4
  59990.227    60259.909    60529.591    60799.273
column 5 to 8
  61068.955    61338.636    61608.318    61878.
column 9 to 11
  62147.682    62417.364    62687.045

e=
column 1 to 4
  17226.773 - 5139.9091 - 8213.5909 - 9940.2727
column 5 to 9
  - 1187.9545  1358.3636 - 1077.3182 - 1993.    5643.3182
column 10 to 11
  8619.6364 - 5296.0455

e_sr=
  1.984D-12

VAR=
  63203478.

stde=
  7950.0615

d_o_pred=

column 1 to 4
  62956.727    63226.409    63496.091    63765.773
column 5
  64035.455

d_o_pred_zg=

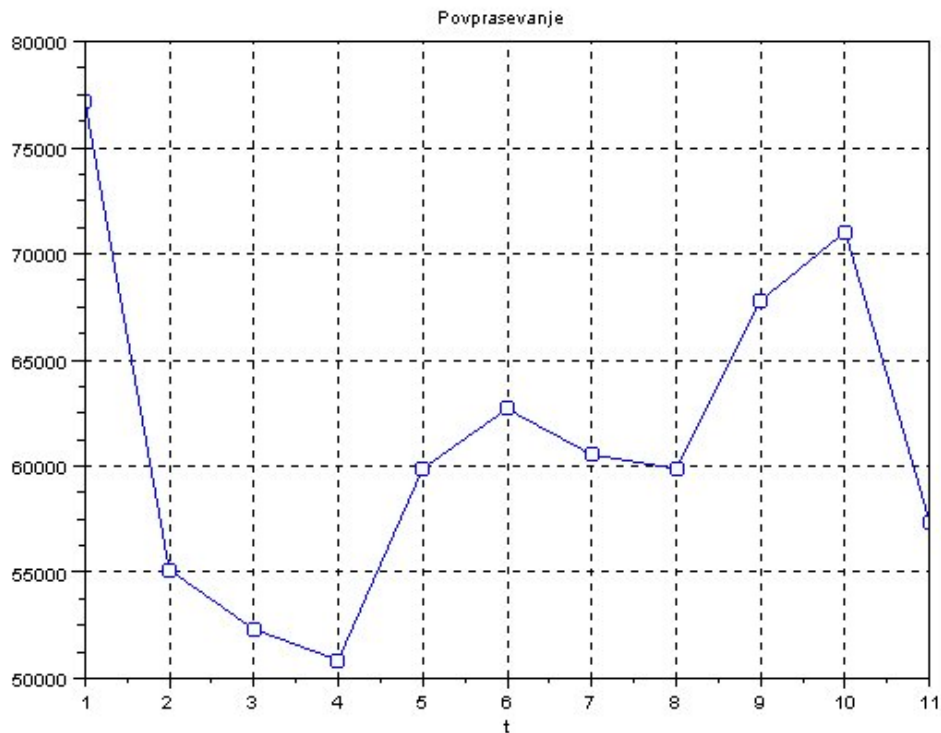
column 1 to 4
  78856.85    79126.532    79396.214    79665.896
column 5
  79935.578

d_o_pred_sp=

column 1 to 4
  47056.604    47326.286    47595.968    47865.65
column 5
  48135.332

```

Izračun pogreška podaja vrednosti odstopanja od linije regresijske premice, uporabimo pa tudi izračun standardne deviacije (STD), s pomočjo katere



Slika 15.16: Povpraševanje za sedanje obdobje

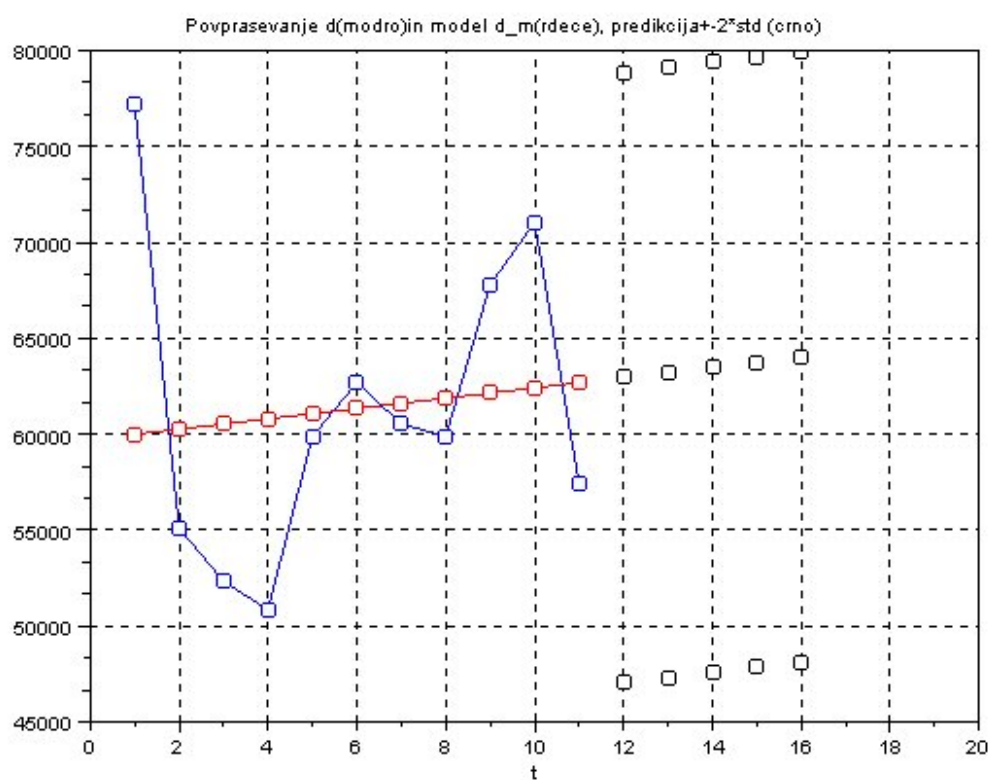
izračunamo povprečni odklon tako, da dobimo minimalno in maksimalno vrednost napovedi posameznega leta.

Program izriše dva grafa. Slika 15.16 prikazuje graf povpraševanja za sedanje obdobje, na podlagi podatkov iz preteklih obdobj.

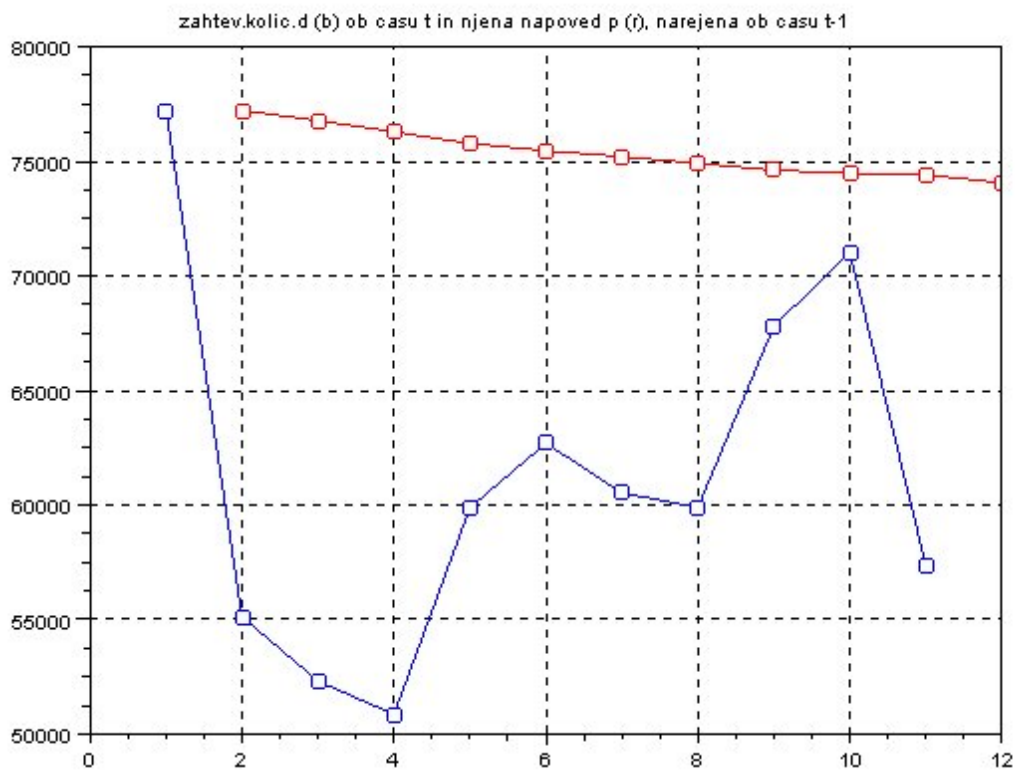
Sledi graf, ki prikazuje povpraševanje (modra barva), oceno modela d_m (rdeča barva) ter predikcijo z upoštevanjem 2-kratnega standardnega odklona (črna barva), za naslednjih 5 let (glej Sliko 15.17).

Iz grafa je razvidno, da se ocena modela razlikuje od realnih podatkov, zato kot primer prikazemo še rezultate, ki jih dobimo z uporabo Holt metode. Rezultati te metode prikazujejo še večje odstopanje, kar pomeni, da je za dani primer metoda še manj primerna kot linearna regresija (glej Sliko 15.18).

S Scilab prikazemo model napovedovanja povpraševanja nakupa novih vozil za naslednjih 5 let, pri čemer upoštevamo podatke preteklih 11 let. Iz rezultatov je razvidno, da je model linearne regresije primernejši od Holt metode, čeprav menimo, da nihče izmed njiju ni optimalen. Za reševanje kompleksnejših



Slika 15.17: Povpraševanje, model in predikcija z upoštevanjem standardnega odklona – regresija



Slika 15.18: Povpraševanje, model in predikcija z upoštevanjem standardnega odklona - Holt

problemov je vsekakor potrebno poglobljeno reševanje problema s pomočjo različnih priročnikov, navodil in dodatne literature. Preučiti in zapisati je potrebno model za nelinearne modele značilne v praksi. Potrebno je zapisati pogoje, ki bodo to tudi izpolnjevali.

Povzetek

Scilab je programsko orodje za računanje numeričnih funkcij. Uporabljamo ga za računanje z različnimi oblikami matematičnih zapisov in struktur. Z danim programom lahko analiziramo linearne in nelinearne dinamične sisteme. Prav tako programsko orodje uporabimo kot pripomoček za numerično optimizacijo, kot sta linearno in nelinearno programiranje.

S pomočjo izdelanega modela in zbranih statističnih podatkov preteklih 11 let, prikažemo model napovedovanja prodaje vozil prihodnjih 5 let. Pridobimo že izdelan program linearnega napovedovanja. Mnogokrat je za razrešitev kompleksnejših problemov potrebno izdelati samostojen program, vendar je samo programiranje zahtevno in vključuje bistveno več časa in truda.

S to vrstnimi programi si lahko pomagamo pri reševanju številnih problemov, tako v praksi, kakor pri študiju. V praksi je poznan plačljiv program Matlab, ki vsebuje podobne funkcije kakor Scilab. Za namene študija je Scilab dovolj zanesljiv, da se naučimo osnove programiranja in reševanja določenih logističnih problemov.

Pri opisu programskega orodja Scilab smo uporabili še dodatne vire in literaturo: [105] [82] [10] [98] [81] [47] [48] [106] [78].

Poglavje 16

OOo PREGLEDNICA - gibanje zalog



Elektronska preglednica
Preglednica gibanja zalog
Primer: izračun zasedenosti zalog in stroškov

16.1 Teoretično ozadje

16.1.1 Elektronske preglednice

Kadar govorimo o elektronskih preglednicah, v prvi meri mislimo na prikaz podatkov, tako v obliki tabele kot tudi v grafični obliki. Tabela na prvi pogled ni nič drugačna od tabel, narisanih v urejevalnikih besedil, kamor vnesemo podatke in jo poljubno oblikujemo. Popravljjanje tabel v urejevalniku besedil je zamudno delo, saj je potrebno vsako stvar popraviti ročno - tu se pokaže prednost t.i. elektronskih preglednic. Ne le prikazovanje podatkov, ampak tudi sortiranje in filtriranje, kar pomeni, da prikažemo zgolj tiste podatke, ki izpolnjujejo določene pogoje. Med številskimi podatki izvajamo računske operacije - od najbolj osnovnih, kot so seštevanje, odštevanje, množenje, deljenje, do izredno zapletenih izračunov po formulah. Če spremenimo vhodne podatke, se samodejno spremenijo tudi izračunani podatki, kar prihrani veliko časa. V navadni tabeli moramo vse vrednosti izračunati ročno in jih ponovno vnesti v tabelo.

Elektronske preglednice uporabljamo za tri poglobitve namene: delo s raznoraznimi sezname (npr. seznam ljudi, storitev, predmetov, izmerjenih vrednosti, razni imeniki, naslovi, datumi, ceniki in podobno); manipuliranje, računanje s številskimi podatki, prikaz končnih vrednosti v tabeli, analiziranje podatkov in rezultatov; prikaz z grafi in diagrami.

16.2 O programskem orodju

OOo Preglednica (v nadaljevanju Preglednica) je programsko orodje, namenjeno urejanju različnih podatkov. Začetniki ga spoznavajo kot intuitivnega in enostavnega za učenje. Vsi, ki se profesionalno ukvarjajo z oblikovanjem podatkov najdejo številni nabor naprednih funkcij. Na voljo so formule za izračun zneska skupne prodaje, za izračun dobička, za množico statističnih spremljanj, za izračun sedanje ali prihodnje vrednosti prodaje ipd. Omogoča izdelavo raznovrstnih grafikonov, ki jih pozneje uporabimo v drugem pisarniškem paketu OOo, kot je *Writer* ali *Impress*. Preglednica omogoča povezavo na katero izmed drugih zbirk podatkov, ki prav tako temeljijo na odprti kodi (npr. MySQL). Preglednice lahko shranimo v zapisu OpenDocument (odf.) format - novi mednarodni standard za pisarniške dokumente. Zapis temelji na XML, kar pomeni, da nismo omejeni zgolj na izbrano programsko orodje, saj lahko svoje preglednice uporabljamo v vsakem programskem orodju, ki podpira standard OpenDocument. Uporabimo lahko tudi preglednice shranjene v *Microsoft Excel*, ki jih enostavno prenesemo v OOo oz. jih že



Slika 16.1: Prenos programskega paketa OOo

prej shranimo kot OpenDocument. Če želimo zgolj vpogled v naše rezultate, uporabimo Portable Document Format (.pdf).

Prenos in namestitvev

Preglednica je del programskega paketa OOo, ki med drugimi ponuja delo z dokumenti z besedilom, risanjem, zbiranjem podatkov, formulami itd. Programski paket OOo prenesemo s spletne strani OOo [38] in klikom na *Prenesite OOo 3.2.1* (Slika 16.1).

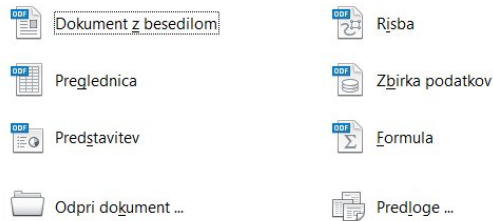
Ko iz spletnega mesta namestimo OOo se prikažejo možnosti izbire med posameznimi programi. Izberemo program *Preglednica* (glej Sliko 16.2).

Odpre se nova preglednica (Slika 16.3), ki jo programsko orodje poimenuje *Neimenovana 1*, mi pa jo lahko ustrezno preimenujemo. Preglednico lahko zaženemo tudi iz menija *Start* → *OpenOffice*. Na prvi pogled je preglednica podobna Microsoft Excel.

Vrstice

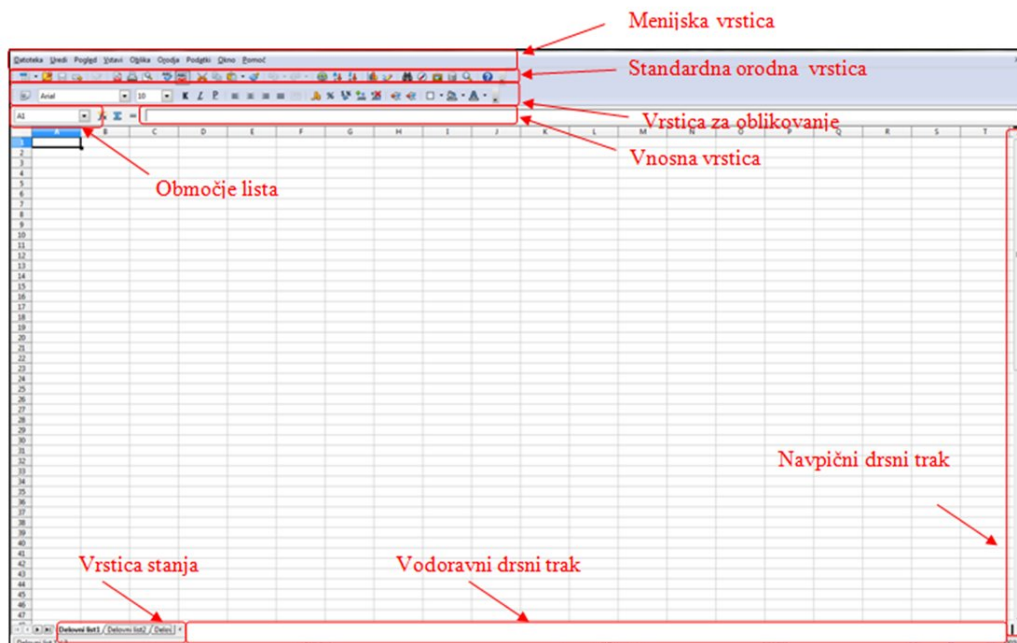
Naslovna vrstica se nahaja najvišje v oknu OOo Preglednica (*Calc*). Prikaže ime odprte datoteke, ki jo poljubno spreminjamo (glej Sliko 16.4).

V *menijski vrstici* so meniji, iz katerih izbiramo ukaze, ki jih v Preglednici potrebujemo za delo (glej Sliko 16.5).



ORACLE®

Slika 16.2: Naslovna stran OOo



Slika 16.3: Osnovno okno



Slika 16.4: Naslovna vrstica

A horizontal menu bar with a light gray background. It contains nine menu items, each with a small icon to its left and a letter underlined for keyboard navigation: 'Datoteka' (D), 'Uredi' (U), 'Pogled' (P), 'Vstavi' (V), 'Oblika' (O), 'Orodja' (O), 'Podatki' (P), 'Okno' (O), and 'Pomoč' (P).

Slika 16.5: Menijska vrstica

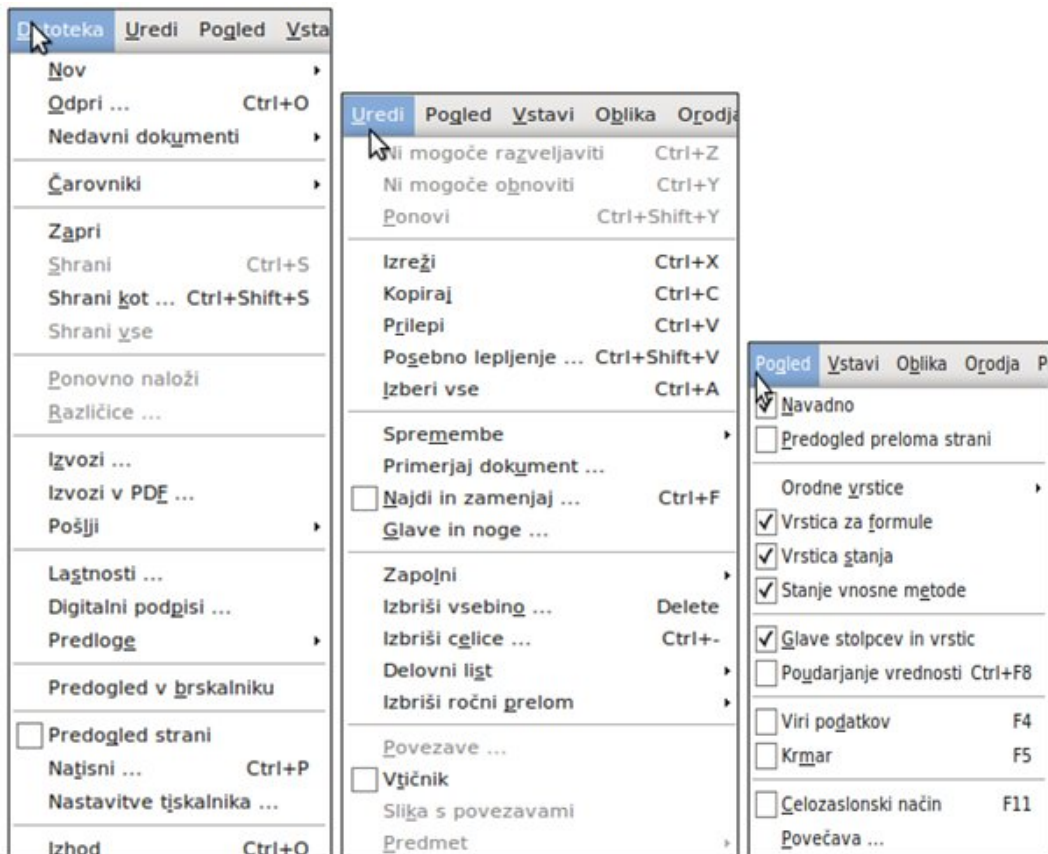
V meniju *Datoteka* so ukazi, ki vplivajo na celoten dokument. Izbiramo med možnostmi izdelave novega dokumenta, zapiranjem ali odpiranjem obstoječega dokumenta, vnosom lastnosti dokumenta, shranjevanjem, izvozom ali pošiljanjem dokumenta, tiskanjem dokumenta ipd. V meniju *Uredi* se nahajajo ukazi, ki jih uporabljamo za preklic zadnjega dejanja, kopiranje in lepljenje iz odložišča, brisanje vsebine celic in delovnih listov, oblikovanje glave in noge ipd. Meni *Pogled* vsebuje ukaze za upravljanje videza okna (glej Sliko 16.6).

V meniju *Vstavi* je omogočeno vstavljanje posameznih vrstic, stolpcev, imen celic in listov v preglednico, kakor tudi seznam kategorij in funkcij. Vstavimo lahko tudi posamezne slike iz že shranjenih datotek ali spleta, grafikone in ostale predmete. V meniju *Oblika* so možnosti za oblikovanje elementov, ki so trenutno v dokumentu. Meni *Orodja* vsebuje možnosti preverjanja besedila v celicah, določitve jezika, spoja dokumenta, odkrivanje napak ipd. Od tu lahko prekličemo makre, posodobimo videz orodjarn, menijev, tipkovnic in osnovnih nastavitev programa (glej Sliko 16.7).

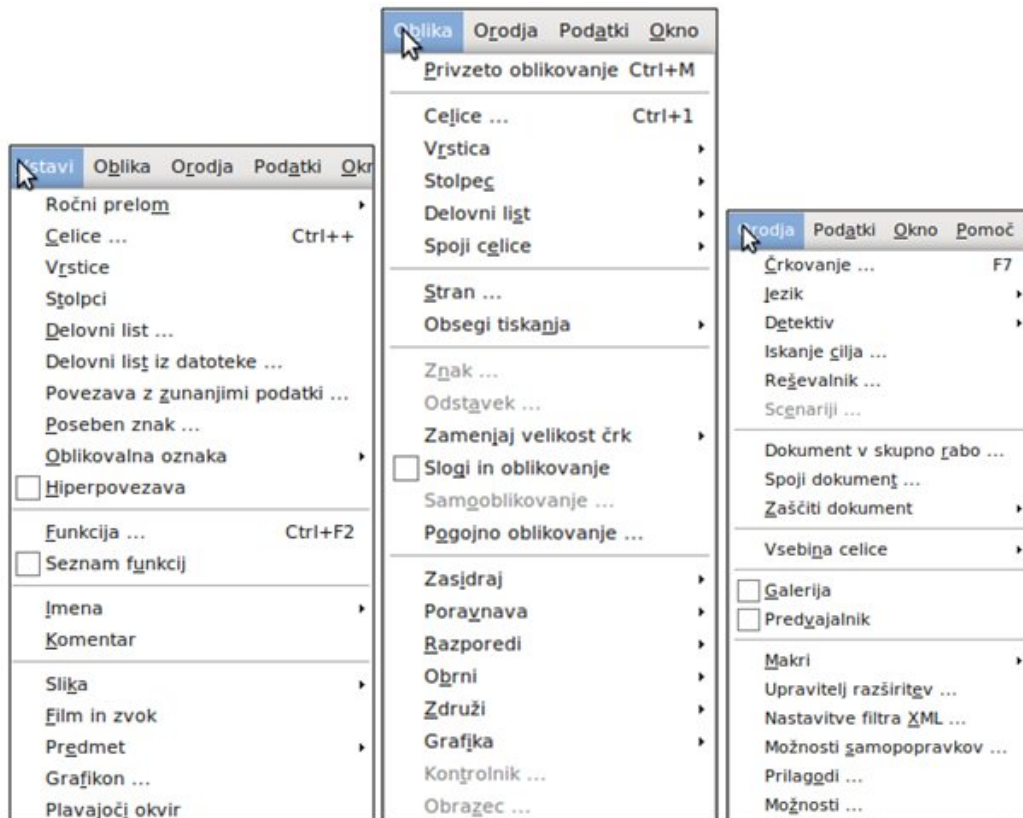
Meni *Podatki* omogoča upravljanje s podatki v preglednici (obseg, izbira, filtriranje). Napredna tehnologija *DataPilot* povsem enostavno izvede uvoz surovih podatkov iz korporacijskih zbirk podatkov. Meni *Okno* omogoča odpretje, razdelitev ... programskega okna (glej Sliko 16.8).

V menijski vrstici se nahaja še meni *Pomoč*, ki ponuja *Pomoč* za OOO (hiter dostop s tipko F1) in ostale podatke, ki nas v zvezi s programom zanimajo (podpora, registracija itd.) (glej Sliko 16.9).

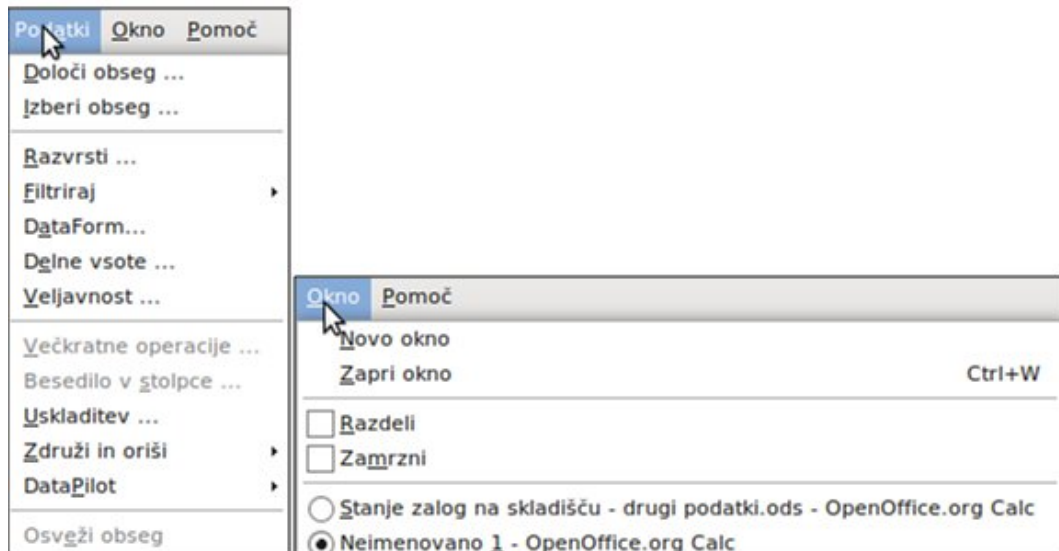
V standardni orodni vrstici se nahajajo gumbi za odpiranje, shranjevanje, tiskanje, izrezovanje, kopiranje, lepljenje itd. (glej Sliko 16.10).



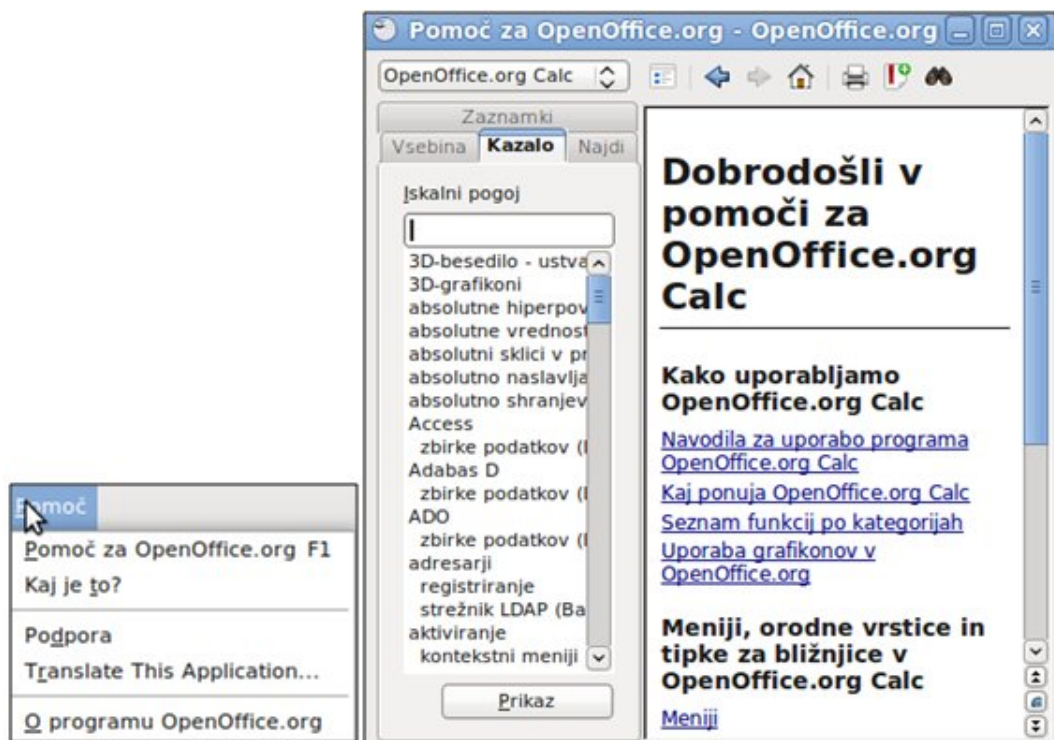
Slika 16.6: Menijska vrstica - Datoteka, Uredi in Pogled



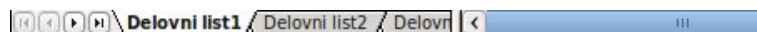
Slika 16.7: Menijska vrstica – Vstavi, Oblika in Orodja



Slika 16.8: Menijska vrstica – Podatki, Okno



Slika 16.9: Menijska vrstica Pomoč



Slika 16.12: Vrstica stanja



Slika 16.13: Vnosna vrstica

Vrstica z orodji je vrstica, kjer se nahajajo orodja, ki jih potrebujemo pri delu, glede na podatke v preglednici. Oblikujemo pisavo, spreminjamo velikost in oblike zapisanega besedila, poravnavo, zapis števila decimalnih mest, barve itd. (glej Sliko 16.11).

Spodaj levo se nahaja vrstica stanja z možnostjo pogleda med posameznimi delovnimi listi (glej Sliko 16.12).

Vnosna vrstica je vrstica, ki prikazuje natipkane podatke označene celice (podatke lahko poljubno vnašamo v celico ali v vnosno vrstico). Na levi strani vrstice so trije gumbi: *Čarovnik za funkcije*, *Vsota* in *Funkcija* (glej Sliko 16.13).

Problem

S programskim orodjem OOo Preglednica izdelamo model s katerim izvedemo izračun stanja zalog v skladišču. Pri tem tedensko vodimo stanje treh komponent (platišče, pnevmatika in vijak). Programsko orodje omogoča številne možnosti reševanja logističnih problemov, vendar se osredotočimo zgolj na en segment.

Vhodni podatki za preglednico so: začetno stanje zalog na skladišču; vrednost posameznega kosa polizdelka in kapaciteta skladišča (največja vrednost). Vsakdanje delo skladiščnika je, da sproti vpisuje vse prejeme in izdaje. Izdaje so označene z negativnimi vrednostmi, prejemi pa s pozitivnimi. Izračunati želimo: trenutno stanje zalog in njihovo vrednost; promet s polizdelki v trenutnem tednu in vrednost tega prometa ter zasedenost skladišča po posameznih polizdelkih.

V primeru, ko se zaloge v skladiščih znižajo pod 8 %, je potrebno uporabnika na to dejstvo posebej opozoriti (z rdečo barvo). Prav tako je potrebno pravilno formatirati vse podatke (količine, denar in odstotke). V nadaljevanju prikažemo dano rešitev problema.

The screenshot shows a spreadsheet titled "Stanje zalog na skladišču - drugi podatki.ods" in OpenOffice Calc. The spreadsheet contains a table with the following data:

	A	B	C	D	E
1	Stanje zalog na skladišču (tedensko)				
2					Seštevki
3	Datum	Platišča	Pnevmatike	Vijaki	
4	Cena/kos	35,00 €	60,00 €	3,00 €	
5	Zač. St. (kos)	200			800
6	12.05.2010	-96	-20		-300
7	12.05.2010	-96	-80		-100
8	13.05.2010	284	393		0
9	13.05.2010	-96	-200		880
10	13.05.2010	0	0		0
11	14.05.2010	284	393		100

Slika 16.14: Stanje zalog na skladišču

16.3 Uporaba

Vnos podatkov

Za dani problem si v Preglednici izrišemo tabelo. Zapišemo datumsko opredelitev stanja zalog v skladišču od 12. 05. 2010 do 14. 05. 2010. Zapišemo tudi začetno stanje platišč, pnevmatik in vijakov ter ceno (za kos) posamezne komponente (glej Sliko 16.14).

Sledi matematičen izračun prometa in stanja posameznih komponent (glej Sliko 16.15).

V Tabeli 16.1 je prikazan izračun posameznih vrednosti.

Za izračun prometa in stanja pnevmatik ter vijakov izvedemo podoben postopek, le da upoštevamo druge vrednosti stolpcev in vrstic. Na koncu izdelamo še seštevke prometa (v €) in stanja (v €), na sliki je označen z zeleno barvo. Izračunamo še zasedenost skladišča s posameznimi komponentami.

Izveden izračun za platišča prikazuje Tabela 16.2 in Slika 16.16.

Enako storimo za izračun zasedenosti skladišča s pnevmatikami in vijaki, le da spremenimo številke vrstic. Enostavneje izračun izvedemo s klikom na izračun, ki ga naredimo za platišča, nato pa z miško prvo vrednost prenesemo na isto raven vrstice za pnevmatike in vijake. Na takšen način se izpišejo podatki za druge polizdelke. Iz predhodnih slik so razvidni različni formati podatkov. Primer formatiranja podatka za denarno vrednost (celica B4) prikazuje Slika 16.17. V meniju *Oblika* izberemo možnost *Celice* in kategorijo (valuta, število,

Vrstica	Naslov spremenljivke	Opis	Funkcija
16 vrstica	Promet (kos)	Izračunamo tako, da zapišemo vsoto platišč od 12. do 14. maja 2010.	[=SUM:(B6:B11)]
17 vrstica	Promet (€)	Izračunamo tako, da prikažemo promet (kos), katerega pomnožimo s ceno za eno platišče.	[=B16*B4]
18 vrstica	Stanje (kos)	Izračunamo tako, da seštejemo začetno stanje platišč in promet za posamezen kos.	[=B5+B16]
19 vrstica	Stanje (€)	Izračunamo tako, da pomnožimo stanje platišč s ceno za kos.	[=B16*B4]

Tabela 16.1: Izračun prometa in stanja posameznih komponent

Vrstica	Naslov spremenljivke	Opis	Funkcija
28 vrstica	Kapaciteta	Je podana.	/
29 vrstica	Zasedenost (%)	Izračunamo tako, da stanje za kos pomnožimo s kapaciteto.	[=B16*B28]
30 vrstica	Vrednost celotne zasedenosti (€)	Izračunamo tako, da kapaciteto pomnožimo s ceno za kos	[=B28*B4]

Tabela 16.2: Zasedenost skladišča s posameznimi komponentami

Stanje zalog na skladišču (tedensko)				
Datum	Platišča	Pnevmatike	Vijaki	Seštevki
Cena/kos	35,00 €	60,00 €	3,00 €	
Zač. St. (kos)	200	200	800	
12.05.2010	-96	-20	-300	
12.05.2010	-96	-80	-100	
13.05.2010	284	393	0	
13.05.2010	-96	-200	880	
13.05.2010	0	0	0	
14.05.2010	284	393	100	
Promet (kos)	280	486	580	
Promet (€)	9.800,00 €	29.160,00 €	1.740,00 €	40.700,00 €
Stanje (kos)	480	686	1.380	
Stanje (€)	16.800,00 €	41.160,00 €	4.140,00 €	62.100,00 €

Slika 16.15: Izračun prometa in stanja posameznih komponent

odstotek itd.) ter določimo obliko in možnosti (decimalna mesta in vodilne ničle). Določimo tudi zapis valute, enako storimo še za zapis številke ali katerih drugih kategorij.

Z veljavnostjo vnosov delno preprečimo neveljavne vnose. V primeru platišč to pomeni, da je ob vnosu podatkov smiselno testirati vrednost vnosa, ki mora zadostiti naslednjima pogojevoma: ob razknjižbi izdelkov iz skladišča ne moremo izdati več blaga, kot ga je na zalogi in hkrati ob vknjižbi blaga na skladišče ne moremo sprejeti več blaga, kot je kapaciteta skladišča.

Pravilo za veljavnost vpišemo v obrazec, ki ga prikazuje Slika 16.18. Obrazec prekličemo preko menija *Podatki* → *Veljavnost*. Na drugem zavihku tega obrazca vpišemo tekst, ki ga želimo izpisati uporabniku vsakič, ko vstopi v celico (*Pomoč pri vnosu*). Na tretjem zavihku vpišemo akcijo, ki se sproži ob napačnem vnosu s pripadajočim tekstom opozorila (*Opozorilo o napaki*). Akcije so prenehanje izvajanja vnosa, izpis nalepke in vrnitev v prejšnje stanje ali zgolj izpis informacij.

Če želimo, da se ob izdaji iz skladišča upošteva pravilo, da ob enem platišču izdamo eno pnevmatiko in 5 vijakov (razmerje 1:1:5), vnosi pa so lahko poljubni, preračunamo in določimo še veljavnost pri pnevmatikah in platiščih. Ker gre pri vijakih za analogijo s pnevmatikami, si pogledjmo le slednje.

Najprej izračunamo vrednost v polju C6 (prvo polje za pnevmatike). V polje vpišemo formulo " $=IF(B6 < 0; B6;)$ ", ki pove, da v primeru, da je v B6 negativno

Stanje zalog na skladišču - drugi podatki.ods - OpenOffice.org Calc

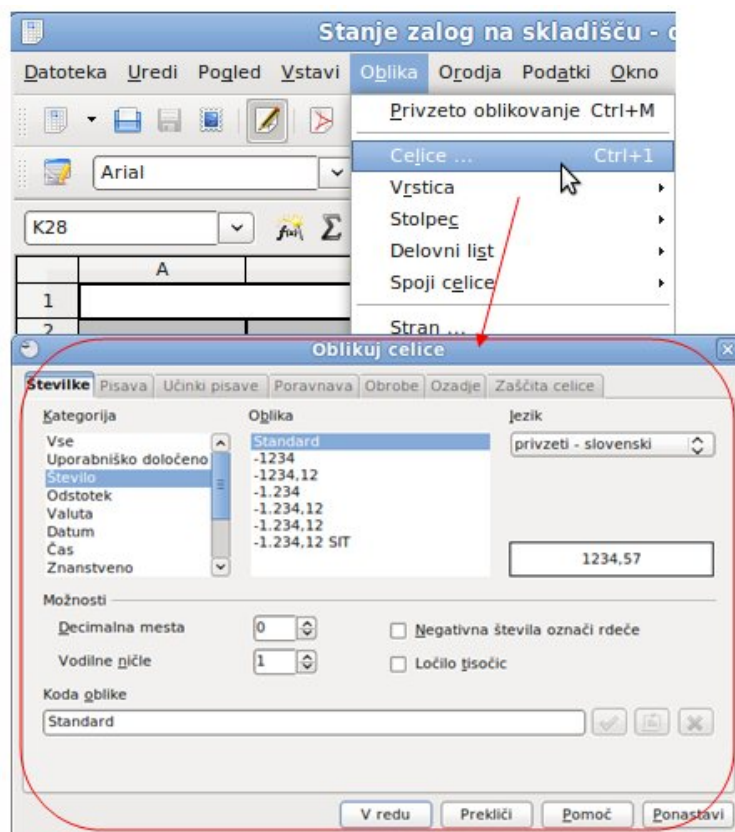
Datoteka Uredi Pogled Vstavi Oblika Orodja Podatki Okno Pomoč

Arial 10 B I U

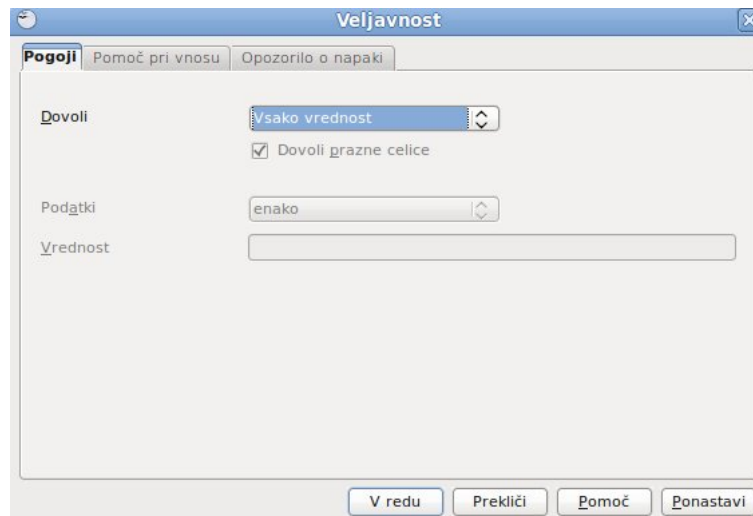
K28

	A	B	C	D	E
1	Stanje zalog na skladišču (tedensko)				
2					Seštevki
3	Datum	Platišča	Pnevmatike	Vijaki	
4	Cena/kos	35,00 €	60,00 €	3,00 €	
5	Zač. St. (kos)	200	200	800	
6	12.05.2010	-96	-20	-300	
7	12.05.2010	-96	-80	-100	
8	13.05.2010	284	393	0	
9	13.05.2010	-96	-200	880	
10	13.05.2010	0	0	0	
11	14.05.2010	284	393	100	
12					
13					
14					
15					
16	Promet (kos)	280	486	580	
17	Promet (€)	9.800,00 €	29.160,00 €	1.740,00 €	40.700,00 €
18	Stanje (kos)	480	686	1.380	
19	Stanje (€)	16.800,00 €	41.160,00 €	4.140,00 €	62.100,00 €
20					
21					
22					
23					
24					
25	Zasedenost skladišč				
26					
27		Platišča	Pnevmatike	Vijaki	
28	Kapaciteta	800	800	2600	
29	Zasedenost (%)	60%	86%	53%	
30	Vrednost celotne	28.000,00 €	48.000,00 €	7.800,00 €	83.800,00 €
31					

Slika 16.16: Zasedenost skladišča s posameznimi komponentami



Slika 16.17: Oblikovanje celic



Slika 16.18: Določanje veljavnosti

število (to je izdaja platišča), potem v obstoječe polje (to je C6) vpišemo vrednost iz B6. Sicer ne storimo nič drugega. Poleg tega napišemo še pogoj za veljavnost [=AND (AND (AND (C20≥0; C27≥C20); B6≥0; C6≥0)]. Sestavljen je iz treh pogojev, ki so:

- AND (C20≥0; C27≥C20) s katerim testiramo ali so vnosi ali iznosi v okviru zalog in kapacitete skladišča.
- AND (AND (C20≥0; C27≥C20); B6≥0) – zgornjemu pogoj iz točke 1 dodamo pogoj, da lahko v celico vpisujemo le, če je vrednost B6 pozitivno število ali število 0. Če je B6 negativno število, se v C6 vpiše pravilno negativno število, ki zagotavlja, da sta števili v B6 in C6 enaki (število pnevmatik je enako številu platišč). Vendar lahko v primeru, da je B6 pozitivno število, v okviru tega pogoja vpišemo v C6 tudi negativno število. Tega pa ne želimo, ker potem ne velja, da za vsako izdano platišče izdelamo eno pnevmatiko in obratno - za vsako izdano pnevmatiko izdamo eno platišče. To pomanjkljivost odpravimo s tretjim pogojem.
- AND (AND (AND (C20≥0; C27≥C20); B6≥0); C6≥0) s katerim zagotavljamo, da smo v celico vpisali pozitivno število ali število 0.

Z zadnjima dvema pogojema pri pnevmatikah onemogočimo kakršen koli vpis/iznos iz skladišča. Iznos, ki je mogoč, je že izračunan. Podobno ravnamo v primeru vijakov, le da v tem primeru upoštevamo v formuli faktor 5. To napravimo z naslednjim pogojem "=IF (B6 <0; 5*B6;)".

Povzetek

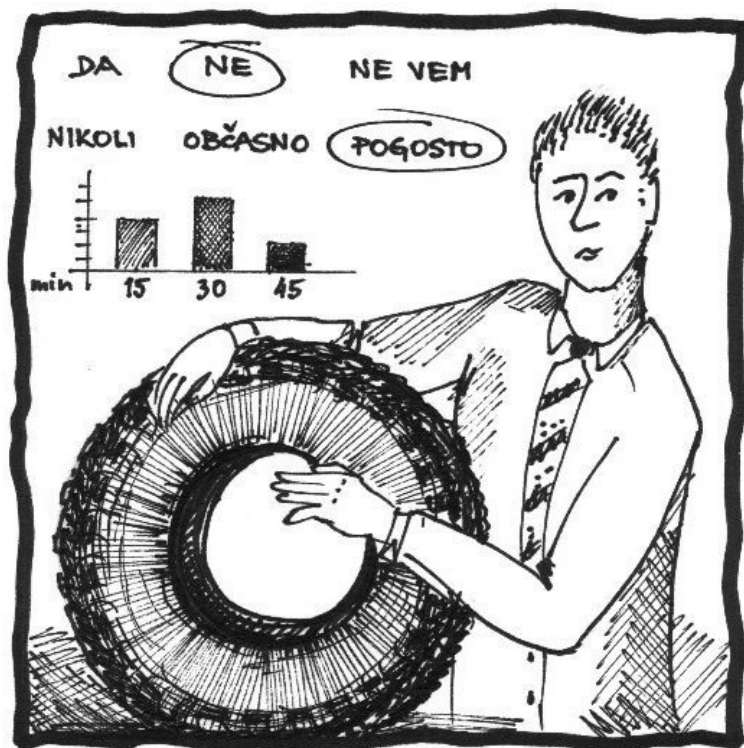
OOo je najnovejša različica osrednjega svetovnega brezplačnega in odprto-kodnega pisarniškega paketa, ki je na voljo, v številnih svetovnih jezikih, za vse pomembnejše operacijske sisteme. Vsebuje šest različnih modulov s katerimi lahko upravljamo. V izbranem primeru smo se osredotočili na Preglednico, katero uporabljamo za delo z računalniškimi tabelami ter s številnimi drugimi temami (vnos in spreminjanje podatkov, premikanje aktivne celice v preglednici, vstavljanje in brisanje vrstic, oblikovanje celic in podatkov, izdelava grafikonov, uporaba funkcij itd.).

Z izbranim programskim orodjem smo prikazali enostaven problem izračuna zalog in zasedenosti komponent v skladišču. S to vrstnimi programskimi orodji brezplačno in na enostaven način rešimo marsikateri logistični problem.

Pri opisu programskega orodja OOo Preglednica smo uporabili še dodatne vire in literaturo: [72] [39].

Poglavje 17

PSPP - statistična analiza podatkov



Vnos podatkov zbranih z anketnimi vprašalniki
Izvajanje opisne statistike, testov ipd.
Primer: statistična analiza zadovoljstva prevzema pnevmatik

17.1 Teoretično ozadje

17.1.1 Statistična analiza podatkov

Statistična analiza podatkov je ena izmed pomembnih aktivnosti na vseh področjih raziskovanja. V podjetjih jo pogosto uporabljamo, kadar želimo analizirati določeno količino podatkov ali se morda osredotočiti zgolj na en vidik posameznega procesa ipd. Običajno se izvaja na podlagi že zbranih podatkov oz. z zbiranjem podatkov z anketnimi vprašalniki.

Najpogosteje se lotimo preučevanja podatkov z metodami srednjih vrednosti. Okrog reprezentativnih vrednosti spremenljivke se gostijo posamezne vrednosti enot populacije. Med najpogostejše srednje vrednosti uvrščamo aritmetično sredino, mediano ali središnico ter modus ali gostišnico. Sledijo mere variacij, med katere prištevamo variacijski razmik, varianco in standardni odklon. S tema dvema vrednostma opišemo značilnosti frekvenčne porazdelitve obravnavane spremenljivke.

Za učinkovito analizo je smiselno uporabiti nekatere parametrične in neparametrične teste. Pri preučevanju nekaterih kompleksnih pojavov upoštevamo veliko medsebojno odvisnih spremenljivk. Na voljo sta dve možnosti, in sicer multipla regresija in faktorska analiza.

17.2 O programskem orodju

PSPP je programsko orodje za statistično analizo podatkov. Na videz je podobno plačljivemu programskemu orodju SPSS. Zavedati se je potrebno, da je programsko orodje PSPP del GNU GPL licence, torej ni omejitev za število primerov ali spremenljivk, ki jih uporabimo, ni dodatnih paketov za pridobitev "naprednih" funkcij ipd. Vse funkcije, ki jih PSPP podpira, so v jedru paketa. PSPP je še posebej namenjen statistikom, sociologom in študentom, ki zahtevajo hitro in priročno analizo vzorčnih podatkov. S pomočjo programskega orodja PSPP izvajamo opisne statistike, T-teste, linearno regresijo, ne-parametrične teste itd.

Prednosti PSPP so predvsem, da:

- podpira več kot 1 milijardo primerov;
- podpira več kot 1 milijardo spremenljivk;
- sintakse in podatkovne datoteke so združljive s SPSS;
- ponuja možnost izbire terminala ali grafičnega uporabniškega vmesnika;

- deluje znotraj Gnumeric, OOo in drugih prostih programov;
- je enostaven uvoz podatkov iz preglednic, tekstovnih datotek in baz virov;
- omogoča hitre statistične postopke, tudi pri veliki količini podatkov;
- nima licenčnih pogodb;
- deluje na več različnih platformah (različni računalniki, različni operacijski sistemi) itd.

Prenos in namestitvev

Programsko orodje PSPP namestimo s spletne strani PSPP [42] (okolje Windows), kjer izberemo možnost *Get PSPP*. Verzijo izberemo glede na izbrani operacijski sistem (MS Windows, Ubuntu, Mac OS X, Open SuSe ipd.). Če uporabljamo Ubuntu, programsko orodje poiščemo v zbirki programov in ga namestimo. V nadaljevanju prikažemo posamezne razlike med verzijami, ki so dostopne.

Programsko orodje PSPP zaženemo s klikom na ikono PSPP, na kar se odpre okno za vnos podatkov. V menijski vrstici so možnosti s katerimi upravljamo. Na voljo sta dva lista (spodaj levo): *Data View* in *Variable View*. Vsaki enoti ustreza ena vrstica, vsaki spremenljivki pa en stolpec (pogled *Data View*). S klikom na ime spremenljivke (vrh stolpca) ali z izborom pogleda *Variable View* pogledamo, kako je spremenljivka definirana (glej Sliko 17.1).

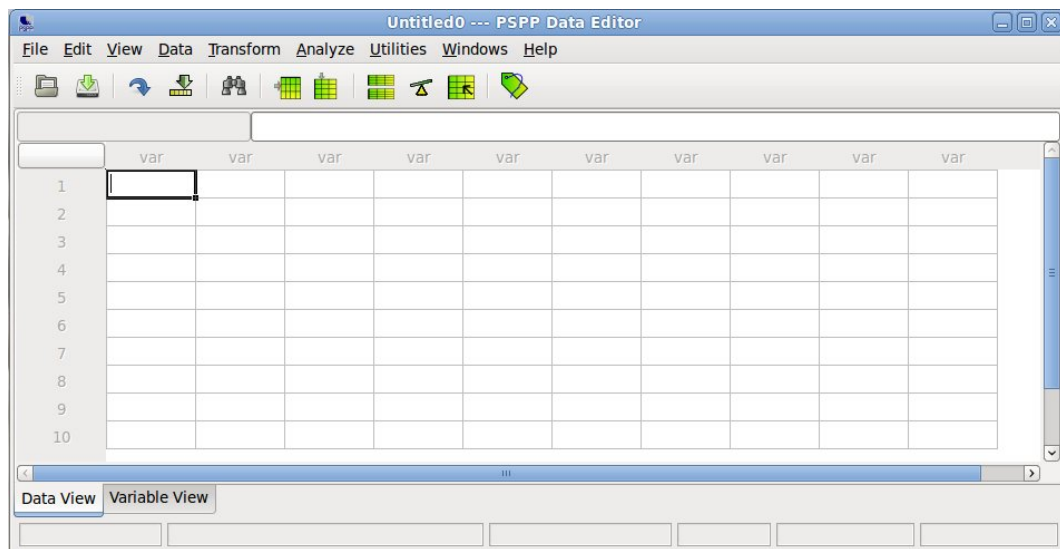
V menijski vrstici *File* je možnost izbire posameznih orodij. Izbiramo med možnostmi odpiranja nove datoteke (*New*), odpremo že shranjeno datoteko (*Open*), shranimo dokument (*Save, Save As*), zapremo dokument (*Quit*) itd. (glej Sliko 17.2).

Menijska vrstica *Edit* prikaže možnosti dodajanja vrstic (*Insert Variable*) in stolpcev (*Insert Cases*) (glej Sliko 17.3).

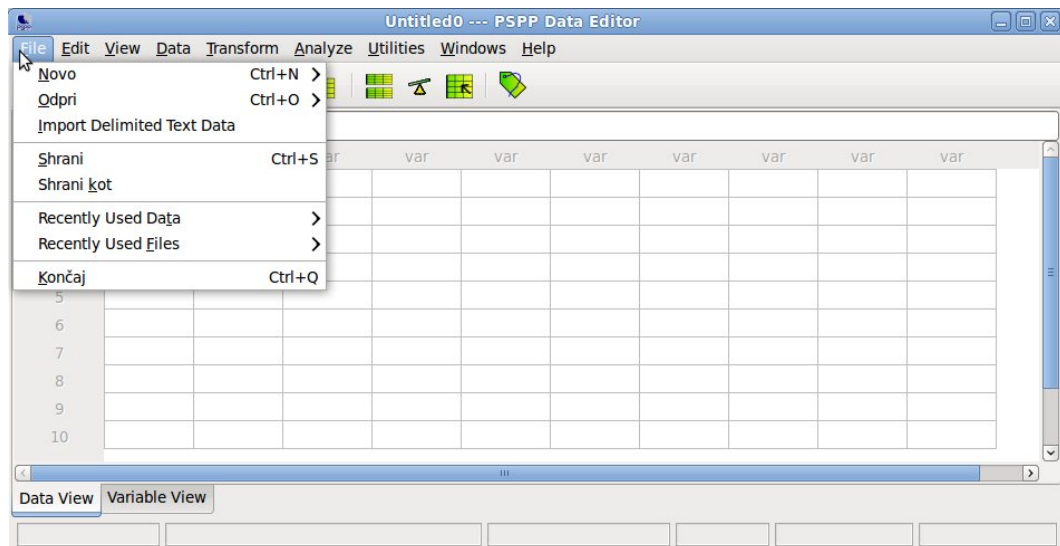
Menijska vrstica *View* omogoča urejanje osnovnega okna (dodajanje črt, pisava ipd.) (glej Sliko 17.4).

Menijsko vrstico *Data* uporabljamo pri nadaljnji analizi podatkov, kjer imamo možnost razdružiti posamezne spremenljivke (*Split File*), jih obtežiti (*Weight Cases*) ipd. (glej Sliko 17.5).

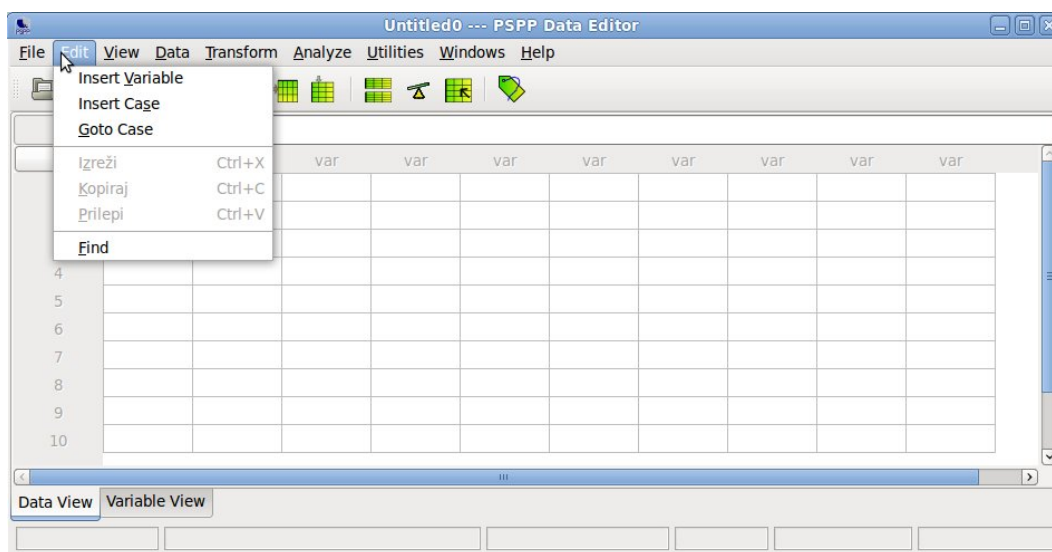
Pretvorbo in manipulacijo podatkov je mogoče hitro doseči z uporabo PSPP transformacije. Transformacija omogoča, da določimo dejavnosti, ne da bi se pri tem podatki ponavljali. V menijski vrstici *Transform* imamo možnost transformiranja podatkov (glej Sliko 17.6). Primer: zapisane podatke v številkah (starost) želimo združiti v skupine.



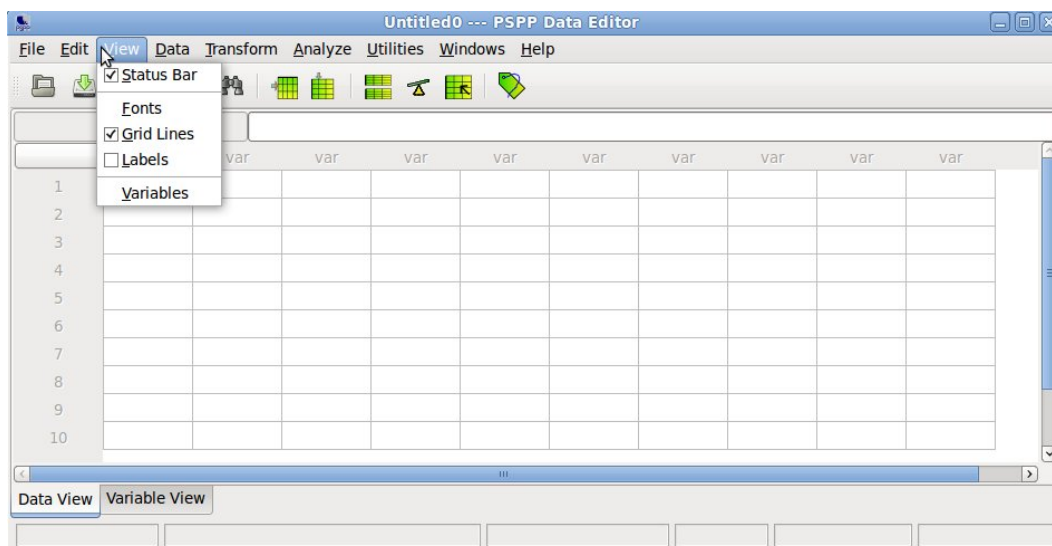
Slika 17.1: Osnovno okno



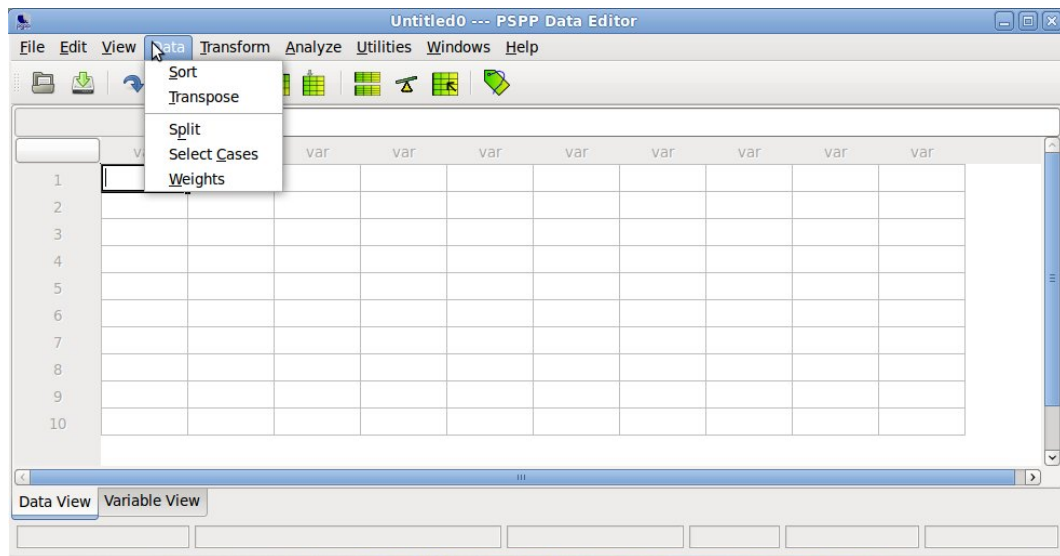
Slika 17.2: Menijska vrstica File



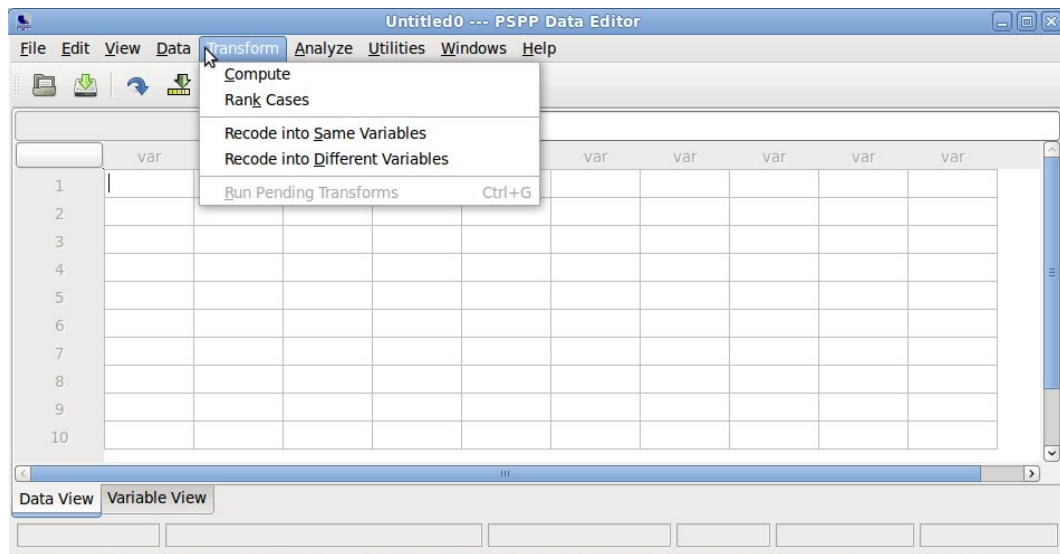
Slika 17.3: Menijska vrstica Edit



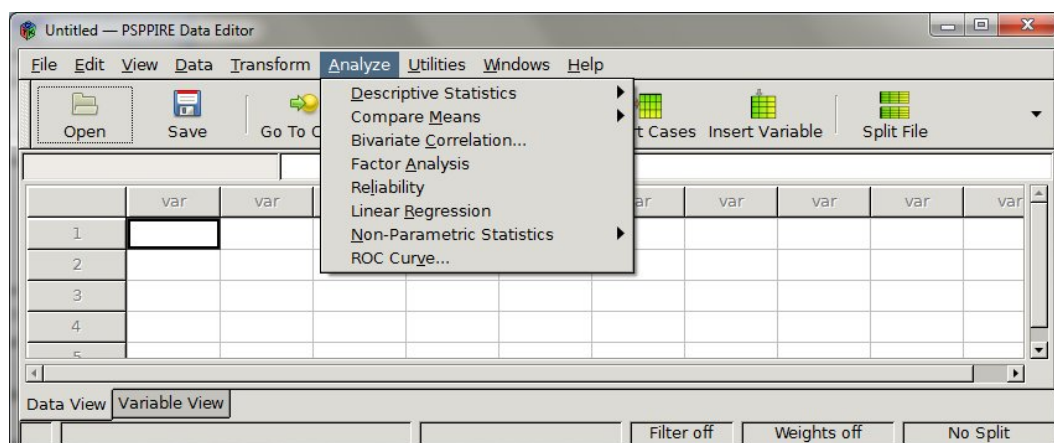
Slika 17.4: Menijska vrstica Edit



Slika 17.5: Menijska vrstica Data



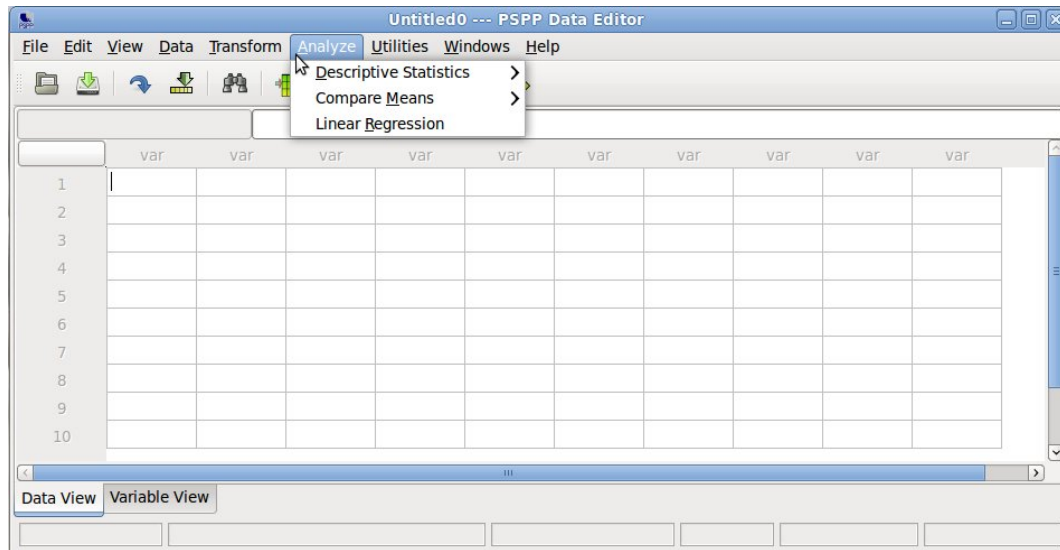
Slika 17.6: Orodna vrstica Transform



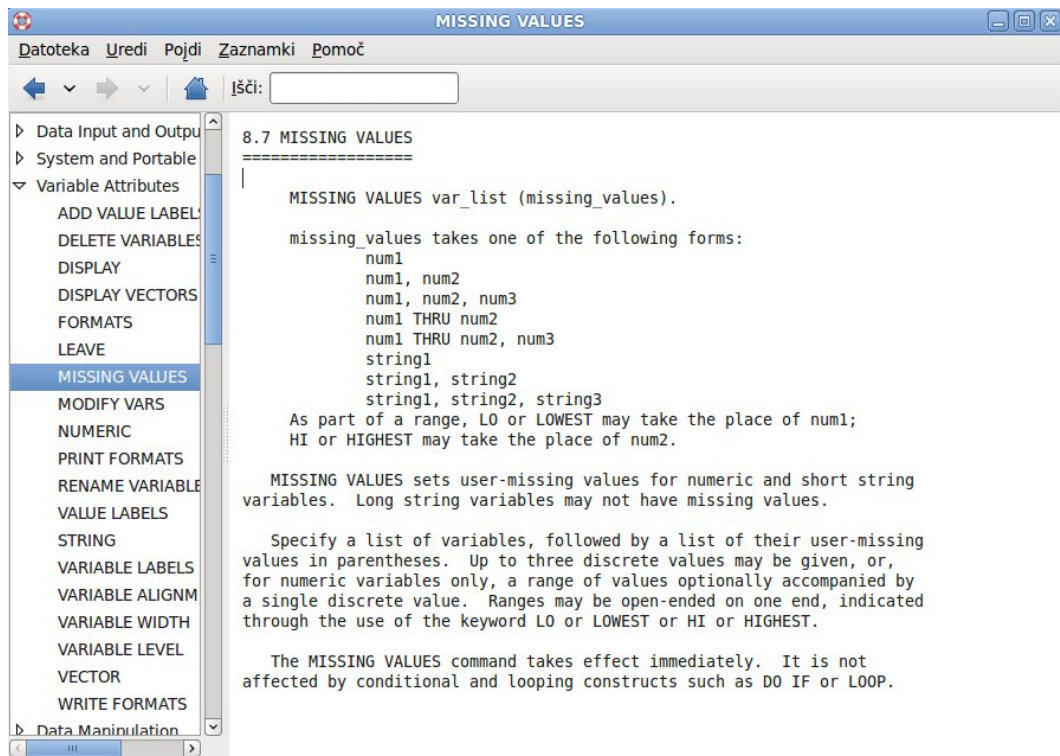
Slika 17.7: Menijska vrstica Analyze v Windows okolju

Menijska vrstica *Analyze* je ena izmed najpomembnejših vrstic v programu SPSS, kjer izbiramo med analizami, ki jih izvedemo. Verzija dostopna za okolje Windows vsebuje več različnih orodij za obdelavo podatkov, kot verzija za Ubuntu Linux. Izberemo lahko med deskriptivno statistiko (*Descriptive Statistics*), primerjalno analizo (*Compare Means*), bivariatno analizo (*Bivariate Correlation*), faktorsko analizo (*Factor Analysis*), linearno regresijo (*Linear Regression*), neparametričnimi testi (*Non-Parametric Statistics*) itd. Posamezna analiza vključuje še podpoglavja z notranjimi analizami (npr. pri neparametričnih testih lahko izbiramo med hi-kvadratom (*Chi-Square*) in binomskim testom (*Binomial*)). Število dostopnih orodij je posledica različnih verzij, ki so dostopne (glej Sliki 17.7 in 17.8).

V osnovni orodni vrstici je na voljo še troje programskih oken, in sicer *Utilities*, *Windows* in *Help*. Pomoč (*Help*) uporabimo, kadar želimo izvedeti kaj več ali pa kadar se pojavi težava, katero lahko s SPSS priročnikom enostavneje razrešimo (glej Sliko 17.9).



Slika 17.8: Menijska vrstica Analize v Ubuntu Linux



Slika 17.9: Razdelek v menijski vrstici Help

Problem

V izbranem primeru izvedemo analizo podatkov, zbranih na podlagi anketnega vprašalnika. Zanima nas, kako so zaposleni v podjetju OpenStorage zadovoljni z delovnim mestom v skladišču, natančneje s postopkom prevzema pnevmatik. Zastavimo jim 7 krajših vprašanj, katere s pomočjo programskega orodja analiziramo in predstavimo rezultate. Pomembno je, da se pri izbranem problemu ne osredotočimo le na matematične izračune, ampak da za izboljšanje delovnih navad upoštevamo tudi mnenje zaposlenih na izbranih položajih. Anketni vprašalnik je prikazan na Sliki 17.10.

Anketni vprašalnik

1. Kakšen je vaš položaj v podjetju?					
a) vodja skladišča		b) skladiščnik		c) pomočnik skladiščnika	
2. Menite, da ste na svojem delovnem mestu preveč obremenjeni?					
a) da			b) ne		
3. V kolikšni meri, poleg svojega dela, opravljate zadolžitve še za druga delovna mesta (prosim označite s križcem)?					
	1-nikoli	2-redko	3-občasno	4-pogosto	5-zelo pogosto
Opravljanje drugega dela					
4. Število prevzemov pnevmatik, ki jih v podjetju opravite na izmeno.					
Napišite število: _____					
5. V kolikšni meri ste zadovoljni z izvedbo procesa prevzema pnevmatik?					
1-zelo nezadovoljen	2-nezadovoljen	3-niti ne zadovoljen niti zadovoljen	4-zadovoljen	5-zelo zadovoljen	6-ne vem
6. V povprečju koliko časa je trajal prevzem enega kamiona pnevmatik, pred uvedbo elektronskega poslovanja?					
a) Manj kot 15 min		b) od 15 do manj kot 30 min		c) od 30 do manj 1h	
d) več kot 1 h					
7. Koliko časa traja v povprečju, prevzem enega kamiona pnevmatik (po uvedbi elektronskega poslovanja)?					
a) Manj kot 15 min		b) od 15 do manj kot 30 min		c) od 30 do manj 1h	
d) več kot 1 h					

Slika 17.10: Anketni vprašalnik

17.3 Uporaba

Priprava baze - Variable View

Anketni vprašalnik oz. drugo bazo podatkov je potrebno ročno vnesti v programsko orodje PSPP. Bazo pripravimo tako (glej Sliko 17.11), da nastavimo pogled na *Variable View*, kjer spremenljivkam določimo: format (glede na zapis vrednosti); tip (glede na merske lestvice); imena (glede na vsebino); labele in manjkajoče vrednosti.

Bazo nastavimo na način, da v razdelek *Name* vnesemo ime spremenljivke (npr. položaj). V okolju Ubuntu v samem zapisu ne smemo uporabljati šumnikov, kar je v okolju Windows dovoljeno. V razdelek *Type* zapišemo tip spremenljivke, kjer je na voljo več možnosti. Spremenljivka je lahko številka (*Numeric*), datumska (*Date*), valutna (*Dollar*), besedna (*String*) itd. (glej Sliko 17.12).

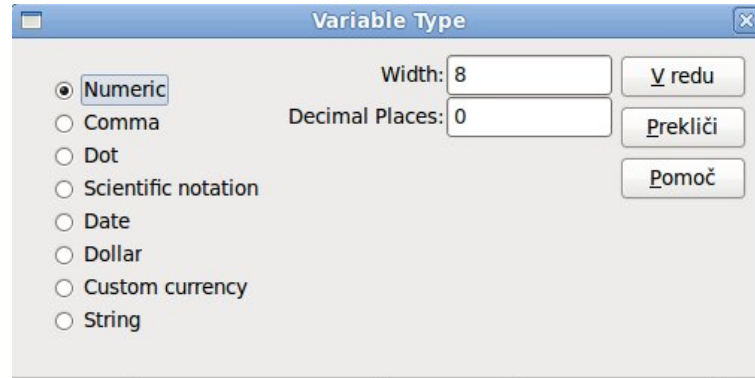
V razdelku *Width* določimo, koliko števil je lahko maksimalno postavljeno pred decimalno vejico, v razdelki *Decimals* pa koliko za decimalno vejico. V razdelek *Label* opisno vpišemo, kaj spremenljivka predstavlja. Običajno v to polje vnesemo vprašanje, ki se nanaša na spremenljivko (npr. Kakšen je vaš položaj v podjetju?).

V razdelek *Values* vnesemo opis posameznih spremenljivk. V PSPP vnesemo vrednosti spremenljivk številčno (npr. 1 - vodja skladišča, 2 - skladiščnik in 3 - pomočnik skladiščnika). Pod *Value* vnesemo številko, katere pomen razložimo pod *Value Label*. V primeru, ko imamo številčno in ne opisno spremenljivko, in imamo npr. starost anketirancev, polje pustili prazno (*None*) (glej Sliko 17.13).

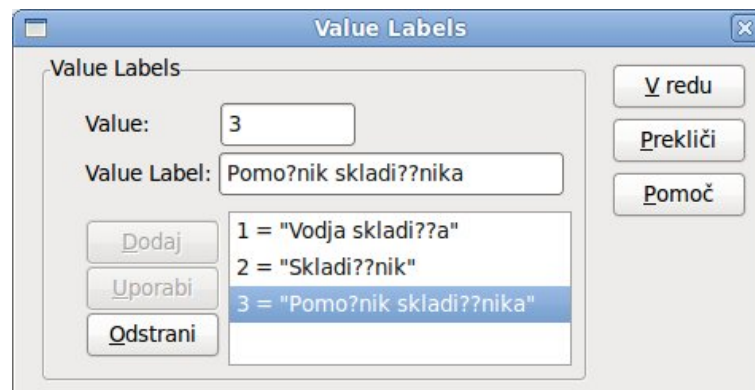
V razdelku *Missing* določimo manjkajoče vrednosti. Manjkajoče vrednosti dobimo kadar npr. anketiranec ne odgovori na vprašanje. Običajno jih označimo s številko 9 ali 99. Kadar se vprašanje nanaša na starost običajno uporabimo vrednost -1. Za odgovor ne vem (ali ne želimo odgovoriti) uporabimo številko 9 in 99, če ni odgovora. Določeno vrednost vnesemo v okvirček *Discrete missing values*. Napišemo lahko več, eno ali nobene številke. Pomen označbe manjkajočih vrednosti je v tem, da program avtomatsko izloči te odgovore iz



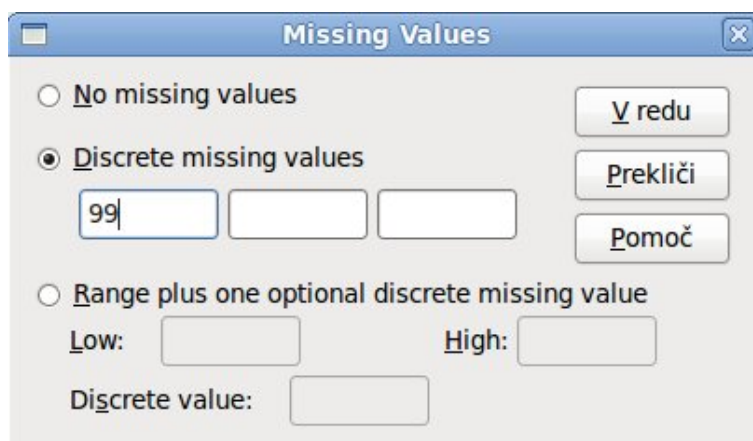
Slika 17.11: Priprava baze Variable View



Slika 17.12: Tip spremenljivke



Slika 17.13: Opis spremenljivk



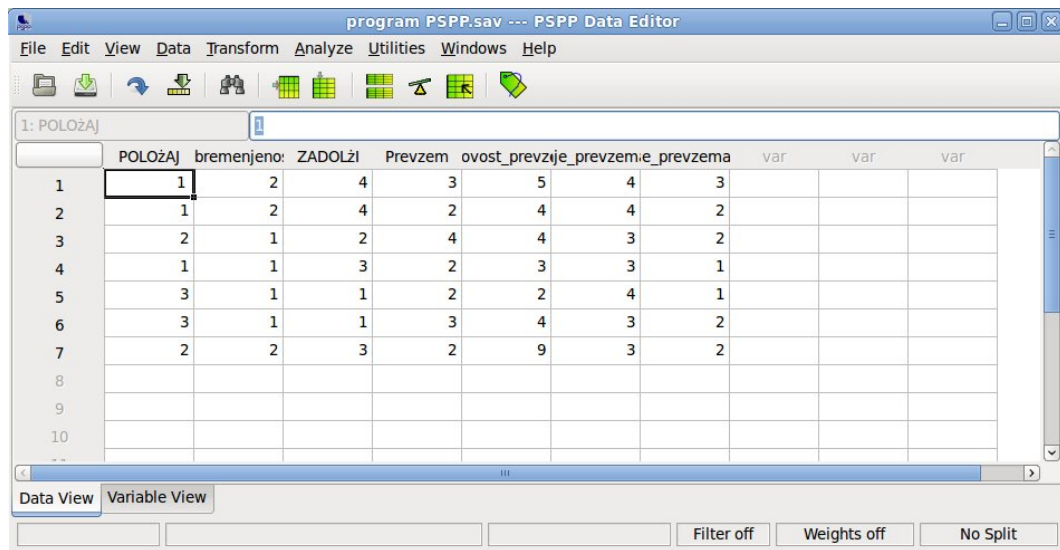
Slika 17.14: Manjkajoče vrednosti

analize (glej Sliko 17.14).

Razdelek *Align* pove kakšna naj bo poravnava besedila (levo - *Left*, desno - *Right* ali na sredini - *Center*). Na koncu je še razdelek *Measure*, s katerim določimo ali gre za nominalne, ordinalne ali številske spremenljivke. Nominalne spremenljivke (*Nominal*) – vrednosti lahko le razlikujemo med seboj, lahko sta enaki ali različni (npr. spol, ime, barva, poklic, vrste dela itd.) *Ordinalne spremenljivke* (*Ordinal*) – vrednosti lahko uredimo po velikosti, lahko sta enaki ali pa je ena izmed njiju večja oz. manjša (npr. izobrazba, strinjanje – stopnjevanje ipd.). Intervalne in razmernostne spremenljivke (*Scale*) – primerjamo lahko razlike med vrednostmi, dve vrednosti sta enaki ali pa je razlika med njima enaka nekemu številu, ki je različen od nič (npr. temperatura, strinjanje – izraženo s številčno oceno itd.); ali pa sta dve vrednosti enaki oz. je njun kvocient enak nekemu številu, različnemu od števila ena (npr. starost, dohodek, ure dela itd.).

Vnos podatkov - Data View

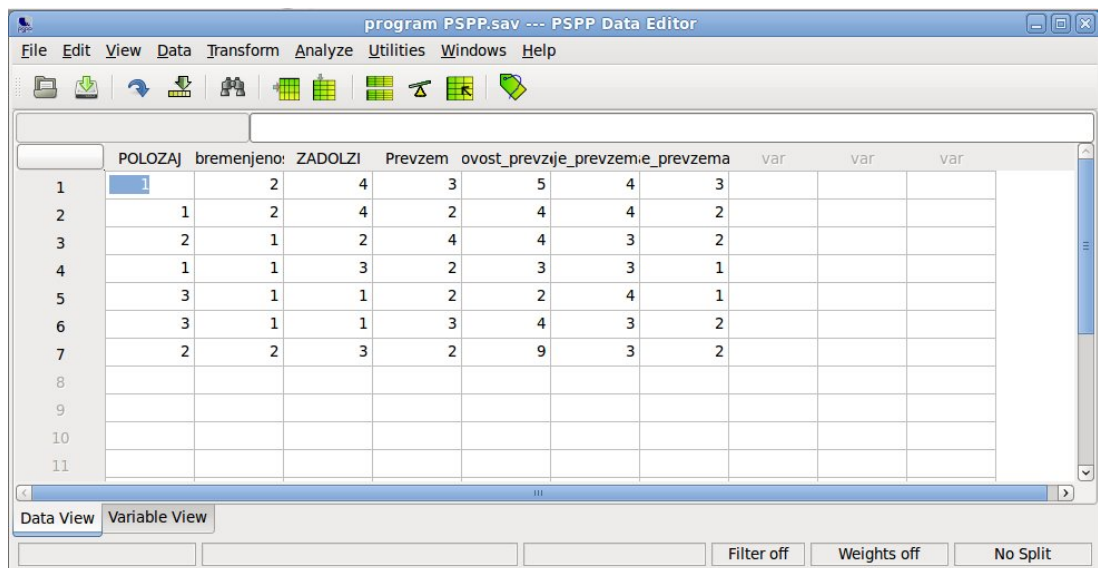
Ob zapisani bazi podatkov, levo spodaj preklopimo na *Data View*, kamor vnesemo rezultate naše analize. Številka ob levi strani pove, koliko enot je zajetih v analizo. Največkrat so enote anketiranci oz. izpolnjeni anketni vprašalniki. Kot primer prikažemo zgolj vnos manjšega števila podatkov, seveda pa je PSPP namenjen obdelavi velike baze podatkov (glej Sliko 17.16).



The screenshot shows the PSPP Data Editor window with a data table. The table has 10 rows and 8 columns. The first column is labeled '1' and contains row numbers. The second column is labeled 'POLOŽAJ' and contains values 1, 1, 2, 1, 3, 3, 2, and empty cells for rows 8-10. The third column is labeled 'bremenjeno:' and contains values 2, 2, 1, 1, 1, 1, 2, and empty cells for rows 8-10. The fourth column is labeled 'ZADOLŽI' and contains values 4, 4, 2, 3, 1, 1, 3, and empty cells for rows 8-10. The fifth column is labeled 'Prevzem' and contains values 3, 2, 4, 2, 2, 3, 2, and empty cells for rows 8-10. The sixth column is labeled 'ovost_prevzije_prevzeme_prevzema' and contains values 5, 4, 4, 3, 2, 4, 9, and empty cells for rows 8-10. The seventh column is labeled '4' and contains values 4, 4, 3, 4, 3, 3, 3, and empty cells for rows 8-10. The eighth column is labeled '3' and contains values 3, 2, 2, 1, 4, 2, 2, and empty cells for rows 8-10. There are also three empty columns labeled 'var'.

	POLOŽAJ	bremenjeno:	ZADOLŽI	Prevzem	ovost_prevzije_prevzeme_prevzema	4	3	var	var	var
1	1	2	4	3	5	4	3			
2	1	2	4	2	4	4	2			
3	2	1	2	4	4	3	2			
4	1	1	3	2	3	3	1			
5	3	1	1	2	2	4	1			
6	3	1	1	3	4	3	2			
7	2	2	3	2	9	3	2			
8										
9										
10										

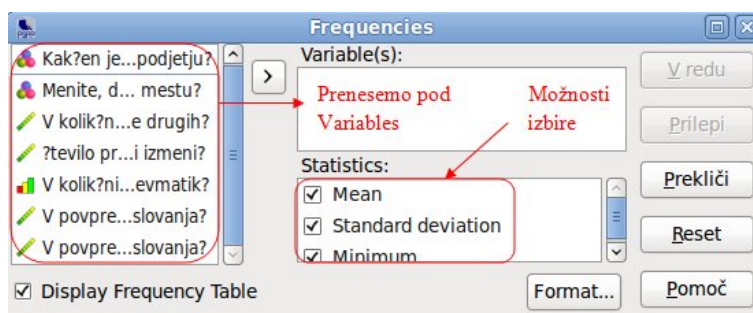
Slika 17.15: Pripravljena baza



The screenshot shows the PSPP Data Editor window with the same data table as in Slika 17.15. The first row is highlighted, and the value '1' is entered in the 'POLOŽAJ' column. The rest of the data in the table is the same as in the previous screenshot.

	POLOŽAJ	bremenjeno:	ZADOLŽI	Prevzem	ovost_prevzije_prevzeme_prevzema	4	3	var	var	var
1	1	2	4	3	5	4	3			
2	1	2	4	2	4	4	2			
3	2	1	2	4	4	3	2			
4	1	1	3	2	3	3	1			
5	3	1	1	2	2	4	1			
6	3	1	1	3	4	3	2			
7	2	2	3	2	9	3	2			
8										
9										
10										
11										

Slika 17.16: Vnos podatkov



Slika 17.17: Frekvenčna statistika

Statistična analiza podatkov

Urejena baza je pogoj za pričetek statistične analize podatkov. Kadar želimo izvesti osnovno opisno statistiko v orodni vrstici *Analyze* izberemo možnost *Descriptive Statistics*, kjer so na voljo štiri možnosti. Izvedemo lahko statistiko imenovano *Frequencies* (na desni strani okna so zapisane vse spremenljivke). V okno *Variable(s)* prenesemo le tiste, ki jih želimo analizirati. V programskem oknu *Statistics* označimo možnosti, katere želimo, da so vključene v obdelavo (glej Sliko 17.17).

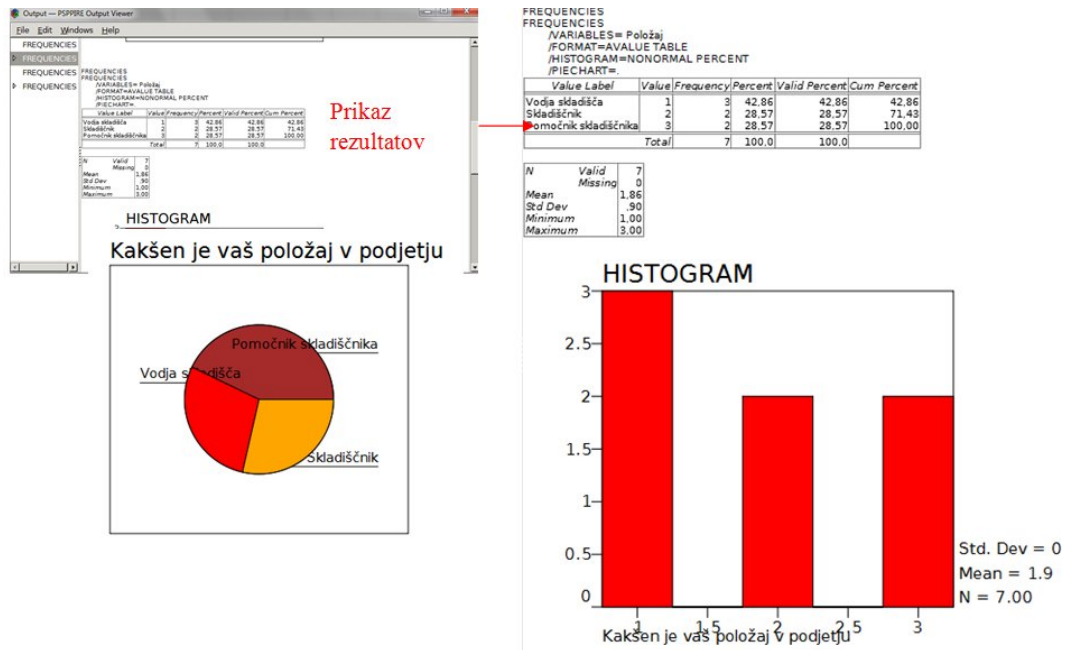
Možnost prikaza podatkov je, ne le s tabelami, ampak tudi z grafi (kolač, histogram). S klikom na miško označimo možnosti, ki jih želimo vključiti. Vse skupaj potrdimo s tipko *Continue*, in na koncu *OK*. Rezultati izvedene analize se izpišejo v drugem oknu, ki ga lahko shranimo kot svojo datoteko (glej Sliko 17.18).

Na primerljiv način se izvede opisna statistika, nahajajoča v razdelku *Descriptives*, kjer posamezne spremenljivke, ki jih želimo analizirati prenesemo v okno *Variables*. V programskem oknu *Statistics* označimo možnosti, katere želimo, da so vključene v obdelavo podatkov (glej Sliko 17.19).

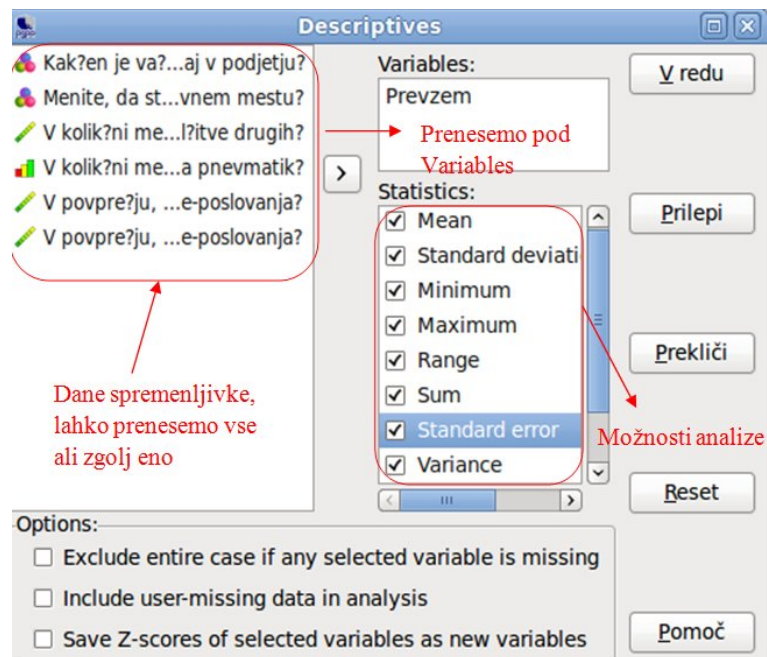
Izpišejo se rezultati o skupnem številu anketirancev (*N*), povprečni vrednosti danega odgovora (*Mean*), standardni odklon (*Std. Dev.*), varianca (*Variance*) in številni drugi statistični podatki (glej Sliki 17.20 in 17.21).

V razdelku *Descriptive Statistics* se nahaja podrazdelek *Crosstabs*. Če je izbrana spremenljivka nominalna ali ordinalna, njeno povezanost s pripadnostjo skupini ugotavljamo s kontingenčno tabelo in izračunom hi-kvadrata (npr. Ali obstaja povezanost med obremenitvijo na delovnem mestu, glede na položaj v podjetju?) (glej Sliki 17.22 in 17.23).

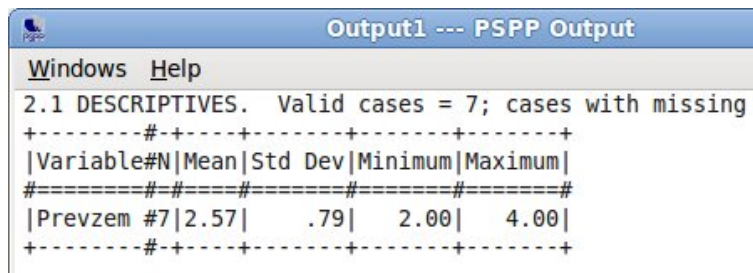
Prikazali smo zgolj nekaj izmed možnih analiz. V nadaljevanju izvedemo analize tudi za več spremenljivk skupaj. Izdelamo lahko analizo T-test za en vzorec (*One Sample T Test*), T-test za dva neodvisna vzorca (*Independent Samples*



Slika 17.18: Izpis rezultatov frekvenčne statistike



Slika 17.19: Opisna statistika



Output1 --- PSPP Output

Windows Help

2.1 DESCRIPTIVES. Valid cases = 7; cases with missing

```

+-----#-----+
|Variable#N|Mean|Std Dev|Minimum|Maximum|
#-----#-----#-----#-----#-----#
|Prevzem #7|2.57| .79| 2.00| 4.00|
+-----#-----+

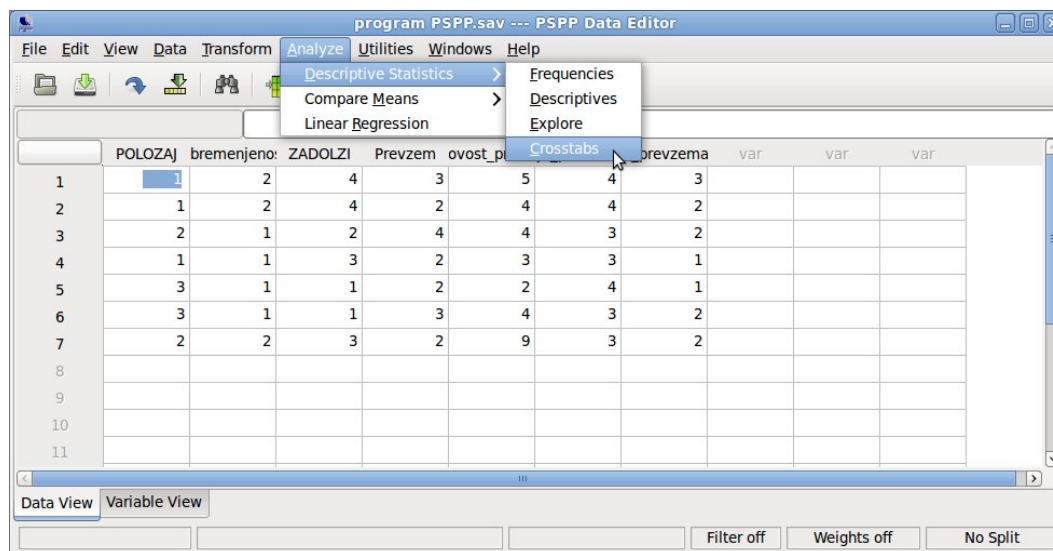
```

Slika 17.20: Rezultati opisne statistike - Ubuntu in Windows

DESCRIPTIVES
DESCRIPTIVES
/VARIABLES= Prevzem
/STATISTICS=ALL.

Variable	N	Mean	S.E. Mean	Std. Dev.	Variance	Kurtosis	S.E. Kurt	Skewness	S.E. Skew	Range	Minimum	Maximum	Sum
Prevzem	7	2.57	.30	.79	.62	.27	1.59	1.11	.79	2.00	2.00	4.00	18.00

Slika 17.21: Rezultati opisne statistike - Windows



program PSPP.sav --- PSPP Data Editor

File Edit View Data Transform Analyze Utilities Windows Help

Descriptive Statistics > Frequencies
Compare Means > Descriptives
Linear Regression
Crosstabs

	POLOZAJ	bremenjeno	ZADOLZI	Prevzem	ovost_p	prevzema	var	var	var
1	1	2	4	3	5	4	3		
2	1	2	4	2	4	4	2		
3	2	1	2	4	4	3	2		
4	1	1	3	2	3	3	1		
5	3	1	1	2	2	4	1		
6	3	1	1	3	4	3	2		
7	2	2	3	2	9	3	2		
8									
9									
10									
11									

Data View Variable View

Filter off Weights off No Split

Slika 17.22: Razdelek Crosstabs

CROSSTABS
CROSSTABS
/TABLES= Obremenjenost BY Položaj
/FORMAT=AVALUE LABELS TABLES PIVOT
/STATISTICS=CHISQ
/CELLS=COUNT ROW COLUMN TOTAL.

		Cases					
		Valid	Missing	Total			
		N	Percent	N			
Menite da ste preveč obremenjeni na svojem delovnem mestu? * Kakšen je vaš položaj v podjetju		7	100.0%	0	0.0%	7	100.0%

Obremenjenost	Položaj			Total
	Vodja skladišča	Skladiščnik	Pomočnik skladiščnika	
Da	1,0 25,0% 33,3% 14,3%	1,0 25,0% 50,0% 14,3%	2,0 50,0% 100,0% 28,6%	4,0 100,0% 57,1% 57,1%
Ne	2,0 66,7% 66,7% 28,6%	1,0 33,3% 50,0% 14,3%	0,0 .0% .0% .0%	3,0 100,0% 42,9% 42,9%
Total	3,0 42,9% 100,0% 42,9%	2,0 28,6% 100,0% 28,6%	2,0 28,6% 100,0% 28,6%	7,0 100,0% 100,0% 100,0%

Statistic	Value	df	Asymp. Sig. (2-sided)
Pearson Chi-Square	2,24	2	.33
Likelihood Ratio	2,97	2	.23
Linear-by-Linear Association	1,78	1	.18
N of Valid Cases	7		

Kontigenčna tabela

Hi-kvadrat

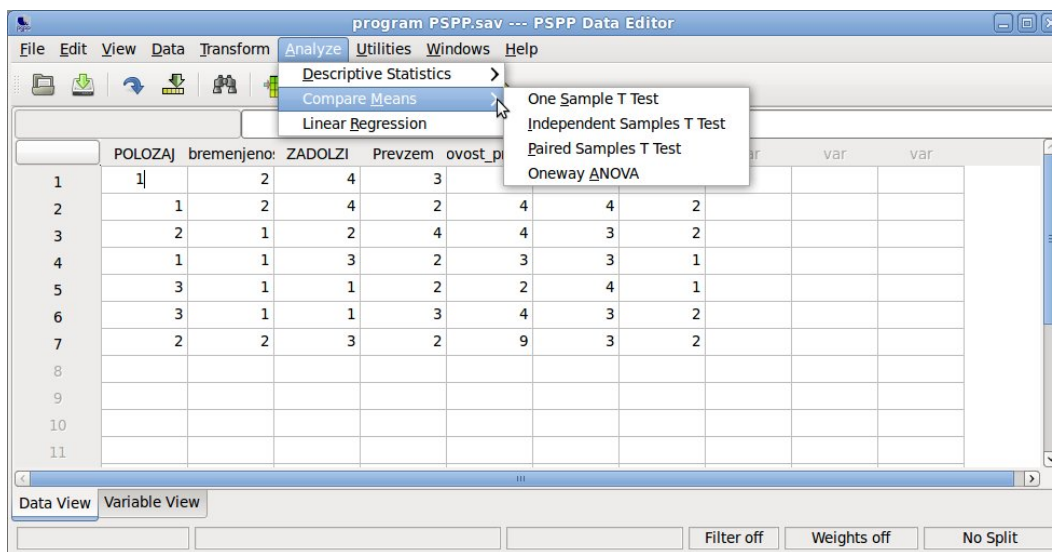
Slika 17.23: Rezultati povezanosti dveh spremenljivk

T Test), T-test za dva odvisna vzorca (*Paired Samples T Test*) in Anovo (*One Way ANOVA*) (glej Sliko 17.24).

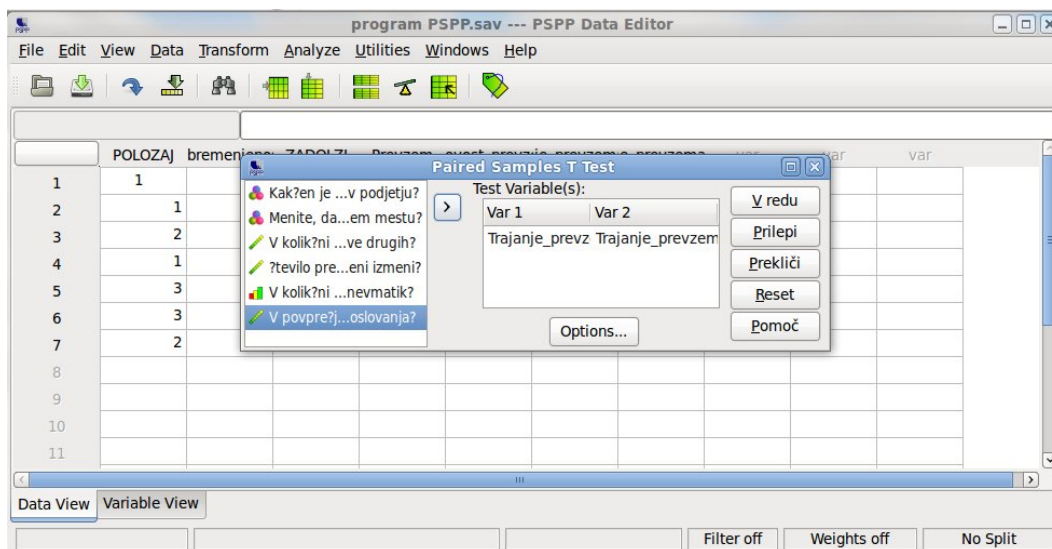
Prikaz T-testa za dva odvisna vzorca je prikazan na Sliki 17.25. V okence *Var1* in *Var2* prenesemo spremenljivke, ki jih želimo primerjati med sabo. Te vrste primerjava je smiselna predvsem, če imamo na voljo podatke stanja v preteklosti in v sedanjosti.

Pri analizi in za lažje razumevanje rezultatov je priporočljivo, da predhodno preučimo določena statistična področja. Če je vrednost *Sig. (2-tailed)* manj kot 5 % pomeni, da med spremenljivkama obstaja statistično signifikantna razlika, kar drži tudi v našem primeru (glej Sliko 17.26).

V menijski vrstici *Data* obstaja možnost izbire *Split File*, ki omogoča, da ločeno prikažemo rezultate posameznih analiz. Primer: Menite, da ste na svojem delovnem mestu preveč obremenjeni? Anketiranec odgovori z Da ali Ne. Še posebej je ločeno obravnavanje rezultatov smiselno v primeru, ko vprašanje povežemo z drugim vprašanjem (glej Sliki 17.27 in 17.28).



Slika 17.24: Možnosti izbire analiz



Slika 17.25: T-test za dva odvisna vzorca

T-TEST
T-TEST
PAIRS = Trajanje_pre vzema_prej WITH Trajanje_pre vzema_sedaj (PAIRED)
/MISSING=ANALYSIS
/CRITERIA=CIN(0.95).

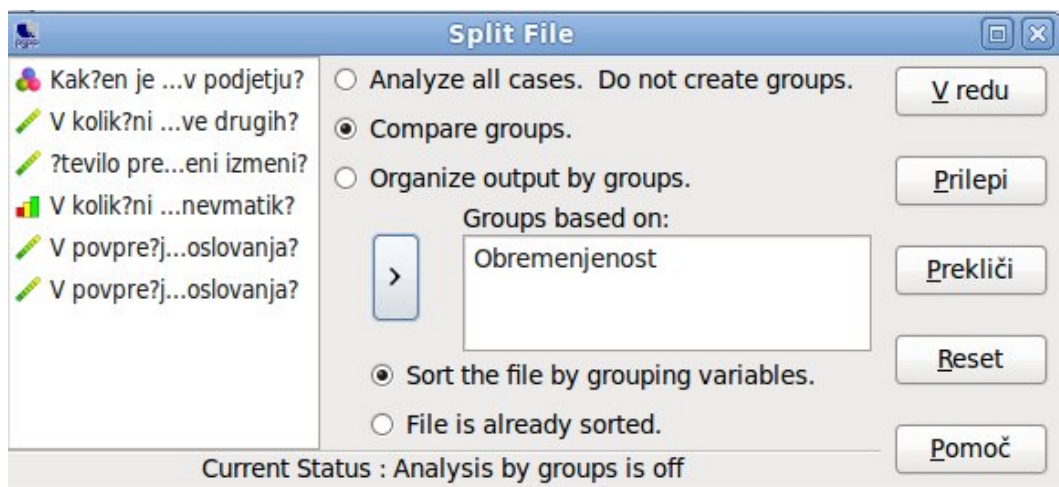
	Mean	N	Std. Deviation	S.E. Mean
Pair 0 Trajanje_pre vzema_prej	3.43	7	.53	.20
Trajanje_pre vzema_sedaj	1.86	7	.69	.26

	N	Correlation	Sig.
Pair 0 Trajanje_pre vzema_prej & Trajanje_pre vzema_sedaj	7	.19	.68

	Paired Differences					t	df	Sig. (2-tailed)
	Mean	Std. Deviation	Std. Error Mean	95% Confidence Interval of the Difference				
				Lower	Upper			
Pair 0 Trajanje_pre vzema_prej - Trajanje_pre vzema_sedaj	1.57	.79	.30	.84	2.30	5.28	6	.00

Obstaja statistično
signifikantna razlika

Slika 17.26: Rezultati T-testa za dva odvisna vzorca



Slika 17.27: Razdruževanje

Output1 --- PSPP Output

Windows Help

1.1(1) FREQUENCIES. Kakšen je vaš položaj v podjetju?

Value Label	Value	Frequency	Percent	Valid Percent
Vodja skladišča	1	3	42.86	42.86
Skladišnik	2	2	28.57	28.57
Pomočnik	3	2	28.57	28.57
Total		7	100.0	100.0

1.1(2:2) FREQUENCIES. Kakšen je vaš položaj v podjetju?

Cum Percent
42.86
71.43
100.00

N	Valid	Missing
7		0

Mean	1.86
Std Dev	.90
Minimum	1.00
Maximum	3.00

Slika 17.28: Rezultati razdruženih datotek

Povzetek

Statistična analiza podatkov je ena izmed pomembnih aktivnosti na vseh področjih raziskovanja. Pogosto jo uporabljamo, kadar želimo analizirati obsežnejšo količino zbranih podatkov. Poleg statističnih testov za preizkušanje hipotez s T-testi, analizo variance, neparametričnimi testi ipd., lahko v PSPP izvedemo linearno regresijo, faktorsko analizo in še mnogo več.

Pri uporabi programskega orodja je potrebno paziti predvsem na natančen vnos podatkov in določitev spremenljivk. Morda komu izmed bralcev testi, ki smo jih prikazali na primeru niso razumljeni, vendar ko se v praksi srečamo s tovrstno problematiko, zadeve postanejo povsem preproste. Ko je izdelana baza podatkov, lahko v kratkem času izvedemo in preizkušamo analize, ki nas zanimajo.

Na izbranem primeru izvedemo analizo podatkov zbranih z anketnimi vprašalniki. Zaposlene v namišljenem podjetju vprašamo o zadovoljstvu s postopkom prevzema pnevmatik. Pridobljeni rezultati niso relevantni, saj v vzorec zajemo zgolj nekaj zaposlenih. S programskim orodjem smo želeli prikazati, da tudi človeški faktor pomembno vpliva na procese dela v podjetju.

Poglavje 18

DOSTOP DO PROGRAMSKIH ORODIJ

Opomba: V stolpcu "Upravljalca paketov Synaptic" so navedene besede, ki jih je potrebno napisati v iskalniku programskih orodij v "Upravljalca paketov Synaptic" za želena programska orodja. Velja izključno za uporabnike operacijskega sistema Ubuntu (Tabela 18.1).

Programska orodja	Spletna povezava	Upravljalec paketov Synaptic
Ubuntu 10.04 LTS	http://www.ubuntu.com/desktop/get-ubuntu/download	/
WINE HQ 1.2	http://www.winehq.org/	WINE
Planner 0.14.4	http://live.gnome.org/Planner/Downloads	PLANNER
DIA 0.97.1	http://projects.gnome.org/dia/	DIA
Zint 2.3.2	http://www.zint.org.uk/zintSite/	/
ASDN 1.217	http://asdn.sourceforge.net/download.htm	/
Google Zemlja 5.1	http://earth.google.com/intl/sl/	GOOGLE EARTH
Quantum GIS 1.01	http://www.qgis.org/wiki/Download	/
QCAD 2.0.5 (Ubuntu) 2.2.2 (Windows)	http://www.qcad.org/qcad-downloads.html	QCAD
Simple Warehouse Mapper	http://www.wildmousesoftware.com/-productsmapsdownload.htm	/
Petersen 3.2.3	http://www.mathcove.net/petersen/lessons/-gettingPetersen?les=0	/
LINDO 6.1	http://www.lindo.com/index.php?option=com-contentview=articleid=34Itemid=14	/
DEXI 3.02	http://www-ai.ijs.si/MarkoBohanec/dexi.html	/
GnuCash 2.2.9	http://www.gnucash.org/	GNUCASH
GPSS World 5.2.2	http://www.minutemansoftware.com/downloads.-asp	/
Scilab 5.2.1	http://www.scilab.org/products/scilab/download	SCILAB
OoO Preglednica 3.2.0	http://sl.openoffice.org/	OoO CALC
PSPP 0.602	http://www.gnu.org/software/pspp/get.html	PSPP

Tabela 18.1: Dostop do programskih orodij

Poglavje 19

PREVODI

Analyze	Analiza
Abort	Prekinitev
Actions	Dejavnosti
Add Arrow	Dodaj smer
Add Node	Dodaj vozlišče
Add vector layer	Dodaj vektorski sloj
Adjacency List	Seznam sosednosti
Adjacency Matrix	Matrica sosednosti
Administration	Skrbništvo
Advance	Vnaprej
Air	Zračni transport
Align	Uskladitev
Appearance	Videz
Applications	Programi
Apply	Vstavi
Arrow	Puščica
Assets	Sredstva
Assets and Liabilities	Sredstva in obveznosti
ATO Assembly-to-Order	Skupinska izvedba
Attribute	Atribut
Automatically	Samodejno
AutoUpdate	Avtomatsko osveževanje
Background Color	Barva ozadja
Binomial	Binom
Bivariate Correlation	Bivariantna korelacija
Breadth First	Pregled grafa v širino
Brown methods	Brownova metoda eksponentnega glajenja
Business	Poslovanje
CAD Computer aided design	Računalniško podprto načrtovanje
Calc	Preglednica
Calculate	Izračun
Capacity-Selling	Prodajna količina
Capital tid per day	Vezani kapital (dan)
Cash Flow	Denarni tok
Cell	Celice
Change current directory	Spremeni direktorij
Change title	Sprememba naslova
Charts	Grafikoni
Chi-Squrare	Statistični hipotezni test
Chromatic Number	Kromatično število
Circuit Graph	Krožni graf
Clear History	Zbriši zgodovino
Client	Stranka

Tabela 19.1: Prevodi A-C

Close	Zapri
Color Scheme	Barvna shema
Command	Ukaz
Compare Means	Primerjava povprečij
Complete Bipartite Graph	Popolni dvostranski graf
Complete Tripartite Graph	Popolni tristranski graf
Continue	Naprej
Control	Kontrola
Copy	Kopiraj
Cost	Stroški
Create	Kreirati
Create Simulation	Kreiraj simulacijo
Crosstabs	Kontingenčne tabele
Cruscal algorithn	Kruskal algoritem
Current Version	Trenutna verzija
Custom	Po meri
Data	Podatek
Data View	Podatkovni pogled
Decimals	Decimalke
Decision rules	Odločitvena pravila
Demand	Potreba
Demand Rate x	Odstotek potreb
Depart	Odstopiti
Depth Firs	Pregled grafa v globino
Descriptive Statistics	Deskriptivna statistika
Discrete missing values	Diskretne manjkajoče vrednosti
Display labels	Prikaz oznak
Downloads	Prenos
Drawing Exchange	Izmenjava risb
Dual Price	Dualna cena
Duration	Trajanje
E-billing	E-plačevanje računov
E-buying	E-nakupi
E-wallet	E-denarnica
Edges	Robovi (poti)
Edit	Uredi
Edit Task	Uredi opravilo
Editor	Urejevalnik
End Value	Končna vrednost
Enter	Vhod
Equal	Enako
Equity	Lastniški kapital
ESRI	Vrsta oblikovnih podatkov
ETO Engineer-to-Order	Inženir-naročilo
Euler graph	Eulerjev graf

Tabela 19.2: Prevodi C-E

Evaluation	Vrednotenje
Execute	Izvedi
Exit	Izhod
Expenses	Odhodki
Export	Izvozi
Export format	Izhodni format
Facility	Objekt
Factor Analysis	Faktorska analiza
File	Datoteka
Financial Calculator	Finančni izračun
Financials	Finančni podatki
Find	Najdi
Finish to finish	Konec – konec
Finish to start	Konec – začetek
Flip graph	Obrniti graf
Flow chart	Diagram poteka
Fly simulation	Simulator letenja
Font	Pisava
Forecasting	Napovedovanje
Format	Oblika
Frequencies	Frekvenca
From - To	Od – do
Gain	Ojačevalnik
GDAL Geospatial Data Abstraction Library	Pretvornik rastrskih podatkov
General	Splošno
General Ledger	Glavna knjiga
Generate	Generirati
GIS Geographic information system	Geografski informacijski sistem
GML Geography Markup Language	Geografski označevalni jezik
GNU	Računalniški operacijski sistem (prosto programiranje)
Goods in Transport	Blago v transportu
Google Earth	Google Zemlja
Google Maps	Google zemljevidi
GPL General Public Licence	Splošna javna licenca
GPS Global Position System	Globalni pozicionirni sistem
GPSS General Purpose Simulation Systems	Simulacijski sistem za splošno rabo
GPX	Format za izmenjavo podatkov
Graph	Graf
Graph Size	Velikost grafa

Tabela 19.3: Prevodi E-G

Graph View	Pogled mreže
GRASS Geographic Resources Analysis Support System	GRASS Podporni sistem za grafično analizo
Ground	Cestni transport
Hamilton cycle	Hamiltonov cikel
Help	Pomoč
Highlight critical Tasks	Osvetlitev kritičnih opravil
Holding Cost	Stroški zalog
Holt methods	Holtova linearna metoda
How many vertices?	Koliko točk?
Ident Task	Dodajanje opravil (horizontalno)
Import	Uvoz
Income	Prihodek
Income and Expense	Prihodki in odhodki
Increment By	Prirastek
Input	Vložek
Input Adjacency Matrix	Vstavi matrico sosednosti
Input LCF Notation	Vstavi LCF zapis (predstavitev Hamiltonovih kubičnih grafov)
Insert Cases	Vstavi primer
Insert Subtask	Vstavi podopravilo
Insert Task	Vstavi opravilo
Insert Variable	Vstavi spremenljivko
Integer Programming	Celoštevilsko programiranje
Integrator	Računsko orodje za računanje z integrali
Interrupt	Napaka
Introduction	Uvod
Inventory Value	Vrednost zaloge
Inventory	Zaloge
Isomorphism	Izomorfizem
Label	Oznaka
Lag	Zakasnitev (časovna)
Layer	Sloji
Layout	Postavitev
Leave	Izhod (zapustiti)
Legend	Legenda
Letters	Črke
Liabilities	Obveznosti
Line Graph	Črta grafa
Linear Regression	Linearna regresija
Link Task	Poveži opravilo
Linux	Prost operacijski sistem
Load environment	Naloži okolje
Lot size calculator	Kalkulator proizvodnih serij

Tabela 19.4: Prevodi G-L

LTS Long Term Support software	Dolgoročna programska podpora
Mac OSX	Operacijski sistem podjetja Apple Inc.
Manage Calendars	Kolendar
Measure	Merilo
Metadata	Podatkovna baza
Minimal speening tree	Minimalno vpeto drevo
Missing	Manjkajoče
Model	Model
Mouse	Miška
Move Task Down	Premakni opravilo dol
Move Task Up	Premakni opravilo gor
MTO Make-to-Order	Naročilo
MTS Market to Stock	Na zalogo
Named Graph	Izberi vrsto grafa
Network	Mreža
New	Nov
New Project	Nov projekt
Node	Vozlišče
Node table view	Tabela vozlišč
Nominal	Nominalno
Notepad	Beležnica
Null Graph	Prosto izbirni graf
Number of orders is n	Število naročil n
Numbers	Številke
OMT (Object Model Technique)	Programska inženirska metoda (objektno orientiran razvoj analize in oblikovanja)
OOo OpenOffice.org	Odprtokodna aplikacija
OOSE (Object-Oriented Software Engineering)	Objektno usmerjen model programske opreme
Open	Odpri
Open Attribute Layer	Odpri atributni sloj
Open With	Odpri z/s
Operation	Operacija
Optimal Cycle is T^*	Optimalni krog T
Optimal Ordering is Q^*	Optimalno število naročil Q
Options	Možnost
Order Decouplin Point	Točka nevezanosti
Ordering Cost	Stroški naročanja
Ordinal	Ordinalno
Other	Drugo
Outline color	Barva skice
Outline style	Stil skice
Outline width	Širina skice
Output	Izhod (izdelek)

Tabela 19.5: Prevodi L-O

Page Down	Navzdol
Page setup	Nastavitev strani
Page Up	Navzgor
Panels	Plošče
Paste	Prilepi
Permissions	Dovoljenja
Physic	Fizika
Picture	Slika
Place	Kraj
Place	Mesta
Plugins	Vmesniki
Predecessors	Predhodniki
Price	Cena
Prims algorithym	Primov algoritem
Print	Tiskanje
Print Composer	Tiskanje
Print Preview	Predogled tiskanja
Prism	Prizma
Production Throughtput time	Čas proizvodnje
Prognosis	Prognoza
Programming	Programiranje
Project	Projekt
QIF Quicken Interchange Format	Odprta specifikacija za branje in pisanje finančnih podatkov
Queue	Vrsta
Rail	Železniški transport
Read Adjacency Matrix From Disk	Beri matrico iz diska
Read Graph From Disk	Beri graf iz diska
Redo	Naprej (ponovi)
Reduced Cost	Zmanjšani stroški
Refresh	Osveži
Relations	Relacija
Release	Sprostitev
Remove	Odstrani
Remove Task	Odstrani opravilo
Reports	Poročilo
Reset	Resetiranje
Resources	Viri
Rotate Graph	Vrtenje grafa
Save	Shrani
Save As	Shrani kot
Save environment	Shrani okolje
Scale	Lestvica
Scale	Zaloga vrednosti
Scenario	Scenarij

Tabela 19.6: Prevodi P-S

Science and Engineering	Znanost in Inženirstvo
Scilab Demonstration	Scilab demonstracija
SDTS Spatial Data Transfer Standard SDTS	Standard prenosa prostorskih podatkov
Search	Išči
Seize	Velikost grafa
Seller	Prodajalec
Sequence	Zaporedje
Setup	Nastavitve
Shadow Price	Senčna cena (popravljen tržna cena)
Shapefiles	Vrsta GIS podatkov (oblikovni podatki)
Sheets and Objects	Listi in predmeti
Ship	Ladijski transport
Shortest Path	Najkrajša pot
Slack	Ostanek
Solve	Reši
Spanning Trees	Vpeto drevo
Split File	Razdeli datoteko
Standard deviation OLT	Standardni odklon OLT
Start to finish	Začetek – konec
Start to start	Začetek – začetek
Start Value	Začetna vrednost
Statistics	Statistika
Storage	Skladišče
Subgraph	Podgraf
Sum	Seštevek
Surplus	Presežek
Symbology	Simbologija
Synaptic Package Manager	Upravljalnik paketov Synaptic
System	Sistem
Table	Tabela
Tabulate	Tabelirati
Taxes	Davki
Terminate	Prekiniti
Test	Test
The Classical Model	Klasični model
Toggle editing	Urejanje
Tools	Orodja
Total Cost is TC*	Celotni stroški TC
Total Inventory Value	Skupna vrednost zalog
Transaction Report	Transakcije
Transfer	Prenos

Tabela 19.7: Prevodi S-T

Transport table view	Transportna tabela
Tree diagram	Drevo diagrama
Tutorial	Navodila
Type	Tip
Ubuntu	Prosto dostopni operacijski sistem
Ubuntu Software Center	Programsko središče Ubuntu
UML (Unified Modeling Language)	Univerzalni jezik modeliranja
Undo	Nazaj
Unident Task	Odvzemanje opravil (horizontalno)
Unix	Računalniški operacijski sistem
Unlink Task	Prekini povezavo opravila
User Guide	Uporabniška navodila
Utilities	Uporabnost
Utility function	Funkcija koristnosti
Variable View	Pogled sprejemljivk
View	Pogled
Web links	Spletne povezave
Weight Cases	Utežitev primerov
Weighted Directed Edge	Utežitev direktnih robov (poti)
Weighted Edge	Utežitev robov grafa
What	Kaj
Whell	Kolo
Where	Kje
Width	Širina
WMS Warehouse Management System	Skladiščni informacijski sistem
Work	Delo
Workers	Zaposleni
Working - nonworking	Delovni – nedelovni čas
Write Graph To Disk	Shrani graf
Writer	Dokument z besedilom

Tabela 19.8: Prevodi T-W

Poglavje 20

PRILOGA

20.1 Scilab

By: Dejan Dragan [77]

20.1.1 Brownov model

```
// brown.m
=====

function brown(d, alfa)

fakt = 5;

N = length(d);

t = [1:1:N];

// izhodi prediktor-filtra (projekcija prihodnosti za en korak naprej - napoved
kolicine za cas t+1 ob casu t):
// (relevantno je k = 2:N)

p = [];
p(1) = d(1)
p(2) = d(1)

for k=3:N
    p(k) = alfa*d(k-1) + (1-alfa)*p(k-1)
end;

disp('p=')
disp(p)

p_pred = alfa*d(N) + (1-alfa)*p(N)

disp('p_pred=')
disp(p_pred)

// Narisemo zahtevano kolicino d in njeno napoved p:
scf(0)
plot(t,d,'b');
set(gca(), 'auto_clear', 'off')
plot(t,d,'bo')
plot([t(2:N) N+1],[p(2:N)' p_pred], 'r');
plot([t(2:N) N+1],[p(2:N)' p_pred], 'ro')
```

```

title('zahtev.kolic.d (b) ob casu t in njena napoved p (r), narejena ob casu
t-1')
mtlb_grid

// pogresek (kar se dejansko zgodi ob casu t minus kar smo napovedali ob casu
t-1):
e = d(2:N)-p(2:N)';

disp('e=')
disp(e)

scf(1)
plot(t,[0 e], 'k')
plot(t,[0 e], 'ko')
title('pogresek napovedi kolicine')
mtlb_grid
mtlb_axis([0 N -max(d)/fakt max(d)/fakt])

// izracun kriterij. funkcije MAD
ea = 0;
for i = 1:length(e)
    ea = ea + abs(e(i));
end
MAD_N = ea/(N-1)

disp('MAD_N=')
disp(MAD_N)

endfunction

```

20.1.2 Holtov model

```

// holt.m
=====

// (pri variab. linear. trendu)

function holt(d,alfa,beta,stlet)

N = length(d);

t = [1:1:N];

```

```

tao = [1:1:stlet]; // vektor premika projekcije v prihodnost od trenutka N naprej

//alfa = 0.3;      // konstanti holt prediktor filtra
//beta = 0.3;

[ah,bh] = holt_rekurzija(d,alfa,beta) // vrne ah(k) in bh(k) - AD1:prvo ju
"naucis" preko k=1:N, kaka sta optimalna

disp('ah=')
disp(ah)
disp('bh=')
disp(bh)

scf(0)
set(gca(),'auto_clear','off')
subplot(211)
plot(t,ah,'b')
plot(t,ah,'bo'),
mtlb_grid
title('holtov parameter ah(k)')

subplot(212)
plot(t,bh,'r')
plot(t,bh,'ro'),
mtlb_grid
title('holtov parameter bh(k)')

scf(1)

p = ah(N) + bh(N)*tao // p(N+tao) = ah(N) + bh(N)*tao, tao = 1,2,...; AD2:
nato ah(N) in bh(N) uporabis za predikc. pri t>N

disp('p=')
disp(p)

// Narisemo zahtevano kolicino d in njeno napoved p:

plot(t,d,'b');
plot(t,d,'bo')
plot(N+tao,p,'r');plot(N+tao,p,'ro')
mtlb_grid
title('Zahtevane kolicine (k = 1:N) in predikcija (k>N)')

endfunction

```

20.1.3 Regresijski model

```
// regresija.m
=====

function regresija(d,stlet)

N = length(d);

t = [1:1:N];
t_pred = [N+1:1:N+stlet]

scf(0)
set(gca(),'auto_clear','off')
plot(t,d,'b')
plot(t,d,'bo')
title('Povprasevanje')
xlabel('t')
mtlb_grid

t_s = sum(t)/N
d_s = sum(d)/N

disp('t_s=')
disp(t_s)
disp('d_s=')
disp(d_s)

mean_t_d = t*d'/N
mean_t_t = t*t'/N

disp('mean_t_d=')
disp(mean_t_d)
disp('mean_t_t=')
disp(mean_t_t)

clen1 = mean_t_d
clen2 = t_s*d_s
clen3 = mean_t_t
clen4 = t_s*t_s

disp('clen1=')
disp(clen1)
disp('clen2=')
disp(clen2)
disp('clen3=')
disp(clen3)
```

```
disp('clen4=')
disp(clen4)

a = (clen1 - clen2)/(clen3 - clen4)
b = d_s - a*t_s

disp('a=')
disp(a)
disp('b=')
disp(b)

d_o = a*t+b    // model

disp('d_o=')
disp(d_o)

scf(1)
plot(t,d,'b')
plot(t,d,'bo')
plot(t,d_o,'r')
plot(t,d_o,'ro')
title('Povprasevanje d(modro)in model d_m(rdece), predikcija+-2*std (crno)')
xlabel('t')
mtlb_grid

e = d - d_o

disp('e=')
disp(e)

e_sr = sum(e)/N

disp('e_sr=')
disp(e_sr)

VAR = (e-e_sr)*(e-e_sr)'/(N-1)

disp('VAR=')
disp(VAR)

stde = sqrt(VAR)

disp('stde=')
disp(stde)

d_o_pred = a*t_pred + b
d_o_pred_zg = d_o_pred + 2*stde
d_o_pred_sp = d_o_pred - 2*stde
```

```
disp('d_o_pred=')
disp(d_o_pred)
disp('d_o_pred_zg=')
disp(d_o_pred_zg)
disp('d_o_pred_sp=')
disp(d_o_pred_sp)

plot(t_pred,d_o_pred,'ko','linewidth',2)
plot(t_pred,d_o_pred_zg,'ko','linewidth',2)
plot(t_pred,d_o_pred_sp,'ko','linewidth',2)

endfunction
```

20.1.4 Funkcija napovedovanja (Forecast)

```
// forecast-main
=====

function forecast-main()

clear
clc

dch = input('Povprasevanje rocno (1), default(2), nakljucno (3)')
if dch == 1
    d = input('povprasevanje d= ? npr. [7 9 12 14 12 30 8 12 14 15 25 40]')
elseif dch == 2
    dd = input('primer1 (1), primer2 (2), primer3 (3), primer4 (4)')
    if dd == 1
        d = [77217 55120 52316 50859 59881 62697 60531 59885 67791 71037 57391]
        disp('Daj regresijo')
    elseif dd == 2
        d = [77217 55120 52316 50859 59881 62697 60531 59885 67791 71037 57391]
        disp('Daj regresijo')
    elseif dd == 3
        d = [77217 55120 52316 50859 59881 62697 60531 59885 67791 71037 57391]
        disp('Daj Brown')
    else
        d = [77217 55120 52316 50859 59881 62697 60531 59885 67791 71037 57391]
        disp('Daj Holt')
    end
end
```

```
else
    tt = input('Koliko vzorcev za d')
    d(1) = input('d(1)=')
    hh = input('konstanten trend(1)/linearen (2)')
    srvr = d(1)
    stres = d(1)/20
    d_rand = grand(1,tt,'unf',srvr-stres,srvr+stres)
    if hh == 1
        d = d_rand
    else
        strm = d(1)/10
        konst = d(1)
        d = strm*[1:1:tt] + konst + d_rand
    end
end

if dch == 3
    if hh == 1
        ch = 2;
        disp('Delamo Browna')
    else
        ch = input('regresija, mnk(1)/holt(3)')
    end
else
    ch = input('regresija, mnk(1)/brown(2)/holt(3)')
end

if ch == 1
    stlet = input('Za koliko let predikcija =')
elseif ch == 2
    alfa = input('alfa =')
else
    stlet = input('Za koliko let predikcija =')
    alfa = input('alfa =')
    beta = input('beta =')
end

disp('d=')
disp(d)

if ch == 1
    regresija(d,stlet)
elseif ch == 2
    brown(d,alfa)
else
    holt(d,alfa,beta,stlet)
end

endfunction
```

20.2 GPSS World

By: Borut Jereb [71]

20.2.1 Model

MODEL

=====

* Time is in minutes

* Initialization

*

```
GENERATE    ,,1
SAVEVALUE  TrafficLight,Green
ENTER      StoRim,StartNoRim
ENTER      StoTire,StartNoTire
ENTER      StoScrew,StartNoScrew
```

TERMINATE

*** Rim section begin *****

*

* Input to warehouse :: rims

*

```
GENERATE    InRimTimeMean,InRimTimeRange
QUEUE      QueueInWarehouse,1
TEST GE    R$StoRim,InCapVehRim
TEST E     X$TrafficLight,Green
SAVEVALUE  TrafficLight,Red
ENTER      StoRim,InCapVehRim
ADVANCE    InVehManipulRim,InVehManipulRimRange
SAVEVALUE  TrafficLight,Green
DEPART     QueueInWarehouse,1
```

TERMINATE

```

*
*   Output from warehouse :: rims
*

GENERATE      OutRimTimeMean,OutRimTimeRange
TABULATE      TableRim
TEST GE       S$StoRim,OutCapVehRim,RimStorageEmpty
LEAVE         StoRim,OutCapVehRim

TERMINATE

RimStorageEmpty TERMINATE
*
*
*** Rim section end *****

*** Tire section begin *****

*
*   Input to warehouse :: tire
*

GENERATE      InTireTimeMean,InTireTimeRange
QUEUE         QueueInWarehouse,1
TEST GE       R$StoTire,InCapVehTire
TEST E        X$TrafficLight,Green
SAVEVALUE     TrafficLight,Red
ENTER         StoTire,InCapVehTire
ADVANCE       InVehManipulTire,InVehManipulTireRange
SAVEVALUE     TrafficLight,Green
DEPART        QueueInWarehouse,1

TERMINATE

*
*   Output from warehouse :: tire
*

GENERATE      OutTireTimeMean,OutTireTimeRange
TABULATE      TableTire
TEST GE       S$StoTire,OutCapVehTire,TireStorageEmpty
LEAVE         StoTire,OutCapVehTire

TERMINATE

TireStorageEmpty TERMINATE
*
*
*** Tire section end *****

```

```

*** Screw section begin *****
*
*   Input to warehouse :: screw
*

GENERATE      InScrewTimeMean,InScrewTimeRange
QUEUE        QueueInWarehouse,1
TEST GE      R$StoScrew,InCapVehScrew
TEST E       X$TrafficLight,Green
SAVEVALUE    TrafficLight,Red
ENTER        StoScrew,InCapVehScrew
ADVANCE      InVehManipulScrew,InVehManipulScrewRange
SAVEVALUE    TrafficLight,Green
DEPART       QueueInWarehouse,1

TERMINATE

*
*   Output from warehouse :: screw
*

GENERATE      OutScrewTimeMean,OutScrewTimeRange
TABULATE     TableScrew
TEST GE      S$StoScrew,OutCapVehScrew,ScrewStorageEmpty
LEAVE        StoScrew,OutCapVehScrew

TERMINATE

ScrewStorageEmpty TERMINATE
*
*
*** Screw section end *****

*
*   Simulation duration
*

GENERATE      ,,SimDur,1 ;Duration in minutes
TERMINATE    1

START 1

```

20.2.2 Vhodni podatki

VHODNI PODATKI

=====

* Time is in minutes

* By: Borut Jereb

StoRim	STORAGE 800	; Warehouse storage capacity for rims
StoTire	STORAGE 800	; Warehouse storage capacity for tires
StoScrew	STORAGE 2600	; Warehouse storage capacity for screws

; Frequency distribution table for rims, tires and screws

TableRim	TABLE	S\$StoRim,19,19,100
TableTire	TABLE	S\$StoTire,19,19,100
TableScrew	TABLE	S\$StoScrew,79,79,100

; Initial number of items in the warehouse (at the start time of simulation)

StartNoRim	EQU	200
StartNoTire	EQU	200
StartNoScrew	EQU	800

; Mean arrival time for rims, tires and screws

InRimTimeMean	EQU	180
InTireTimeMean	EQU	360
InScrewTimeMean	EQU	170

; Arrival_time = (InXXTimeMean-InXXTimeRange..InXXTimeMean+InXXTimeRange)

; XX = {Rim, Tire, Screw}

InRimTimeRange	EQU	10
InTireTimeRange	EQU	20
InScrewTimeRange	EQU	10

; Capacity of one input vehicle carrying rims, tires and screws

InCapVehRim	EQU	390
InCapVehTire	EQU	800
InCapVehScrew	EQU	1450

; Vehicle upload manipulation time for rims, tires and screws

InVehManipulRim	EQU	6
InVehManipulTire	EQU	6
InVehManipulScrew	EQU	6

; Mean depart time for rims, tires and screws

OutRimTimeMean	EQU	8
OutTireTimeMean	EQU	8
OutScrewTimeMean	EQU	8


```
; Departure_time = (OutXXTimeMean-OutXXTimeRange..OutXXTimeMean+OutXXTimeRange)
; XX = {Rim, Tire, Screw}
OutRimTimeRange      EQU    1
OutTireTimeRange     EQU    1
OutScrewTimeRange    EQU    1

; Capacity of one input vehicle carrying rims, tires and screws
OutCapVehRim         EQU    20
OutCapVehTire        EQU    20
OutCapVehScrew       EQU    80

; unloading manipulation time for rims, tires and screws
InVehManipulRim      EQU    6
InVehManipulTire     EQU    6
InVehManipulScrew    EQU    6

; Unloading_time = unloading manipulation time for rims, tires and screws
InVehManipulRimRange EQU    0
InVehManipulTireRange EQU    0
InVehManipulScrewRange EQU    0

Green                 EQU    1      ; Boolean value for green is 1
Red                   EQU    0      ; Boolean value for red is 0

SimDur                EQU    7200   ; Simulation duration in minutes
```

Literatura

- [1] Agencija republike slovenije za okolje - arso. Najdeno 15. junija 2010 na spletnem naslovu <http://www.arso.gov.si>.
- [2] American digital cartography. Najdeno 23. junija 2010 na spletnem naslovu http://www.adci.com/html/oracle_navteq.php.
- [3] Apache http server project. Najdeno 7. julija 2010 na spletnem naslovu http://httpd.apache.org/dev/images/apache_logo.gif.
- [4] Asdn. Najdeno 5. junija 2010 na spletnem naslovu <http://asdn.sourceforge.net/>.
- [5] Dexi. Najdeno 4. junija 2010 na spletnem naslovu <http://www-ai.ijs.si/MarkoBohanec/dexi.html>.
- [6] Dia. Najdeno 7. junija 2010 na spletnem naslovu <http://live.gnome.org/Dia>.
- [7] Example business. Najdeno 21. julija 2010 na spletnem naslovu <http://www.austintek.com/gnucash/ncsa-gnucash-talk-4.html>.
- [8] Gant chart. Najdeno 7. junija 2010 na spletnem naslovu http://en.wikipedia.org/wiki/Gantt_chart.
- [9] Geografski informacijski sistem [rešitve za terenski zajem podatkov]. Najdeno 22. junija 2010 na spletnem naslovu <http://www.geoservis.si/uporabno/gis/gis.htm>.
- [10] Getting started with scilab. Najdeno 6. junija 2010 na spletnem naslovu http://www-irma.u-strasbg.fr/~sonnen/SCILAB_HELP/frame.html.
- [11] Gnucash. Najdeno 20. julija 2010 na spletnem naslovu <http://www.gnucash.org>.

- [12] Gnucash 2. Najdeno 20. julija 2010 na spletnem naslovu http://www.mojmikro.si/pod_lupo/programska_oprema/gnu_cash_2.
- [13] Gnucash 2.0. Najdeno 20. julija 2010 na spletnem naslovu <http://www.linux.com/archive/articles/114189>.
- [14] Gnucash 2.2.5. Najdeno 20. julija 2010 na spletnem naslovu <http://www.povejnaprej.org/gnucash-225/>.
- [15] Gnucash project goals. Najdeno 20. julija 2010 na spletnem naslovu <http://linas.org/linux/xacc/projects.html>.
- [16] Gnucash [putting all together (examples)]. Najdeno 21. julija 2010 na spletnem naslovu http://svn.gnucash.org/docs/guide/currency_examples1.html.
- [17] Google earth. Najdeno 9. junija 2010 na spletnem naslovu <http://earth.google.com/intl/sl/userguide/v5/>.
- [18] Gs1 [sledljivost]. Najdeno 24. junija 2010 na spletnem naslovu http://www.gs1si.org/doc/Traceability_v1_web2.pdf.
- [19] Gs1 [standard sledljivosti]. Najdeno 24. junija 2010 na spletnem naslovu http://www.gs1si.org/doc/Traceability_v1_web2.pdf.
- [20] If you knew cash like gnucash knows cash. Najdeno 21. julija 2010 na spletnem naslovu <http://www.linux-mag.com/id/6779>.
- [21] Institut oko. Najdeno 8. julija 2010 na spletnem naslovu <http://www.institut-ok.si/osfcd/html/linux.html>.
- [22] Java. Najdeno 12. junija 2010 na spletnem naslovu <http://java.sun.com>.
- [23] Jdexi. Najdeno 4. junija 2010 na spletnem naslovu <http://www-ai.ijs.si/MarkoBohanec/jdexi.html>.
- [24] Keep track of your money. Najdeno 21. julija 2010 na spletnem naslovu <http://www.linuxjournal.com/article/5669>.
- [25] Ko začnemo na stroške gledati z viška. Najdeno 20. julija 2010 na spletnem naslovu http://cekin.si/clanek/posojila_in_varcevanja/kontrolirajte-svoje-stroske.html.
- [26] Last. Najdeno 8. julija 2010 na spletnem naslovu www.last100.com/.../03/ubuntu-music-300x300.png.

- [27] Lindo. Najdeno 25. junija 2010 na spletnem naslovu <http://www.lindo.com>.
- [28] A lindo (dos) tutorial. Najdeno 25. junija 2010 na spletnem naslovu <http://www.txstate.edu/cs/course/3348/lp/lndtut.htm>.
- [29] Lindo: Linear, quadratic and integer programming software. Najdeno 25. junija 2010 na spletnem naslovu <http://www.udel.edu/topics/software/special/statmath/lindo/>.
- [30] Linux na namizju. Najdeno 17. avgusta 2010 na spletnem naslovu http://matej.owca.info/linux/Linux_na_namizju.pdf/.
- [31] Lugos [splošno dovoljenje gnu]. Najdeno 8. julija 2010 na spletnem naslovu <http://www.lugos.si/linux/licence/gpl#TOC3>.
- [32] Master cartons codes [itf-14 aurea media]. Najdeno 14. julija 2010 na spletnem naslovu http://www.aureamedia.com/mastercarton_info.html.
- [33] Math cove. Najdeno 18. junija 2010 na spletnem naslovu <http://www.mathcove.net/petersen/lessons/gettingPetersen?les=0>.
- [34] Minuteman software [computer simulation]. Najdeno 26. julija 2010 na spletnem naslovu <http://www.minutemansoftware.com/>.
- [35] Minuteman software [reference manual]. Najdeno 26. julija 2010 na spletnem naslovu <http://www.minutemansoftware.com/>.
- [36] Mozilla firefox. Najdeno 8. julija 2010 na spletnem naslovu online.keuka.edu/file.php/1/mozilla-firefox.jpg.
- [37] Nasvet [google earth, google maps, world wind in ostali digitalni zemljevidi]. Najdeno 9. junija 2010 na spletnem naslovu <http://www.nasvet.com/digitalni-zemljevidi/>.
- [38] Openoffice.org. Najdeno 17. junija 2010 na spletnem naslovu <http://sl.openoffice.org/izdelek/calc.html>.
- [39] Openoffice.org calc in pictures. Najdeno 17. junija 2010 na spletnem naslovu <http://inpics.net/calc.html>.
- [40] Pdfdatabase.com. Najdeno 25. junija 2010 na spletnem naslovu <http://pdfdatabase.com/search/tutorial-lindo-or.html>.

- [41] Planner. Najdeno 26. junija 2010 na spletnem naslovu <http://live.gnome.org/Planner/About>.
- [42] Psp. Najdeno 8. junija 2010 na spletnem naslovu <http://www.gnu.org/software/pspp/>.
- [43] Qcad. Najdeno 4. junija 2010 na spletnem naslovu <http://www.ribbonsoft.com/index.html>.
- [44] Quantum gis. Najdeno 15. junija 2010 na spletnem naslovu <http://www.qgis.org/>.
- [45] Ribonsoft [qcad]. Najdeno 4. junija 2010 na spletnem naslovu <http://www-ai.ijs.si/MarkoBohanec/jdexi.html>.
- [46] Scilab. Najdeno 5. junija 2010 na spletnem naslovu <http://www.scilab.org/>.
- [47] Scilab group. Najdeno 6. julija 2010 na spletnem naslovu http://elm.eeng.dcu.ie/~ee317/Matlab_Clones/manual.pdf.
- [48] Scilab manual. Najdeno 10. junija 2010 na spletnem naslovu http://gd.tuwien.ac.at/comp/scilab/manual_scilab-5.1.1_en_US.pdf.
- [49] Techeye.net. Najdeno 7. julija 2010 na spletnem naslovu <http://www.techeye.net/software/open-source-outfit-snubs-linux>.
- [50] Using gnuCash 2.0 to balance your checkbook. Najdeno 20. julija 2010 na spletnem naslovu <http://www.linux.com/archive/articles/58300>.
- [51] Using wine on ubuntu. Najdeno 17. avgusta 2010 na spletnem naslovu <http://www.psychocats.net/ubuntu/wine>.
- [52] Uvod v simulacijo. Najdeno 27. julija 2010 na spletnem naslovu http://www.e-studij.si/Uvod_v_simulacijo.
- [53] Večparametrski odločitveni model. Najdeno 4. junija 2010 na spletnem naslovu <http://www-ai.ijs.si/MarkoBohanec/org95/>.
- [54] Visualization [computer graphics]. Najdeno 29. junija 2010 na spletnem naslovu [http://en.wikipedia.org/wiki/Visualization_\(computer_graphics\)](http://en.wikipedia.org/wiki/Visualization_(computer_graphics)).
- [55] Webgps. Najdeno 27. julija 2010 na spletnem naslovu <http://www.webgps.com/index.php>.

- [56] What is gnuCash. Najdeno 20. julija 2010 na spletnem naslovu <http://www.gnuCash.org/docs/v2.0/C/gnuCash-help/what-is-gnuCash.html>.
- [57] Wikipedia [cad standards]. Najdeno 18. junija 2010 na spletnem naslovu http://en.wikipedia.org/wiki/CAD_standards.
- [58] Wikipedia [computer aided design]. Najdeno 4. junija 2010 na spletnem naslovu <http://www-ai.ijs.si/MarkoBohanec/jdexi.html>.
- [59] Wikipedia [gnuCash]. Najdeno 21. julija 2010 na spletnem naslovu <http://en.wikipedia.org/wiki/GnuCash>.
- [60] Wikipedia [logistika]. Najdeno 7. junija 2010 na spletnem naslovu <http://sl.wikipedia.org/wiki/Logistika>.
- [61] Wikipedia [odprtokodna programska oprema]. Najdeno 7. julija 2010 na spletnem naslovu http://sl.wikipedia.org/wiki/Odprtokodna_programska_oprema.
- [62] Wikipedia [simulation language]. Najdeno 26. julija 2010 na spletnem naslovu http://en.wikipedia.org/wiki/Simulation_language.
- [63] Wikipedia [traceability]. Najdeno 5. junija 2010 na spletnem naslovu <http://en.wikipedia.org/wiki/Traceability>.
- [64] Wikipedia [visualization (computer graphics)]. Najdeno 28. junija 2010 na spletnem naslovu [http://en.wikipedia.org/wiki/Visualization_\(computer_graphics\)](http://en.wikipedia.org/wiki/Visualization_(computer_graphics)).
- [65] Wikipedia [wine]. Najdeno 17. avgusta 2010 na spletnem naslovu [http://en.wikipedia.org/wiki/Wine_\(software\)](http://en.wikipedia.org/wiki/Wine_(software)).
- [66] Wild mouse software. Najdeno 28. junija 2010 na spletnem naslovu <http://www.wildmousesoftware.com/>.
- [67] Winehq. Najdeno 17. avgusta 2010 na spletnem naslovu <http://www.winehq.org/about/>.
- [68] Zint. Najdeno 24. junija 2010 na spletnem naslovu <http://www.zint.org.uk/>.
- [69] Chen A., Leptoukh G., and et. al. Vizualization of a-train vertical profiles using google earth. *Science Direct - Computers and Geosciences*, 35:419–427, February 2009.

- [70] Bergant B. Kratka zgodovina linearnega programiranja. Najdeno 25. junija 2010 na spletnem naslovu http://www.fmf.uni-lj.si/~juvan/Racunalnistvo3/0708/gradivo/Kratka_zgodovina_LP.pdf.
- [71] Jereb B. Neobjavljeno besedilo o zapisanih programih za programsko orodje gpss. Pridobljeno 2010.
- [72] Jereb B. *Informatika in računalništvo za vaje informatika v logistiki*. Abakus in Jereb, Celje, 2. dopolnjena izdaja edition, 2007. ISBN= 978-961-92263-0-8.
- [73] Jereb B. and Jancar R. *Simulations of discrete events based on GPSS*. Fakulteta za logistiko, Celje-Krško, 2009.
- [74] Peters B. A., Smith J. S., Medeiros D. J., Rohrer M. W., and eds. *GPSS - 40 YEARS OF DEVELOPMENT*. Department of Managerial Economics, Stockholm School of Economics, Stockholm, 2001.
- [75] Kobryn C. *UML 2001 A Standardization Odyssey*. Communication of the ACM, New York, oktober 1999.
- [76] Quang C. N. and Keat T. S. Google earth as a tool in 2-d hydrodynamic modeling. *Elsevier - Computers and Geosciences*, June 2010.
- [77] Dragan D. Neobjavljeno besedilo o zapisanih programih za programsko orodje scilab. Pridobljeno 2010.
- [78] Dragan D. *Optimizacija logističnih sistemov. Zbirka rešenih nalog*. Fakulteta za logistiko, Celje-Krško, 2008.
- [79] Skok D. *Zasnova sledenja blaga v podjetju Bosio d.o.o. : diplomsko delo univerzitetnega programa logistika sistemov*. Fakulteta za logistiko, Celje, 2009.
- [80] Jereb E., Bohanec M., and Rajkovič V. *Dexi - Računalniški program za večparametrsko odločanje*. Moderna organizacija, Kranj, 2003.
- [81] Rietsch E. An introduction to scilab from a matlab user's point of view. Najdeno 8. junija 2010 na spletnem naslovu http://web.mit.edu/scilab_v4.0/Scilab4Matlab2.6-1.0.pdf.
- [82] Urroz E. G. Programming with scilab. Najdeno 8. junija 2010 na spletnem naslovu <http://www.infoclearinghouse.com/files/scilab/scilab02.pdf>.

- [83] Farin G., Hoschek F., and Kim M. S. *Handbook of computer aided geometric design*. Elsevier Science B.V, North Holland, 2002.
- [84] Verč G. *Računalniško podprt večparametrski odločitveni model za ugotavljanja problemov izdelkov v razvojni fazi*. Magistrsko delo. Ekonomska fakulteta, Ljubljana, 2004.
- [85] Xiong G., Kitaygordskaya N., and Helo P. *Software Tutorial ASDN Logistics Analysis*. Vaasa, Finland, 18. julij. Najdeno 5. junija 2010 na spletnem naslovu [http://en.wikipedia.org/wiki/Wine_\(software\)](http://en.wikipedia.org/wiki/Wine_(software)).
- [86] GnuCash. *GnuCash Tutorial and Concept Guide [Help]*. GnuCash.org, 2009.
- [87] Maylor H. *Beyond the Gantt Chart: Project Management Moving on*. University of Bath Pergamon, 2001.
- [88] Vrečko I. Projektni pristop k organiziranju logističnih procesov v podjetju, 2. junij 2003. Najdeno 5. junija 2010 na spletnem naslovu <http://rcum.uni-mb.si/~pmi/slo/FORUM2003a.htm>.
- [89] Povh J. *Optimizacija logističnih procesov. Teorija grafov. Gradivo za predavanje*. Fakulteta za logistiko, Celje, 2009.
- [90] Usenik J. *Optimizacija logističnih procesov. Študijsko gradivo*. Fakulteta za logistiko, Celje, 2009.
- [91] Wilson J. M. A centenary appreciation. *Science Direct - European Journal of Operational Research*, 149:430–437, 2003.
- [92] Mckay K. and Black G. The evolution of a production planning system: A 10-year case study. *E-revir*, 2007.
- [93] Van Dorp K. Tracking and tracing: a structure for development and contemporary practices. *Logistics Information Management*, 15:24–33, 2002. Najdeno 5. junija 2010 na spletnem naslovu <http://www.emeraldinsight.com/journals.htm?articleid=852181&show=>.
- [94] LEOSS. *Zanimivosti iz sveta črtne kode in RFID. Drevo odločanja*. Interno gradivo podjetja LEOSS.
- [95] Bastič M. *Operacijske raziskave. Študijsko gradivo*. Fakulteta za logistiko, Celje, 2007.

- [96] Fetih M. *Razvoj programske rešitve za elektronski števec električne energije s pomočjo jezika za modeliranje UML*. Univerza v Ljubljani Ekonomska fakulteta, junij 2004. Najdeno 5. junija 2010 na spletnem naslovu www.cek.ef.uni-lj.si/magister/fetih349.pdf.
- [97] Indihar Štemberger M. and Groznik A. *Informatizacija poslovanja*. Ekonomska fakulteta, Ljubljana, 2007.
- [98] Makarovič M. *Priročnik programskega paketa SCILAB za uporabnika začetnika. Interno gradivo*. Politehnika Nova Gorica, Nova Gorica.
- [99] Martini M. and Šafarič B. *Označevanje in sledenje transportnih enot. Logistični priročnik*. EAN Slovenija, Ljubljana, 2001.
- [100] Mugerle M. *Model upravljanja z znanjem za omrežno pomoč strankam. Magistrsko delo*. Ekonomska fakulteta, Ljubljana, 2003.
- [101] Simoneti M. Finančno načrtovanje: 2005. Najdeno 10. julija 2010 na spletnem naslovu <http://www.pf.uni-lj.si/media/simoneti.marko.financno.nacrtovanje.ppt>.
- [102] Helo P. *Agile Supply-Demand Networks. An Open Source Tool for Industrial Network Design*. Vaasa, Finland, 25. maj 2005. Najdeno 5. junija 2010 na spletnem naslovu <http://www.eglo.info/?file=22>.
- [103] Kovačič R. Spletni priročnik: Računovodstvo. Najdeno 23. julija 2010 na spletnem naslovu <http://alea.dzs.si/dokumenti/dokument.asp?id=13>.
- [104] Smith R. D. Simulation article in encyclopedia of computer science. Najdeno 28. julija 2010 na spletnem naslovu <http://graph-srv.uni-mb.si/Ares/GPSS%20prirocnik.pdf>.
- [105] Annigeri S. *Scilab A Hands on Introduction*. College of Engineering and Technology, Hubli, 17. - 18. april 2004.
- [106] Annigeri S. *An Introduction to Scilab*. College of Engineering and Technology, Hubli, december 2009.
- [107] Steiniger S. and Hay G. Free and open source geographic information tools for landscape ecology. *E-revir*, 2009. Najdeno 18. junija 2010 na spletnem naslovu http://www.geo.uzh.ch/~sstein/manuscripts/sstein_freegitools_ecoinf2009.pdf.

-
- [108] Strajnar U. *Uporaba odprtokodnih in prostih programskih rešitev v zasebne in poslovne namene*. Univerza v ljubljani Ekonomska fakulteta, Ljubljana, november 2008.
- [109] Yamagishi Y., Yanaka H., and et. al. Visualization of geoscience data on google earth. *Science Direct - Computers and Geosciences*, 36:372–382, March 2009.
- [110] Laboratorij za računalniško grafiko in umetno inteligenco. Simulation language. Najdeno 28. julija 2010 na spletnem naslovu <http://graph-srv.uni-mb.si/Ares/GPSS%20prirocnik.pdf>.