

**UNIVERZA V LJUBLJANI**  
**FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO**

Anže Orehek

**VZDRŽEVANJE IN OPTIMIZACIJA CENIKOV PONUDNIKOV TELEFONSKIH STORITEV**

DIPLOMSKO DELO NA UNIVERZITETNEM ŠTUDIJU

MENTOR: PROF. DR. VILJAN MAHNIČ

Ljubljana, 2009



Št. naloge: 01542/2009

Datum: 15.01.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **ANŽE OREHEK**

Naslov: **VZDRŽEVANJE IN OPTIMIZACIJA CENIKOV PONUDNIKOV  
TELEFONSKIH STORITEV  
PRICE LISTS MAINTENANCE AND OPTIMIZATION FOR PHONE  
OPERATORS**

Vrsta naloge: Diplomsko delo univerzitetnega študija

Tematika naloge:

Proučite postopke določanja in optimizacije cen ponudnikov telefonskih storitev in realizirajte aplikacijo, ki bo omogočala avtomatsko izdelavo in vzdrževanje cenikov. Načrt aplikacije predstavite z ustreznimi UML diagrami in opišite funkcionalnost, ki bo na voljo končnemu uporabniku.

Mentor:

  
prof. dr. Viljan Mahnič



Dekan:

  
prof. dr. Franc Solina

# Kazalo

Povzetek.....	1
Ključni pojmi .....	1
Abstract.....	2
Keywords .....	2
1.    Uvod.....	3
1.1.    Opis problemske domene .....	4
1.2.    Preslikava rešitve v problemsko domeno .....	4
1.3.    Cilj naloge .....	4
2.    Analiza problema s teoretičnim pristopom .....	5
2.1.    Najbolj pogosti cilji določanja cene.....	5
2.1.1.    Konkurenčnost .....	5
2.1.2.    Prestiž .....	5
2.1.3.    Dobiček.....	5
2.1.4.    Obseg prodaje .....	6
2.2.    Strategije določanja cen.....	6
2.2.1.    Penetracija.....	6
2.2.2.    “Skimming” .....	6
2.2.3.    Prilagajanje spreminjanju cen konkurence .....	7
2.3.    Analiza cene .....	8
2.3.1.    Vrste prodajnih cen .....	8
2.3.2.    Tržno oblikovanje cene .....	9
2.4.    Metode določanja cene.....	10
2.4.1.    Določitev cene na podlagi dejanskih stroškov s pribitkom .....	10
2.4.1.1.    Prednosti metode .....	12
2.4.1.2.    Slabosti metode .....	12
2.4.2.    Določanje cen na podlagi vrednosti, ki jo predstavlja kupcu .....	12
2.4.3.    Psihološko določanje cen.....	13
2.5.    Vpliv cene na dobiček .....	13
2.5.1.    Analiza.....	13
3.    Analiza problema s praktičnim pristopom .....	15
3.1.    Definicija problema.....	15
3.2.    Rešitev problema .....	15
3.2.1.    Konceptualen opis spletne aplikacije .....	16
3.3.    Predstavljena funkcionalnost.....	19
3.3.1.    Medoperaterski oziroma nabavni ceniki .....	19
3.3.2.    Prodajni ceniki .....	19
3.4.    Predstavitev osnovnih pojmov in funkcij.....	19
3.4.1.    Prefiks.....	19
3.4.2.    Cenik .....	19

3.4.3.	Prioritetni sistem .....	19
3.4.4.	Tabela orig destination .....	20
3.4.5.	Tabela master destination.....	20
3.4.6.	Tabela template destination .....	20
3.4.6.1.	Karakteristike .....	21
3.4.7.	Uvoz medoperaterskih cenikov.....	21
3.4.7.1.	Karakteristike .....	21
3.4.8.	Kreiranje prodajnega cenika .....	21
3.4.8.1.	Izračun prodajne cene .....	21
3.4.8.2.	Izračun osnovne cene .....	22
3.4.8.3.	Karakteristike .....	22
3.5.	Delitev cenika .....	23
3.5.1.	Splošni cenik.....	23
3.5.2.	Akcijski cenik .....	23
4.	Načrt (UML diagrami).....	24
4.1.	Diagram primerov uporabe.....	24
4.1.1.	Definicija .....	24
4.1.2.	Diagram primerov uporabe za uvoz nabavnega cenika.....	25
4.1.2.1.	Opis toka dogodkov za primer uporabe Uvozi cenik .....	25
4.1.2.2.	Opis toka dogodkov za primer uporabe Uvozi cenik ročno .....	26
4.1.3.	Diagram primerov uporabe za kreiranje prodajnega cenika.....	27
4.1.3.1.	Opis toka dogodkov za primer uporabe Kreiranje prodajnega cenika .....	27
4.1.3.2.	Opis toka dogodkov za primer uporabe Kreiranje prodajnega cenika preko "business margin" dialoga.....	27
4.1.3.3.	Opis toka dogodkov za primer uporabe Shranjevanje prodajnega cenika.....	28
4.1.3.4.	Opis toka dogodkov za primer uporabe Dodajanje marž glede na operaterja ..	28
4.2.	Paketni diagram .....	28
4.3.	Diagram zaporedja .....	30
4.4.	Komponentni diagram.....	31
4.5.	Arhitekturni diagram.....	32
5.	Izvedba .....	34
5.1.	Opis nabavnega cenika .....	34
5.2.	Opis predloge za uvoz nabavnega cenika.....	34
5.2.1.	Pravila.....	35
5.2.1.1.	Tipi pravil .....	35
5.2.1.1.1.	Distinct.....	35
5.2.1.1.2.	Split.....	36
5.2.1.1.3.	RemoveEmpty.....	36
5.2.1.1.4.	Substring .....	37
5.2.1.1.5.	Contain.....	37
5.2.1.1.6.	Replace.....	37

5.2.1.1.7.	Trim .....	38
5.2.1.1.8.	Format.....	38
5.2.1.1.9.	FormatDate.....	39
5.2.1.1.9.1.	Atributi.....	39
5.2.1.1.9.2.	Podelementi.....	40
5.2.2.	Elementi.....	40
5.2.2.1.	SOURCES .....	40
5.2.2.2.	SOURCE.....	41
5.2.2.2.1.	Atributi.....	41
5.2.2.2.2.	Pravilo.....	42
5.2.3.	Primer strukture XML konfiguracijske datoteke .....	43
5.3.	Opis izhodne, ChargeXML datoteke.....	43
5.3.1.	Struktura .....	43
5.3.1.1.	Elementi.....	43
6.	Grafični prikaz rešitve.....	45
6.1.	Postopek uvoza nabavnega cenika.....	45
6.2.	Uvoz nabavnega cenika ročno .....	52
6.3.	Postopek kreiranja prodajnega cenika.....	52
6.4.	Postopek kreiranja prodajnega cenika preko dialoga za upravljalno maržo .....	55
6.5.	Postopek dodajanja marže .....	57
7.	Sklepne ugotovitve .....	59
8.	Dodatek A .....	60
8.1.	Priloge .....	60
8.1.1.	Priloga 1: Nabavni cenik v Word formatu.....	60
8.1.2.	Priloga 2: Del nabavnega cenika v Excel formatu .....	60
8.1.3.	Priloga 3: Osnovna struktura XML predloge .....	62
8.1.4.	Priloga 4: Osnovna struktura ChargeXML datoteke.....	63
9.	Literatura.....	65

## Kazalo slik

SLIKA 1: SLAPOVNA ANALIZA CENE.....	9
SLIKA 2: DOLOČITEV PRODAJNE CENE.....	10
SLIKA 3: POVEČANJE DOBIČKA OB 1% DVIGU CENE OZIROMA 1% ZMANJŠANJU STROŠKOV ....	14
SLIKA 4: KONCEPTUALNA SHEMA APLIKACIJE .....	16
SLIKA 5: TOKOVI APLIKACIJE.....	16
SLIKA 6: INTERAKCIJA MED ODJEMALCEM, APLIKACIJSKIM IN PODATKOVNIM STREŽNIKOM	18
SLIKA 7: GRAFIČNI PRIKAZ PRIORITETNEGA DOLOČANJA TRENUTNIH CEN.....	20
SLIKA 8: DIAGRAM PRIMERA UPORABE ZA UVOZ NABAVNEGA CENIKA.....	25
SLIKA 9: DIAGRAM PRIMERA UPORABE ZA KREIRANJE PRODAJNEGA CENIKA.....	27
SLIKA 10: PAKETNI DIAGRAM.....	29
SLIKA 11: PAKETNI DIAGRAM PRICEMANAGER APLIKACIJE .....	29
SLIKA 12: DIAGRAM ZAPOREDJA ZA UVOZ NABAVNIH CENIKOV.....	30
SLIKA 13: DIAGRAM ZAPOREDJA ZA KREIRANJE PRODAJNIH CENIKOV.....	31
SLIKA 14: KOMPONENTNI DIAGRAM.....	32
SLIKA 15: ARHITEKTURNI DIAGRAM.....	33
SLIKA 16: IZBIRA IKONE .....	45
SLIKA 17: POLJA ZA VNOS PARAMETROV NABAVNEGA CENIKA .....	45
SLIKA 18: VHODNI CENIK JE USPEŠNO PRESLIKAN PREKO XML PREDLOGE.....	46
SLIKA 19: NAPAKA - VHODNI CENIK JE NEKONSISTENTEN .....	47
SLIKA 20: DIALOG NOVIH PREFIKSOV.....	47
SLIKA 21: V VHODNEM CENIKU SO NAJDENE NOVE DESTINACIJE.....	48
SLIKA 22: DIALOG NOVIH DESTINACIJ.....	49
SLIKA 23: CENIK S TEM IMENOM ŽE OBSTAJA .....	50
SLIKA 24: CENIK JE USPEŠNO UVOŽEN .....	51
SLIKA 25: PRIKAZ UVOŽENEGA NABAVNEGA CENIKA.....	52
SLIKA 26: DIALOG ZA ROČNI VNOS NABAVNEGA CENIKA .....	52
SLIKA 27: LOKACIJA IKONE ZA KREIRANJE PRODAJNEGA CENIKA .....	53
SLIKA 28: DIALOG ZA KREIRANJE PRODAJNEGA CENIKA.....	53
SLIKA 29: DIALOG NAPREDKA POSTOPKA KREIRANJA CENIKA .....	54
SLIKA 30: KREIRAN PRODAJNI CENIK .....	55
SLIKA 31: LOKACIJA DIALOGA ZA DODAJANJE UPRAVLJALSKE MARŽE.....	55
SLIKA 32: DIALOG ZA DODAJANJE UPRAVLJALSKE MARŽE.....	56
SLIKA 33: PRODAJNI CENIK Z DODANO UPRAVLJALSKO MARŽO.....	56
SLIKA 34: LOKACIJA DIALOGA ZA DODAJANJE MARŽE .....	57
SLIKA 35: DIALOG ZA DODAJANJE MARŽE.....	58

## Kazalo uporabljenih kratic

Kratica	Opis
COM	Component Object Model
CORBA	Common Object Request Broker Architecture
ER	Entity Relationship
EU	Evropska Unija
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IIS	Internet Information Services
OMG	Object Management Group
PTT	Pošta Telegram Telefon
TCP/IP	Transmission Control Protocol / Internet Protocol
UML	Unified Modeling Language
XML	eXtensible Markup Language

## **Povzetek**

Glavni cilj diplomske naloge je izdelava aplikacije za upravljanje s ceniki in optimizacijo cene, namenjene telekomunikacijskim operaterjem za delno avtomatizirano upravljanje s prodajnimi in nabavnimi cenami. Problemskemu področju sem poskušal dodati teoretično ozadje, ki je predstavljeno v uvodnih poglavjih.

Aplikacijo smo razvili glede na zahteve naročnika. Podpira uvoz in generiranje cenikov. Na eni strani govorim o nabavnih cenikih, ki jih uvažamo v aplikacijo, na drugi strani o prodajnih, katerih cene se generirajo na podlagi aktivne usmerjevalne tabele.

Nabavni cenik je prisoten v aplikaciji, ko je postopek uvoza cenika uspešno izveden. Cenik je urejen v Excel oziroma Word formatu. Posamezni ceniki se od operaterja do operaterja močno razlikujejo, zato vsak operater potrebuje nabor XML predlog, preko katerih lahko vhodni cenik pretvorijo v standardiziran format cenika, uporabljenega s strani aplikacije. Če je predloga pravilno napisana, lahko vhodni cenik možno pretvorimo v standardiziran cenik, ki ga, ob uspešno zaključenem postopku, shranimo v podatkovno bazo aplikacije.

Seveda lahko tudi prodajni cenik uvozimo v aplikacijo, vendar je to prej izjema kot pravilo. Te vrste cenik je v večini primerov generiran s strani aplikacije na podlagi internih podatkov aplikacije.

Aplikacija delno avtomatizira postopke, ki smo jih pred tem opravljali ročno. Na enem mestu so zbrani vsi nabavni in prodajni ceniki, ki smo jih uvozili oziroma generirali preko aplikacije. Možen je vpogled v zgodovino cen. Prodajne cene, ki so bile prej izračunane ročno ali z drugimi orodji (npr. Excel), je sedaj možno veliko hitreje in bolj natančno izračunati.

S pomočjo aplikacije lahko hitreje sledimo spremembam na telekomunikacijskem trgu. Pomembna pridobitev pa je tudi točnost. Pri ročnem računanju je točnost pogosto vprašljiva, saj vmes poseže človeški faktor in pomanjkanje podatkov za dovolj točen rezultat.

Pri opisu aplikacije sem si pomagal z UML diagrami primerov uporabe, paketnimi diagrami, diagrami zaporedja, komponentnimi in arhitekturnimi diagrami. S tem sem hotel bolj nazorno predstaviti samo zgradbo aplikacije.

## **Ključni pojmi**

aplikacija za upravljanje s ceniki in optimizacijo cene, telekomunikacije, prodajni cenik, nabavni cenik, UML

## **Abstract**

The main goal of my diploma thesis was to describe the application dealing with pricelists, which is used by common carriers and telephone companies who use them for partial automatization of controlling the selling and purchase prices. This field was dealt with and added a theoretic support in the first chapters of the paper.

It was me as well who cooperated in the progress of the development. The application was based upon the orders of the contracting entity. It supports import and generation of pricelists. One part consists of purchase pricelists which are imported into the application while the other part is represented by sale pricelists the prices of which are generated on the basis of the active routing table.

Purchase pricelist is found in the application only after a successful import of the pricelist. Purchase pricelist is given in Excel or Word file. Individual pricelists differ a lot from one to another telecommunication company. This is why every company needs to obtain its own import template through which the entering pricelist can be transformed into a standard format of the pricelists the application uses. If the template is written correctly, the entering pricelist can be converted into the standard form of the pricelist, which is stored into the database of the application after a successful transfer.

Selling pricelist can be imported into the application as well, but this would be just an exemption from a rule. This type of pricelist is mostly generated by the application on the basis of application's internal information.

The application partially automatizes the processes that have beforehand been done manually. All the purchase and selling pricelists that were imported and/or generated through the application are now stored in one place. An insight into the history of prices is possible as well. Selling prices, which were thus far been calculated manually or in a different tool (Excel), can now be calculated much faster and more precise.

Through this application the changes on the field of telecommunication can be monitored rapidly and precise. When calculating manually, precision is arguable as the presence of the human factor and lack of data might cause an arguable result.

What I used to describe and exemplify the usage of the application were UML diagrams: use case diagrams, package diagrams, sequence diagrams, component diagrams, and deployment diagrams. Through them, the sole construction of the application was shown.

## **Keywords**

application dealing with pricelists, telecommunication, selling pricelist, purchase pricelist, UML

## 1. Uvod

Kako čim boljše tržiti svoj izdelek oziroma storitev? Poleg kakovosti izdelka oziroma storitve, igra pri tem pomembno vlogo prav gotovo cena.

V času monopola na telekomunikacijskem trgu se s tem problemom ni nihče posebej ukvarjal, saj se niti ni bilo treba. Nacionalni operaterji so poljubno določali cene. Do preobrata je prišlo leta 1998, ko se je Evropska unija (EU) odločila za deregulacijo in liberalizacijo telekomunikacijskega trga. V Sloveniji smo to spremembo doživeli z letom 2001. Že pred tem je Slovenija izpeljala dve pomembni reorganizaciji, ki sta bili pogoj za liberalizacijo panoge [7]:

- avgusta 1993 je vzpostavila organ upravljanja – upravne funkcije, ki so jih do tedaj izvajale PTT organizacije, je prenesla na državo,
- leta 1995 je formalno ločila poštno in telekomunikacijsko dejavnost – ustanovila je Pošto Slovenije d.o.o. in reorganizirala preostala PTT podjetja v Telekom Slovenije d.o.o.

Nacionalni telekomunikacijski operaterji so s tem izgubili svoj monopolni položaj, saj jim je konkurenca odvzela del naročnikov in prihodkov. Izgubljeno so pridobili z vstopom na tuje telekomunikacijske trge, kjer so se soočili s tamkajšnjimi nekdanjimi monopolisti, ki so obvladovali domači trg in edini imeli dostop do naročnikov. Gradnja lokalne zanke za novega ponudnika ni bila ekonomsko upravičena, zato je bila ureditev medomrežnega povezovanja bistvenega pomena za vzpostavitev telekomunikacijskega tržnega okolja. Glavni namen vsega tega je, da naročnik opravi klic ne glede na to, v katerem omrežju je klicani naročnik.

Medomrežno povezovanje lahko različni regulatorji in različni režimi definirajo drugače. Eden od uradnih virov pravi, da je medomrežno povezovanje fizična in logična povezava javnih elektronskih komunikacijskih omrežij, ki jih uporablja isto ali drugo podjetje z namenom, da omogoči uporabnikom enega podjetja komunikacijo z uporabniki istega oziroma drugega podjetja ali dostop do storitev drugega podjetja. Storitve lahko zagotavljajo vpletene ali druge stranke, ki imajo dostop do omrežja [8]. Medomrežna povezava je torej posebna vrsta dostopa, ki je realizirana med operaterji javnih omrežij.

Z liberalizacijo trga v zadnjem desetletju, so postali učinkoviti sporazumi o medomrežnem povezovanju ključni za razvoj vse večjega števila storitev. Te storitve vsebujejo lokalne, krajevne in mednarodne fiksne, mobilne in satelitske storitve, ki nudijo vse od osnovne govorne telefonije do širokopasovne povezave v medmrežje za medmrežne multimedijske storitve. Konkurenca je ključna za rast in inovativnost današnjega telekomunikacijskega trga. V državah, kjer je raven konkurence visoka, je opazna večja produktivnost, boljša kakovost in nižje cene telekomunikacijskih storitev [1]. Medomrežno povezovanje je kritični faktor za obstoj konkurence. O pojmu konkurence, brez vključevanja v medomrežno povezovanje, pravzaprav sploh ne moremo govoriti.

Telekomunikacijski trg danes v večini držav velja za enega najbolj inovativnih, konkurenčnih in donosnih. Zaradi vse večje globalizacije in konkurenčnosti pa je seveda cena storitve najbolj pomemben dejavnik na trgu.

### ***1.1. Opis problemske domene***

Kako se soočiti s konkurenco in nenehnimi spremembami na telekomunikacijskem trgu?  
Kakšna naj bo cena storitve, da kljub močni konkurenci preživimo?

Sprva so operaterji naročali razne strateške analize in si s tem delali visoke stroške. Problem teh analiz je, da so statične, zato v trenutku, ko so izdelane, že zastarajo. Ob vse večji globalizaciji pa to predstavlja še toliko večjo težavo.

Zaradi vseh zgoraj naštetih lastnosti trga, je treba na spremembe v njem reagirati v trenutku, zato statična strateška analiza ni več dovolj dobra. Potrebovali bi bolj dinamičen pristop, ki nam zagotavlja hitre odzive na spremembe na trgu. Na tej točki so primerne aplikacije za upravljanje s ceniki in optimizacijo cen, ki zagotavljajo dinamičen pristop in hitre odzive na spremembe na trgu. Tovrstni postopki so se sprva opravljali ročno, vendar danes, v času množične globalizacije trga, to enostavno ni več mogoče, ker bi bilo zamudno in premalo natančno. Aplikacije za upravljanje s ceniki in optimizacijo cene bi ročne procese avtomatizirale, s čimer bi pridobili na času in točnosti izračunov.

### ***1.2. Preslikava rešitve v problemsko domeno***

Aplikacija zagotavlja učinkovito in pregledno upravljanje s ceniki. Poleg tega pa zagotavlja tudi natančno določanje prodajne cene. Ena od možnosti uporabe je "KAJ-ČE" analiza. V nadaljevanju se bom osredotočil predvsem na upravljanje s prodajnimi in nabavnimi ceniki, njihovim uvažanjem in uporabo le-teh.

### ***1.3. Cilj naloge***

Z nalogo želim izdelati in predstaviti del sistema za upravljanje s ceniki in to podpreti s teoretičnim ozadjem področja. Treba se je zavedati, da je to le ena od možnih rešitev. Za razvoj na tem področju še vedno obstaja premalo dokumentov in študij oziroma le-te niso dostopne širši javnosti. Podjetja, ki se ukvarjajo z aplikacijami te vrste, te težave rešujejo po svoje, glede na izkušnje, znanje oziroma zahteve naročnikov.

V nadaljevanju sledi poglavje, kjer bom poskušal umestiti problem določanja cene v širše teoretično ozadje. Temu sledi poglavje, kjer bo to teoretično ozadje preslikano na konkretni problem. Poskušal ge bom čim bolj približati bralcu. Sledijo poglavja z UML diagrami, kjer bom s to metodo opisal predstavljeni del sistema. Zaključna poglavja pa bodo namenjena predstavitvi uporabe opisanega dela sistema s slikovnim materialom in primeri vhodnih in izhodnih dokumentov, ki se pri določenem procesu uporabljajo oziroma generirajo.

## 2. Analiza problema s teoretičnim pristopom

V poglavju se bom osredotočil na splošno teorijo določanja cene, sestavo cene in različne strategije določanja cene.

Kot je bilo nakazano že v samem uvodu, se podjetja čedalje pogosteje sprašujejo kako vrednotiti produkt, ki ga nudijo. Kaj želimo s prodajo izdelka sploh doseči? Kriti stroške? Priti do dobička?

Cena, ki jo nosi izdelek, je ena od najbolj pomembnih poslovnih odločitev. Pravilna določitev prodajne cene je najhitrejši in najbolj uspešen način, da podjetje doseže kar največji dobiček. Pravilno določena cena lahko poveča dobiček hitreje, kot bi to dosegli s povečanjem obsega poslovanja. Določitev previsoke oziroma prenizke cene lahko v najboljšem primeru vpliva na omejitve širitve trga, med tem ko lahko v najslabšem primeru resno vpliva na prodajo ter denarne tokove. To je še posebej pomembno za podjetje, ki šele začenja s poslovanjem.

### 2.1. Najbolj pogosti cilji določanja cene

#### 2.1.1. Konkurenčnost

Primerjanje cene s ceno konkurenčnega produkta na trgu.

#### 2.1.2. Prestiž

Cena produkta je postavljena tako visoko, da omeji dosegljivost večini ljudi. Čeprav podjetje zaradi tega pričakuje omejeno količino prodaje, je prihodek še vedno velik zaradi velikega deleža marže.

Primer: Mercedes-Benz in Rolls-Royce

#### 2.1.3. Dobiček

Poudarek je na maksimiziranju dobička. Osnovna formula za dosego tega cilja je (1).

$$\text{Dobiček} = \text{Prihodek} - \text{Stroški} \quad (1)$$

Prihodek je določen s ceno produkta in obsegom prodaje. Treba je paziti, da cena ni previsoka, saj bi se s tem zelo verjetno drastično zmanjšal obseg, kar bi pomenilo tudi manjši dobiček.

Rešitev je spremljanje cen svojih produktov glede na konkurenco in istočasno zagotavljanje ustrezne marže.

### 2.1.4. Obseg prodaje

Podjetje, ki uporablja ta pristop, določi nekoliko nižjo ceno svojih izdelkov in storitev, kot je to običajno, s tem da pričakuje, da bo razliko nadoknadilo z večjim obsegom. Podjetje tako okrepi svoje ime in zaupanje strank, saj se njihov produkt trži v velikem obsegu in po ugodni ceni.

Podcilj je povečanje tržnega deleža, kar glede na ceno in velik obseg ne bi smelo predstavljati težav.

Sprejemljiva marža je določena s pridobitvijo ustreznega tržnega deleža in ceno produkta.

## 2.2. *Strategije določanja cen*

Podjetja lahko izbirajo med različnimi strategijami. Vsaka od njih pripelje do različnih ciljev, hkrati pa vsaka od teh podjetje oskrbi z dobičkom.

### 2.2.1. Penetracija

Gre za strategijo predstavitve novega produkta na trgu. Strategija je primerna za podjetja, ki bi po hitrem postopku rada pridobila tržni delež, ter za podjetja, ki želijo pridobiti dobiček z naročili. Ta strategija je lahko zelo učinkovita, če jo pravilno uporabijo.

Obstajata dva načina uveljavljanja strategije:

- Podjetje potencialne kupce oskrbi z zastoj primerki svojih izdelkov ter jim, v primeru nakupa, ponudi boljšo oziroma nižjo ceno.
- Alternativa zgornjemu primeru je, da podjetje izdelek ponudi po znižani ceni takoj, nato pa popust postopoma zmanjšuje. Čez določen čas nas zmanjševanje popustov pripelje do polne cene produkta.

Ideja te strategije je, da se seznanjanje kupcev s produktom ne zaračuna po polni ceni. Ko je kupec sčasoma "udomačen", pa se mu nakup produkta lahko zaračuna po polni ceni.

### 2.2.2. "Skimming"

Večfazna strategija generiranja dobička. Nasprotje penetracije.

1. faza

- Ob prihodu na trg se podjetje osredotoči na kupce, ki so pripravljeni za njihove izdelke in storitve plačati visoko ceno.
- Marža je v začetnem obdobju zelo visoka, kar prinaša podjetju večino prihodka.

- Ko podjetje spozna, da so v tej fazi izčrpali možnosti za povečanje tržnega deleža, preide v naslednjo.

## 2. faza

- V tej fazi cena pade, kar privede do povečanja obsega prodaje in čez čas večjega tržnega deleža.
- Ko se tudi v tej fazi povpraševanje umiri, podjetje ponovno preide v naslednjo fazo, kjer se cena še zniža.
- Ukrep zajame tiste kupce, za katere je bila v predhodnih fazah cena previsoka, kar privede do ponovnega povečanja obsega prodaje in posledično večjega tržnega deleža.

Podjetje ima s tem pokrito večino strank na trgu, kjer so za njihov product pripravljene plačati visoko, srednjo in nizko ceno. Strategija poskrbi, da je prihodek že na začetku visok in v primeru zmanjšanja povpraševanja znižuje ceno, tako da podjetje s povečevanjem obsega ostaja na približno enakih prihodkih. Strategija je zelo uporabna v primeru, ko že na začetku potrebujemo visoke prihodke, da s tem krijemo stroške raziskav in razvoja.

### 2.2.3. Prilagajanje spreminjanju cen konkurence

Konkurenčna podjetja lahko v določenem trenutku zvišajo ali znižajo ceno svojega produkta.

Preden se na spremembo odzovemo, je treba opraviti temeljito analizo. V večini primerov se pokaže, da na dvig/spust cen vplivajo:

- Dvig cen surovin
- Večji stroški zaposlovanja
- Višji davki
- Višja inflacija

Za vzdrževanje sprejemljive marže se v takem primeru cena zviša.

Ko konkurenca zviša ceno, imamo na izbiro več možnosti:

- Zvišati ceno na približno isto raven, kot jo je določila konkurenca.
- Zvišati ceno z zakasnitvijo → *price shadowing*.
  - Privablja nove kupce, tiste, ki so občutljivi na visoke cene. Če je rezultat tega večje število strank, se to odraža v obsegu prodaje. Večji obseg prodaje pa prinaša višji prihodek.
  - Če do tega prehoda kupcev ne pride, se cena lahko zviša.
  - V večini primerov se cena zviša na raven, ki je nekoliko nižja od cene konkurence.

Ko konkurenca zniža ceno, imamo prav tako več možnosti:

- Vzdržujemo trenutno ceno, saj verjamemo v zvestobo svojih kupcev, ki nam zaupajo in cenijo edinstveno kvaliteto naših izdelkov. V primeru, da so stranke občutljive na spremembo cen, to ne bi bila pravilna odločitev.
- Analiza znižanja cene konkurence. Če je vzrok tehnološka inovacija, potem bomo morali spustiti ceno tudi mi, saj bo spust cene konkurence v tem primeru verjetno permanenten.
- Spustimo cene ne glede na akcijo konkurence → *price covering*

- Akcija je upravičena, če je naš izdelek oziroma storitev po kakovosti primerljiv ali slabši od konkurenčnega izdelka oziroma storitve.
- Akcija ni upravičena, če je izdelek oziroma storitev boljši od konkurenčnega izdelka oziroma storitve.

Strategija se pogosto uporablja v letalski in telekomunikacijski industriji.

### **2.3. Analiza cene**

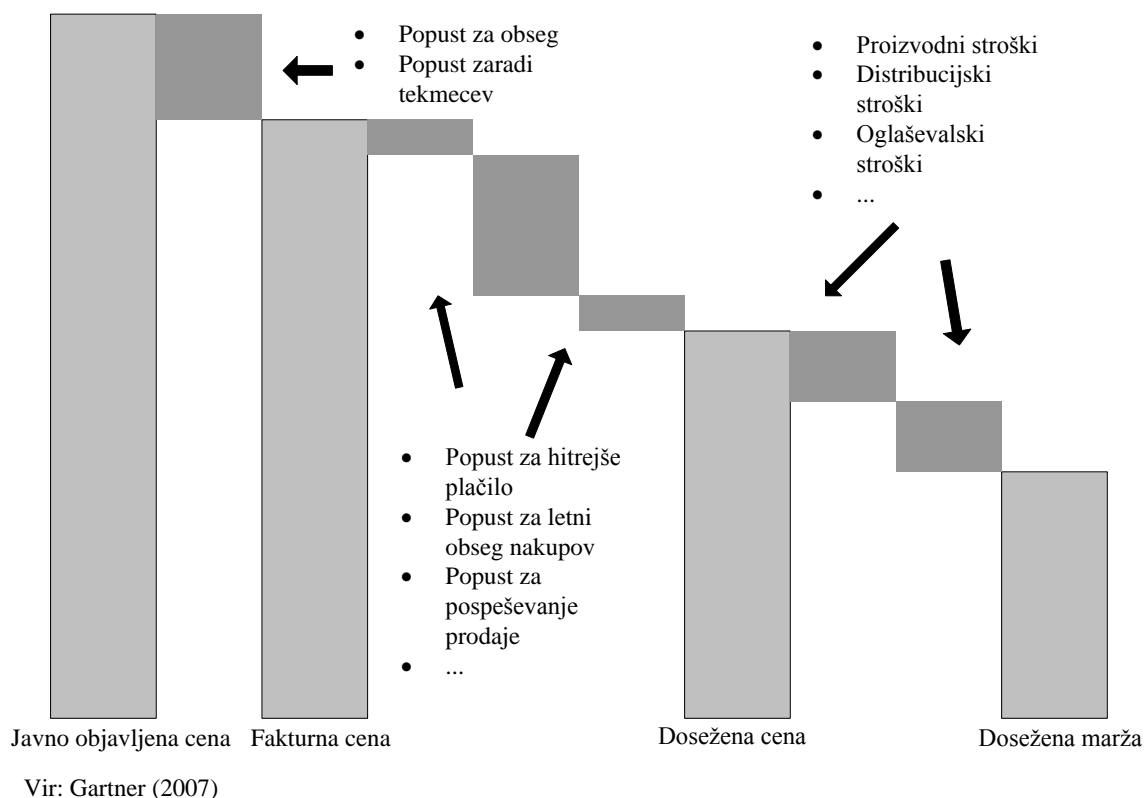
Podjetja pri določanju prodajnih cen največkrat izhajajo iz stroškov, ki so jih ali bodo imela z ustvarjanjem in prodajo svojih produktov. Pri oceni verjetnih pričakovanih stroškov je možno izhajati iz:

- Dejanskih stroškov iz preteklosti
- Standardnih stroškov
- Ocene prihodnjih stroškov

Ob tem pa seveda ne težijo le k cilju kritja stroškov s prodajno ceno, ampak želijo v prodajni ceni doseči tudi dobiček. Druga podlaga in hkrati omejitev za določitev višje prodajne cene pa so seveda tržne razmere.

#### **2.3.1. Vrste prodajnih cen**

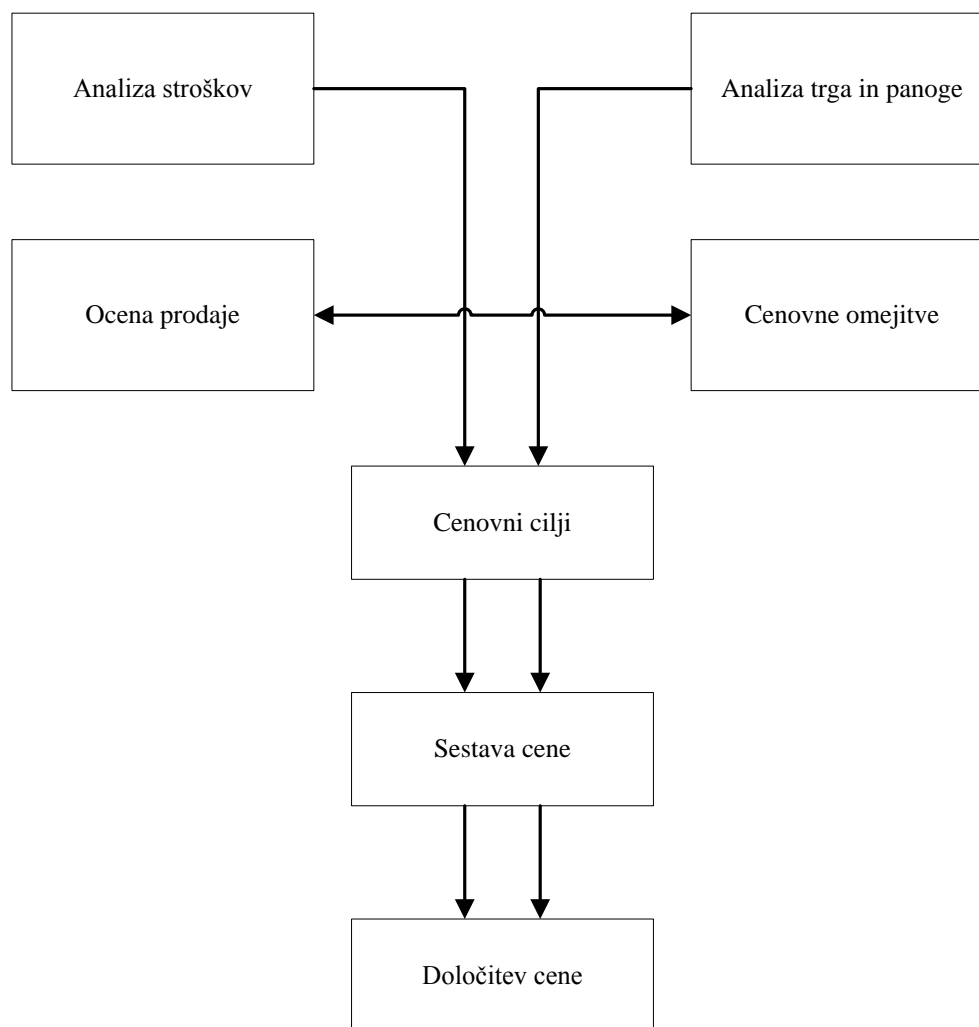
Od prodajne cene, ki jo podjetje javno objavi (*catalogue price*), je treba običajno najprej odšteti popust za obseg in popust zaradi tekmecev. Rezultat tega je fakturna ali izhodiščna cena (*invoice price*). Sledijo lahko še popusti za hitrejše plačilo, popust za letni obseg nakupov, popust za pospeševanje prodaje, popust za skupno oglaševanje prodaje in popust na težo, kar nas vodi do dosežene cene (*pocket price*). Celotna shema nivojev cen je lepo prikazana na sliki 1. Dosežena cena, v primerjavi s stroški, in ne fakturna cena, je prava mera cenovne privlačnosti produkta.



**Slika 1: Slapovna analiza cene**

### 2.3.2. Tržno oblikovanje cene

Dosežene prodajne cene na trgu seveda ne more vedno narekovati ena sama stran, ampak je ta cena posledica razmerij v ponudbi določenega produkta in povpraševanja po njem. V odvisnosti od primera do primera se oblikovalec cene, v okviru cenovne politike, ne sprašuje le o stroških, ampak tudi o verjetnem obsegu prodaje, vrstah prodajnih poti, pogojih plačila, elastičnosti povpraševanja, ovirah za vstop v panogo in drugih omejitvah. Vse to, in še drugi dejavniki ponudbe, nastopa kot osnova za določitev cene. Glej sliko 2. [5].



Vir: M.Mihelčič, Ekonomika poslovanja za inženirje (2005)

**Slika 2: Določitev prodajne cene**

## 2.4. Metode določanja cene

Kot je bilo povedano v enem od predhodnih poglavij, za določanje cene ne obstaja samo ena pot, ampk jih je več. Pri tem pa seveda ni nujno, da je vsaka primerna za vsakogar.

### 2.4.1. Določitev cene na podlagi dejanskih stroškov s pribitkom

Kot že sam naslov pove, gre pri tej metodi za računanje cene na podlagi stroškov in marže, s čimer ustvarjamo dobiček. Ta metoda je še vedno ena najbolj pogosto uporabljenih. Je preprosta in pri izračunu zahteva malo podatkov. Dobiček se lahko meri relativno ali absolutno. Ena od formul za računanje cene, bi bila formula (2), kjer se upoštevajo povprečni spremenljivi stroški, povprečni fiksni stroški ter odstotek marže.

$$C = (PSS + PFS) * (1 + M\%) \quad (2)$$

Kjer je

- $C$  = cena
- PSS = povprečni spremenljivi stroški
- PFS = povprečni fiksni stroški
- $M\check{Z}\%$  = odstotek marže

Primer uporabe formule (2). Imamo:

30€ spremenljivih stroškov

10€ fiksnih stroškov

50% maržo

-----

60€ končna cena produkta

$$P = (30 + 10) * (1 + 0.50)$$

$$P = 40 * 1.5$$

$$P = 60$$

Primer prikazuje, da je prodajna cena za 50% višja od celotnih stroškov.

Alternativa formuli (2) je formula (3).

$$C = (PSS + PFS) / (1 - M\check{Z}\%) \quad (3)$$

Primer uporabe formule (3):

30€ spremenljivih stroškov

10€ fiksnih stroškov

50% maržo

-----

80€ končna cena produkta

$$P = (30 + 10) / (1 - 0.50)$$

$$P = 40 / 0.5$$

$$P = 80$$

Stroški znašajo 50% prodajne cene.

Da bi že tako enostavne izračune še poenostavili, nekatera podjetja, še posebej maloprodajna, ignorirajo fiksne stroške in za stroške računajo samo nabavno ceno produkta. Fiksne stroške pa vključijo v maržo. Da se izračuni še bolj poenostavijo, se v veliko primerih za stroške uporablja absolutna vrednost in ne relativna. Te vrednosti določijo vodilni v podjetju. Takšen način določanja cen se imenuje določanje cen po ključu (turnkey pricing).

Druga različica te metode je določanje cen glede na aktivnost (activity based pricing). Tu se stroški določijo bolj previdno in bolj natančno kot v prvem primeru. Vsaka aktivnost je vezana na vir porabe.

#### **2.4.1.1. Prednosti metode**

Ceno lahko po tej metodi zelo preprosto izračunamo, saj je za izračun treba uporabiti minimalno število podatkov. Metoda deluje kot stabilizator trga, izoliran od vpliva povpraševanja in konkurence. Prodajalce varuje pred nepričakovanimi in nepredvidljivimi stroški, ki bi se lahko pojavili kasneje.

#### **2.4.1.2. Slabosti metode**

Metoda ne prinaša nobene vzpodbude za učinkovitost. Ignorira vlogo kupcev, konkurence in oportunitetnih stroškov. Uporabljajo se pretekli dejanski stroški, namesto standardni stroški. Izvajalec se pri tej metodi ne osredotoči na performance, saj je strošek vedno pokrit s strani klienta.

### **2.4.2. Določanje cen na podlagi vrednosti, ki jo predstavlja kupcu**

Ta metoda določa prodajno ceno na podlagi vrednosti, ki jo zaznava kupec, in ne na podlagi dejanskih stroškov proizvodnje, tržne cene ali konkurence. Njen cilj je, da cena odraža vrednost produkta kupcu. Če kupcu izdelek ali storitev veliko pomeni, bo podjetje zanj seveda lažje zaračunalo več. Cena ni ista za vse kupce, ampak se prilagaja potrebam in vrednosti, ki jo produkt kupcu zagotavlja. Strategija omogoča podjetjem večjo tekmovalnost in večjo donosnost kot bolj običajne in poenostavljene metode. V ta namen je dobro najprej opraviti analizo vrednotenja produktov pri kupcih. Pri tem so zelo uporabne aplikacije s področja podatkovnega rudarjenja, saj pomagajo segmentirati kupce in definirati, kakšno vrednost ima produkt za njih.

Primer: farmacevtsko podjetje je s to strategijo določilo ceno zdravila proti čiru na želodcu in ne s strategijo določanja cene glede na stroške izdelave in distribucije. S to metodo je podjetje lahko določilo višjo ceno kot bi jo, če bi zaračunali ceno, ki bi jo ponudile zdravstvene zavarovalnice.

S tem bi se radi izognili določanju cen nad mejo, ki jo lahko kupci sploh plačajo, vendar se kljub temu prepričali, da cena ni prenizka. Problem tradicionalnih metod je v tem, da ne prinašajo informacije o tem, kakšno vrednost ima produkt za kupca, zato se lahko zgodi, da je denar vržen stran.

Seveda se bi večina kupcev nad to strategijo pritoževala, saj bi to razumeli kot prevaro.

Verjetno so dober primer cene letalskih vozovnic oziroma hotelov. Zgodi se, da so le-te v določenem obdobju izjemno visoke. Predpostavimo, da Slovenija gosti olimpijske igre. Vse hotelske kapacitete bi bile polno zasedene. Kar bi pomenilo, da bi bila cena v tem času temu primerno višja.

### 2.4.3. Psihološko določanje cen

Psihološko določanje cen je upravljalna praksa, ki temelji na teoriji, da imajo nekatere cene psihološki vpliv. Maloprodajne cene so velikokrat poimenovane kot čudne cene, saj so v večini primerov nezaokrožene: 19.99€, 9.96€. Prav te čudne cene, pa večajo povpraševanje.

Študija, objavljena leta 1997 v glasilu *Marketing Bulletin* [3], ugotavlja, da se 60% cen v oglaševalskem materialu konča z 9, 30% s 5, 7% z 0 in 3% z ostalimi sedmimi števki.

Strategija temelji na naslednjih hipotezah:

Kupec raje ignorira številke na zadnjih decimalnih mestih, kot da bi ceno zaokrožil. Hipoteza ima še večjo težo v primeru, če so centi napisani z malimi številkami 19<sup>99</sup>€.

Kupec misli, da so cene napisane z decimalkami, najnižje možne.

Sedaj, ko so kupci tega formata takšnih cen že vajeni, pa zaokrožene cene delujejo čudno.

Prav tako je takšno določanje cen primerno, kadar imamo postavke cen razvrščene po območjih. Na podlagi tega nas nezaokrožena cena vrže v nižje območje, kjer je večja verjetnost, da bo produkt kdo opazil, čeprav je razlika samo en cent.

Naj kot zanimivost navedem rezultate še ene raziskave. Kot primer za opazovanje so uporabili margarino. Ceno so z 89 centov znižali na 71 centov. Prodaja margarine se je pri tem povečala za 65%. Vendar, ko je bila cena iste margarine znižana z 89 centov na 69 centov, se je prodajala povečala kar za 222% [10].

## 2.5. Vpliv cene na dobiček

### 2.5.1. Analiza

80% verjetnost je, da bo imela leta 2009 tehnologija optimizacije cene največji vpliv na izboljšanje bruto prihodkov in dobičkonosnost poslovanja [2].

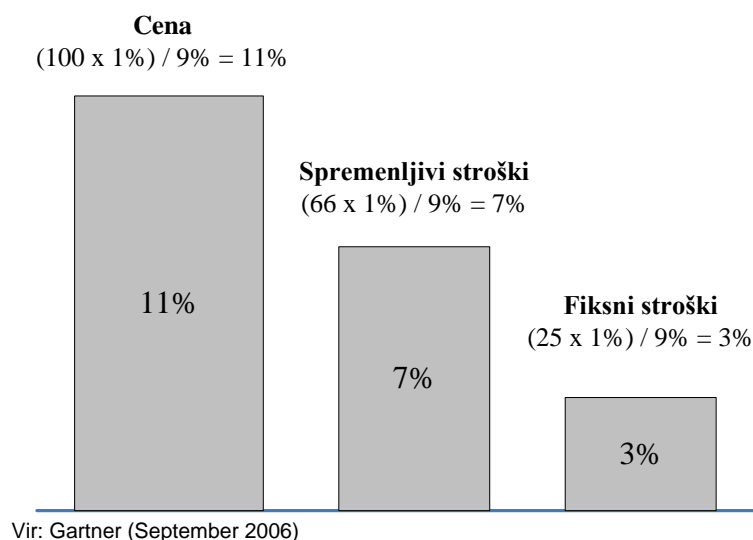
Najboljši način, da se to uresniči, je sprememba cen. Če predpostavljamo, da so stroški konstantni, potem vsako zvišanje cene avtomatsko poveča dobiček. To je idealen primer. Realen primer je bil opisan že v prejšnjih poglavjih. V realnem primeru lahko povečanje cene na določenih trgih zmanjša prodajo ali poveča možnost, da konkurenca poveča tržni delež. Izziv ni v tem, da maksimiziramo cene preko vseh tržnih segmentov, kot sta recimo produkt in geografska lega. Bistvo je v tem, da se podrobno preučijo posamezne ponudbe in opcije ter se določi strategije, občutljive na edinstvene tržne razmere. Rezultat bi pokazal zvišanje cene v nekaterih segmentih in zmanjšanje le-te v drugih.

Študija, objavljena v knjigi [4], prikazuje moč optimiziranja cen za povečanje dobička. V raziskavo so vključili 1200 najboljših javnih podjetij. V študiji so primerjali vpliv dviga cene oziroma zmanjšanja stroškov, za enako odstotno stopnjo, na dobiček.

Ob predpostavki, da so podjetja v povprečju imela:

- 25% fiksnih stroškov,
- 66% spremenljivih stroškov in
- 9% maržo

so rezultati študije sledeči. Kot je razvidno s slike 3, ima zvišanje cene za 1% večji vpliv na dobiček kot zmanjšanje stroškov za 1%. Zadosten razlog, da je v prihodnosti smiselno investirati v programsko opremo za upravljanje in optimizacijo cen.



### Slika 3: Povečanje dobička ob 1% dvigu cene oziroma 1% zmanjšanju stroškov

Optimizacija cene je koristna za odkrivanje cenovnih priložnosti in potencialnih problemov, še preden se ti dejansko pojavijo. Razlikuje se od analize cene, kjer imamo zgodovinski pregled. Pri optimizaciji gre za napovedovanje. Podjetja s tovrstno programsko opremo objavljajo izboljšane rezultate poslovanja. Računajo, da se je v povprečju dvignila marža za 10% ter dobiček za 15% in več [2].

Podjetja se za ta korak odločajo izredno redko. Razlogi za to so sledeči:

- Diskretnost podjetij, ki so to že uspešno realizirala
- Uspešne zgodbe niso javno predstavljene
- Pomanjkanje uporabnih primerov, zgledov
- Pomanjkanje strokovnjakov z ustreznim znanjem
- Politika določanja cen
- Dejstvo, da je treba v proces vključiti vse oddelke v podjetju (prodaja, uprava, proizvodnja in druge).

### 3. Analiza problema s praktičnim pristopom

V tem poglavju se bom osredotočil na predstavitev problema v praksi in ga skušal povezati s teorijo, opisano v prejšnjem poglavju.

#### 3.1. Definicija problema

Postavimo se v vlogo operaterja na globalnem telekomunikacijskem trgu. Če je operater prisoten na tako velikem trgu, kot je danes telekomunikacijski trg, ima opravka z velikim številom drugih operaterjev. Vsak operater ima svoj cenik, vsak cenik pa ima ogromno destinacij in še veliko več prefiksov. Danes tega definitivno ni več mogoče upravljati ročno, ampak potrebujemo programsko rešitev, s katero bi bilo to delo delno avtomatizirano in zato precej hitrejše. Naloga je torej razviti aplikacijo za upravljanje s ceniki in optimizacijo cene. Namenjena bo uvažanju nabavnih in generiranju prodajnih cenikov. Seveda so v ozadju še druge funkcije, njihov ključni namen pa je podpora teh dveh glavnih funkcij.

#### 3.2. Rešitev problema

Rešitev podpira:

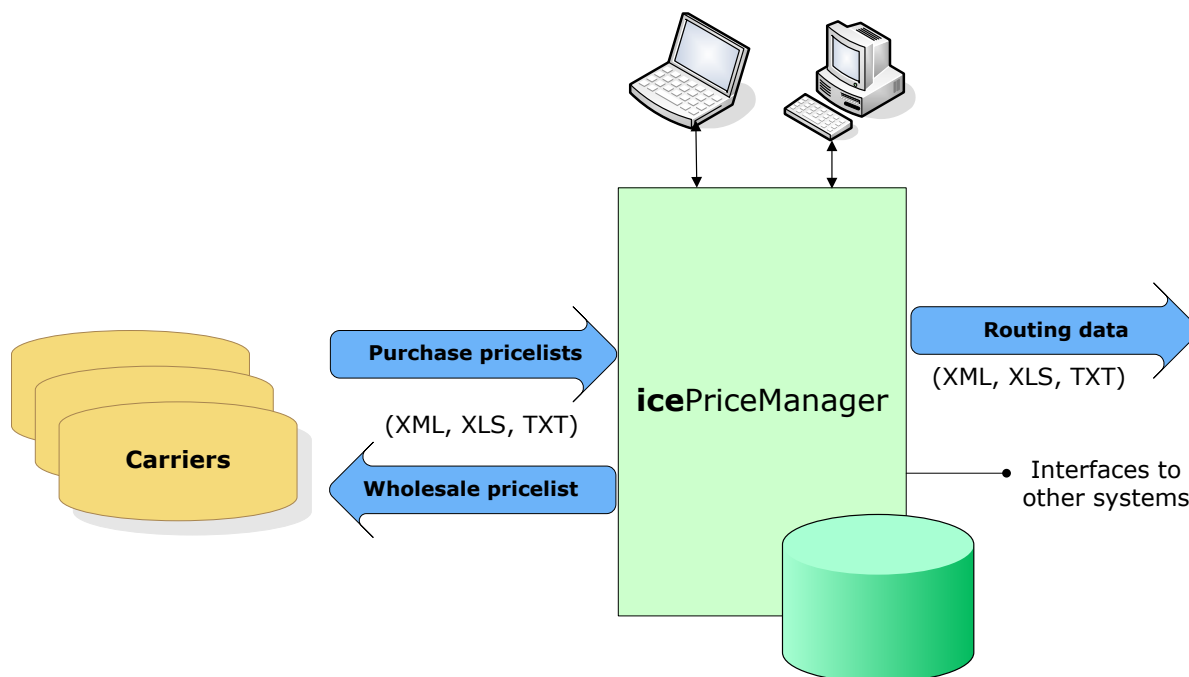
- Registracijo cenikov
- Upravljanje destinacij in prefiksov
- Pripravo prodajnega cenika
- ...

Kot prikazuje slika 4, je rešitev spletna aplikacija za upravljanje s ceniki, ki na vhodu dobiva nabavne cenike in na izhodu vrača prodajne cenike ter optimizirane routing podatke<sup>1</sup>. Operater, ki poseduje aplikacijo (v nadaljevanju podjetje XY), dobiva od operaterjev, preko katerih terminira klic<sup>2</sup>, nabavne cenike, kjer so nabavne cene. Vsota cen klicev za podjetje XY predstavlja strošek. Na drugi strani pa XY vsem operaterjem posreduje svoj prodajni cenik, kjer so definirane prodajne cene za določeno destinacijo preko operaterja/podjetja XY. Vsota prodajnih cen klicev za XY predstavlja prihodek. Predstavljeni tokovi so pregledno prikazani na sliki 5.

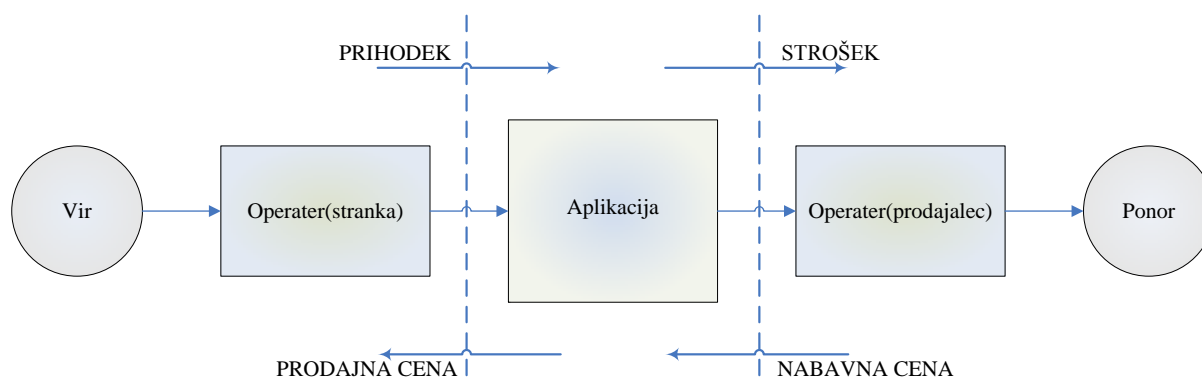
---

<sup>1</sup> Routing podatki oziroma routing tabela je tabela najkrajših poti skozi določeno točko omrežja. Routing tabela vsebuje operaterje ter pripadajoče optimizirane cene.

<sup>2</sup> Terminiranje klica oziroma zaključek klica izvede operater, ki je zadnji v verigi. Nasprotje terminiranju je originacija.



**Slika 4: Konceptualna shema aplikacije**



**Slika 5: Tokovi aplikacije**

### 3.2.1. Konceptualen opis spletne aplikacije

Spletne aplikacije temeljijo na arhitekturi odjemalec/strežnik. Posebnost te vrste aplikacij je komunikacija na osnovi zahtev in odgovorov. Tako strežnik ni sposoben ničesar poslati brez odjemalčeve zahteve.

Za realizacijo tovrstnih aplikacij lahko uporabimo arhitekturo z različnim številom nivojev.

Dvonivojska arhitektura je določena z odjemalcem in podatkovnim strežnikom. Vsa poslovna logika se izvaja na strani odjemalca, strežnik pa skrbi za podatke. Zelo neugodne lastnosti te arhitekture so:

- Obremenjevanje odjemalčevih sistemskih virov
- Težavno vzdrževanje programske opreme

- Zahtevnejša implementacija varnostnih mehanizmov

Nadgradnja osnovne dvonivojske arhitekture je trinivojska arhitektura, ki sem jo uporabil tudi v našem primeru. Trinivojska arhitektura je sestavljena iz predstavitvenega, vmesnega in podatkovnega nivoja. Predstavitveni nivo generira pogled za odjemalca, vmesni nivo implementira poslovno logiko, podatkovni nivo pa skrbi za podatke. Ta pristop odpravlja pomanjkljivosti dvonivojske arhitekture. Aplikacije je enostavno vzdrževati, pridobimo centraliziran dostop do podatkov ter izvajanje logike na strežniku.

Nadgradnja trinivojske arhitekture je N nivojska arhitektura, ki se od trinivojske razlikuje samo v dodatno razdeljenem vmesnem nivoju.

Prednosti N nivojske arhitekture:

- poenostavi porazdelitev aplikacij in virov
- omogoča uporabo različnih varnostnih mehanizmov
- omogoča uporabo naprednega upravljanja z bremenom
- zagotavlja razpoložljivost ob izrednih dogodkih

Kot je bilo nakazano že zgoraj, se v našem primeru uporablja trinivojska arhitektura, ki bo v nadaljevanju tudi podrobneje predstavljena.

Pri tej arhitekturi se pojavi dodatni t.i. aplikacijski strežnik, na katerem je nameščena uporabniška aplikacija. To omogoča dodatno razbremenitev odjemalca.

Naloge, ki jih izvaja odjemalec (Predstavitveni nivo):

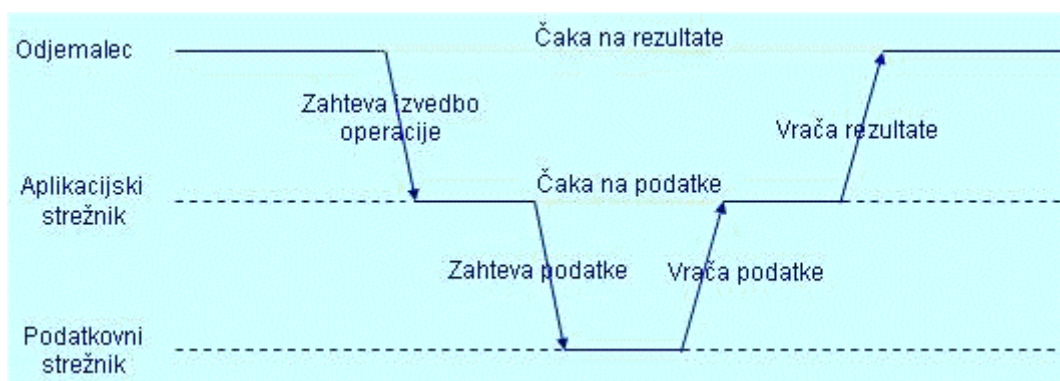
- izvaja akcije na uporabniškem vmesniku, ki pošiljajo zahteve naprej na aplikacijski strežnik

Naloge, ki jih izvaja aplikacijski strežnik (Vmesni nivo):

- preko poslovnih pravil procesira odjemalčeve zahteve

Naloge, ki jih izvaja podatkovni strežnik (Podatkovni nivo):

- shranjuje podatke
- izvaja zahteve, dane na vmesnem nivoju
- zaklepa zapise
- skrbi za integriteto in varnost podatkov



**Slika 6: Interakcija med odjemalcem, aplikacijskim in podatkovnim strežnikom**

Lastnosti trinivojske arhitekture:

- odjemalčeva aplikacija izvaja malo ali nič procesiranja
- aplikacijski strežnik izvaja celotno procesiranje z uporabo poslovne logike
- podatkovni strežnik preverja veljavnost podatkov in nadzoruje dostope do podatkov

Prednosti in slabosti trinivojske arhitekture:

- Prednosti
  - Skalabilnost: lahko imamo več aplikacijskih strežnikov (število aktivnih aplikacijskih strežnikov se lahko dinamično uravnava – Transaction Processing Monitor ali Object Request Broker). Dostop do baze zahteva le povezavo z aplikacijskim strežnikom, zato je lažja zamenjava oziroma nadgradnja posameznih ključnih komponent aplikacije.
  - Večja možnost za ponovno uporabo istih programskih komponent. Če se uporabljajo standardi (COM/COM+ ali CORBA), so značilnosti programskega jezika, kjer je implementiran vmesni nivo, transparentne.
  - Boljša integriteta podatkov. Vse spremembe podatkov potekajo preko vmesnega nivoja, le-ta pa zagotavlja in prepušča le veljavne/dovoljene spremembe (odpravljeno tveganje, da “marsovske” ali “padalske” odjemalčeve aplikacije poškodujejo podatke). V našem primeru prihaja do odstopanja, saj lahko spremembe na bazi izvajajo tudi spletni servisi s pripadajočim izvajajočim modulom, ki niso del vmesnega nivoja in tako ne dostopajo do podatkovne baze preko vmesnega nivoja aplikacije.
  - Boljša varnost podatkov. Število varnostnih nivojev je povečano (+1 na vmesnem nivoju). Zmanjšana potreba po ponovni distribuciji aplikacij – spremembe poslovnih pravil spremenimo le na enem mestu (vmesni nivo), s čimer odpade potreba po ponovni distribuciji aplikacij odjemalcem.
  - Izboljšana razpoložljivost. Lahko imamo podvojene aplikacijske in/ali podatkovne strežnike.
- Slabosti
  - Večja kompleksnost in več napora pri razvoju aplikacij. Opraviti je treba dvojno komunikacijo, z odjemalcem in s podatkovnim strežnikom.
  - Manj programskih razvojnih orodij – na trgu je zaenkrat veliko več razvojnih orodij za razvoj dvonivojskih aplikacij.

### **3.3. Predstavljena funkcionalnost**

#### **3.3.1. Medoperaterski oziroma nabavni ceniki**

Medoperaterski oziroma nabavni ceniki so, iz zornega kota aplikacije, zunanji ceniki. Niso del aplikacije, zato jih je treba v aplikacijo uvoziti. Vsak operater ima lahko svoj format cenika, zato je treba imeti tudi predlogo, preko katere se vhodni cenik preslika v standardni format, ki je aplikaciji znan. Pri postopku uvažanja cenika se v začetku izvede verifikacija pravilnosti cenika. Sledi analiza in poročilo o spremenjenih/novih destinacijah in prefiksih. V primeru, da je cenik akcijski, ga je treba ustrezno obdelati. Podprto je tudi arhiviranje cenikov in zgodovina.

#### **3.3.2. Prodajni ceniki**

Prodajni ceniki so nasprotje nabavnih cenikov in so, iz zornega kota aplikacije, notranji ceniki. Generira jih aplikacija glede na podane parametre, ki so izbrani s strani uporabnika. Prodajna cena je določena glede na usmerjanje, realni promet in dobičkonosnost. Že generiran cenik je možno ročno popravljati in potem shraniti. Tako kot v primeru nabavnega cenika, je tudi v tem primeru možno spremljanje zgodovine.

### **3.4. Predstavitev osnovnih pojmov in funkcij**

#### **3.4.1. Prefiks**

Prefiks je osnovni del informacije, s katerim operira aplikacija. Je del telefonske številke. Začne se z "00" in nadaljuje s kodo države, ki ji pripada. Kodi države sledijo interne kode, ki določajo bolj specifična območja, in številke naročnikov.

Zaradi velikega števila prefiksov, jih je nemogoče upravljati. Zato jih grupiramo v množice, imenovane destinacije. Prefiksom lahko določiš imena in jih organiziraš v direktorije. Aplikacija uporablja sistem, ki je podoben datotečnim sistemom v operacijskih sistemih. Direktoriji so že kreirani v predlogi *master destinations*, lahko pa se dodajajo dodatni prefiksi. Kreiraš lahko celo dodatne predloge in organiziraš prefikse po lastni presoji ali dodatne direktorije za destinacije in podobno.

#### **3.4.2. Cenik**

Vezen je na operaterja in storitev. Namen cenika je definiranje cen za vse stroške, ki se pojavijo med klicem. Tarife stroškov so v večini primerov odvisne od destinacije klica, prav tako pa tudi od drugih faktorjev, kot so kvaliteta storitve in podobno.

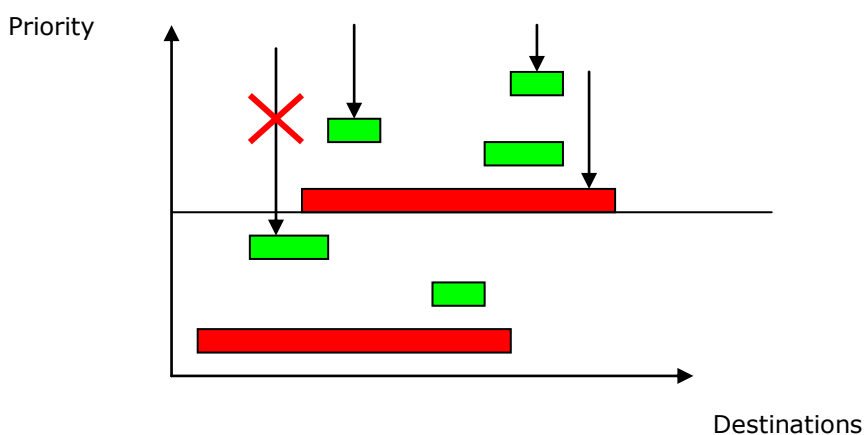
#### **3.4.3. Prioritetni sistem**

Določa, katere cene so veljavne v določenem času. Osnovna logika je sledeča:

- Novejši ceniki imajo višjo prioriteto.

- Novi splošni cenik razveljavi vse starejše cenike.
- Novi akcijski cenik razveljavi cene le za destinacije, definirane v akcijskem ceniku.

Aplikacija najprej prebere destinacije in cene iz cenika z najvišjo prioriteto. To je cenik, ki je najbolj ažuren oziroma smo ga dodali nazadnje. Če je ta cenik splošni, aplikacija preneha z nadaljnim iskanjem in vrne destinacije ter cene cenika. Če je zadnji cenik akcijski, vrača destinacije in cene, glede na prioritete cenikov, vse dokler ne pride do splošnega cenika. Ko pride do splošnega cenika, vrne še vse preostale destinacije in cene, ki niso bile prisotne v prejšnjih akcijskih cenikih. Nadaljnje iskanje se ustavi. Destinacije, ki niso zapisane v nobenem od omenjenih cenikov, ne obstajajo. Operater za te destinacije ne nudi storitev. Grafična predstavitev tega je predstavljena na sliki 7. Rdeči pravokotniki predstavljajo splošne cenike, zeleni pa akcijske. Puščiče kažejo na destinacije, ki predstavljajo veljavne cene.



**Slika 7: Grafični prikaz prioritete določanja trenutnih cen**

#### **3.4.4. Tabela orig destination**

Originalna predloga, kjer so vse destinacije in prefiksi iz uvoženih cenikov.

#### **3.4.5. Tabela master destination**

Predloga, kjer je vsak prefiks originalne predloge razbit na minimalno 10 prefiksov. Število prefiksov se močno poveča.

#### **3.4.6. Tabela template destination**

Ključna težava je, kako učinkovito uporabljati tako veliko število prefiksov in destinacij. To je glavni razlog za implementacijo tabele *template destination* kot del aplikacije. Tabela predstavlja podmnožico destinacij in prefiksov, zato se lahko uporabi kot vhod v ostale postopke aplikacije kot sta routing in priprava prodajnega cenika.

### 3.4.6.1. Karakteristike

- Vsebuje močno omejeno število destinacij in prefiksov (virtualne destinacije)
- Uporabno za routing in pripravo prodajnih cenikov
- Generirana iz master destination tabele s kompresijo prefiksov

### 3.4.7. Uvoz medoperaterskih cenikov

Uvoz zunanjih, Excel ali Word, cenikov v aplikacijo. Postopek uvažanja bom podrobneje predstavil v enem od naslednjih poglavij.

#### 3.4.7.1. Karakteristike

Cenik je Excel oziroma Word datoteka, ki se uvozi v podatkovno bazo aplikacije. Stranski produkt postopka uvoza je ChargeXML datoteka, ki se uporablja za vrednotenje tarif v Traffic aplikaciji. Glavni parametri cenika so operater, veljavnost, tip (splošni, akcijski), smer (originacija, terminacija), vrsta tarifnega intervala (*peak*, *off-peak*, *flat*) ter XML predloga, preko katere se podatki vhodnega cenika preslikajo.

Pomemben del postopka uvoza je verifikacija spremenjenih cen, analiza novih/odstranjenih destinacij in prefiksov. Aplikacija podpira cenike različnih valut. Omogočeno je tudi arhiviranje cenikov, saj so vsi predhodno uvoženi ceniki še vedno dostopni.

### 3.4.8. Kreiranje prodajnega cenika

#### 3.4.8.1. Izračun prodajne cene

Za vsako destinacijo prodajnega cenika se izračuna prodajna cena po eni od poljubnih formul (4), (5) ali (6).

$$\text{Cena} = \text{osnovna cena} + \text{marža [\%]} \quad (4)$$

$$\text{Cena} = \text{osnovna cena} / (1 - \text{marža [\%]}) \quad (5)$$

$$\text{Cena} = \text{osnovna cena} + \text{absolutni znesek} \quad (6)$$

kjer je osnovna cena tista, ki smo jo izračunali po enem izmed algoritmov, ki so predstavljeni v poglavju 3.4.8.2.

Za izračun prodajne cene ni strogo določenega algoritma. V našem primeru smo prodajno ceno izračunali po formuli (5).

Na ceno lahko vpliva tudi kakovost operaterjeve povezave, na kar lahko vplivamo, če izberemo use minimum price oziroma use profit margin. Izbira vpliva na izračun prodajne cene. Le-ta se računa po modificirani formuli (7).

$$\text{Cena} = \text{MAX} (\text{osnovna cena} / (1 - \text{marža}), \text{osnovna cena} + \text{minimalna cena}) \quad (7)$$

kjer je minimalna cena absolutna cena, ki je vezana na operaterja in destinacijo. Minimalna cena se vnaša preko Profit margin dialoga.

### 3.4.8.2. Izračun osnovne cene

Tudi za izračun osnovne cene ni točnega pravila. Za osnovno ceno lahko vzamemo:

- Ceno operaterja na prvi poziciji v routingu<sup>3</sup>
- Maksimalno ceno prvih treh oziroma prvih petih operaterjev v routingu
- Povprečno ceno prvih X operaterjev v routingu
- Minimalno ceno prvih X operaterjev v routingu
- Nabavno ceno na minuto
- Povprečno nabavno ceno na minuto

V našem primeru je osnovna cena enaka povprečni nabavni ceni na minuto, ki se izračuna po vseh pozicijah<sup>4</sup> routinga.

Primer:

1. pozicija → carrier\_1 → volume = 1000 minut, cena = 0,1€
2. pozicija → carrier\_2 → volume = 100 minut, cena = 0,15€

$$\text{Osnovna cena} = (1000 * 0,1 + 100 * 0,15) / 1100$$

### 3.4.8.3. Karakteristike

Prodajni cenik je produkt aplikacije. Parametri cenika so operater, seznam destinacij v tabeli template destination, veljavnost, tip (splošni, akcijski), smer (originacija, terminacija), tarifni interval (peak, off-peak, flat).

Cena za katerikoli tarifni interval se določi z definiranjem naslednjih parametrov: strošek za vzpostavitev klica (setup charge), cena tarifnega intervala (unit charge) in dolžina tarifnega intervala (unit duration).

<sup>3</sup> Routing je proces izbiranja poti v omrežju, po katerem poteka omrežni promet.

<sup>4</sup> Pozicija v usmerjalni tabeli je vrstni red oziroma zaporedna številka operaterja od najboljšega do najslabšega.

Izračun prodajne cene temelji na usmerjanju, realnem prometu in dobičkonosnosti.

Aplikacija omogoča tudi ročno preračunavanje cen, kjer lahko opazujemo spreminjanje marže, dobička in drugih parametrov cenika.

Podprte so različne valute.

### **3.5. Delitev cenika**

Cenike najprej delimo na nabavne in prodajne. Z vidika aplikacije lahko nabavni in prodajni cenik imenujemo tudi notranji in zunanji cenik. Notranji cenik bi bil v tem primeru prodajni, saj je kreiran s strani aplikacije, zunanji cenik pa bi bil nabavni, saj je naši aplikaciji poslan s strani drugih, zunanjih operaterjev. Kot je bilo omenjeno že zgoraj, lahko cenike še dodatno delimo na splošne in akcijske.

#### **3.5.1. Splošni cenik**

Se uporablja za definiranje celotnega cenika za neko določeno kombinacijo operater, storitev. Nov splošni cenik razveljavi vse prejšnje cenike za to kombinacijo. V primeru, da novejši splošni cenik ne definira cene za določeno destinacijo pomeni, da operater za to destinacijo ne zagotavlja več storitev.

#### **3.5.2. Akcijski cenik**

Akcijski cenik je namenjen za popravke in ni nujno, da pokriva celotno območje destinacij, za razliko od splošnega cenika. Ravno tako kot splošni cenik, pa je tudi akcijski definiran za neko določeno kombinacijo operater, storitev. Akcijski cenik, ki je definiran kasneje kot splošni, bo imel višjo prioriteto. Če se za neko destinacijo pojavi več kot ena cena, se upošteva tista iz cenika z višjo prioriteto.

## 4. Načrt (UML diagrami)

UML je standardiziran splošno-namenski načrtovalski jezik za objektno modeliranje. Je rezultat združitve in poenotenja notacij dveh najbolj popularnih objektno usmerjenih metod, Booch-eve in Rumbaugh-jeve, vključuje pa tudi Jacobson-ove primere uporabe (*use cases*). Vključuje množico tehnik grafičnega zapisa za kreiranje abstraktnih modelov določenega sistema. UML ponuja standardni način za opisovanje sistema načrtov, vključno s konceptualnimi stvarmi, kot so poslovni procesi in funkcije sistema, kakor tudi konkretne stvari, kot so izjave programskih jezikov, podatkovne sheme, programska oprema in ponovno uporabljive komponente. Združuje najboljše prakse med koncepti podatkovnega modeliranja, kot so ER diagrami, poslovno modeliranje (delovni tok), objektno modeliranje in komponentno modeliranje. Lahko se uporablja pri vseh procesih, skozi življenjski cikel razvoja programske opreme in pri različnih implementacijskih tehnologijah [9].

Pomembna vloga pri napredku objektne tehnologije je odigrala organizacija OMG (*Object Management Group*), ki skrbi za standardizacijo na tem tehnološkem področju.

UML diagrami so razdeljeni v tri kategorije, ki se dodatno razdelijo v podkategorije:

- Strukturni
  - Razredni diagram
  - Komponentni diagram
  - Arhitekturni diagram
  - Objektni diagram
  - Paketni diagram
- Vedenjski
  - Diagram aktivnosti
  - Diagram stanja
  - Diagram primerov uporabe
- Interakcijski
  - Diagram sodelovanja
  - Diagram zaporedja
  - Časovni diagram

### 4.1. Diagram primerov uporabe

#### 4.1.1. Definicija

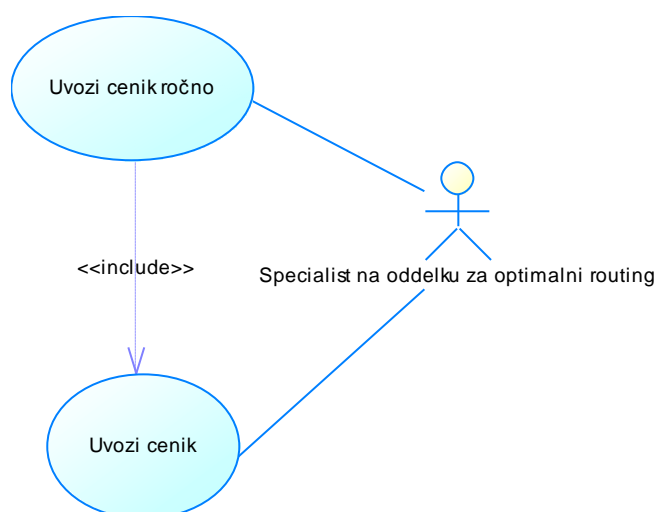
Diagram primerov uporabe predstavlja povezavo med akterji – uporabniki sistema in procesi, ki tečejo v sistemu. Primeri uporabe so običajno namenjeni zajemanju in predstavitvi uporabniških zahtev. Predstavljajo osnovo za objektno modeliranje z uporabnikovim sodelovanjem skozi celoten razvojni cikel. Služijo kot osnova pri definiranju funkcionalnih zahtev, identificiranju in določanju funkcionalnosti poslovnih objektov, definiranju interakcije med poslovnimi objekti, načrtovanju uporabniških vmesnikov, izvajanju integracijskega testiranja in definiranju testnih primerov. Pomagajo pri definiranju systemskega in kontrolnega

načrtovanja, kot sredstvo za zajemanje in sledenje strankam, razgradnji sistema in ocenjevanju velikosti projekta ter virov, ki jih potrebujemo.

S pomočjo diagramov primerov uporabe modeliramo vplive okolja na sistem. Sistem modeliramo kot množico procesov, ki uporabniku ponujajo usluge (storitve) sistema.

Hkrati okolje sistema modeliramo kot množico akterjev, ki vplivajo na sistem oziroma od sistema zahtevajo merljiv rezultat [6].

#### 4.1.2. Diagram primerov uporabe za uvoz nabavnega cenika



**Slika 8: Diagram primera uporabe za uvoz nabavnega cenika**

Diagram primerov uporabe na sliki 8 prikazuje interakcijo med akterjem in dvema primeroma uporabe izmed katerih je primer uporabe “Uvozi cenik” del primera uporabe “Uvozi cenik ročno”.

##### 4.1.2.1. Opis toka dogodkov za primer uporabe Uvozi cenik

Primer uporabe omogoča uporabniku uvoz cenika iz Excel oziroma Word datoteke v aplikacijo. Vhodni cenik, se preko predloge, ki jo uporabnik izbere še pred uvozom, pretvori v interno standardno obliko. Predloge so med seboj različne, odvisno od operaterja in tipa cenika. Ob uspešni pretvorbi cenika v interni standardni format, ga lahko shranimo v podatkovno bazo.

Osnovni tok:

- A. Ob kliku na zavihek `Import` se prikažejo vnosna in izbirna polja.
- B. Uporabnik izbere operaterja, tip cenika, predlogo, valuto, cenik.
- C. Aplikacija prebere vhodni cenik, predlogo ter ga pretvori v standardni format, katerega podatki se shranijo v podatkovno bazo.

Alternative:

Alt. 1: Pretvorba cenika v standardni format. Napaka pri pretvorbi. Popravi vhodni cenik ročno glede na tip napake (podvojene destinacije, različne cene za isto destinacijo,...).

Alt. 2: V vhodnem ceniku se pojavi nov prefiks. Postopek uvoza je prekinjen. Izbere se ikona `Manage master destinations` v zavihku `Prefiks`. Kreira se nova destinacija. Postopek uvoza se ponovi.

Alt. 3: V vhodnem ceniku se pojavi nova destinacija. Postopek uvoza je prekinjen. Izbere se ikona `Manage master destinations` v zavihku `Prefiks`. Kreira se nova destinacija. Postopek uvoza se ponovi.

#### 4.1.2.2. Opis toka dogodkov za primer uporabe Uvozi cenik ročno

Primer uporabe omogoča uporabniku ročni uvoz cenika z dodajanjem posameznih destinacij in prefiksov ter pripadajočih cen. Razlika je samo v tem, da tokrat nimamo danega vhodnega cenika v Excel oziroma Word format, ampak dodajamo le posamezne postavke. Pri tem postopku ne potrebujemo predloge, saj se vhodni podatki ne pretvarjajo, ampak se po uspešno izvedenem primerjanju dodajo v podatkovno bazo. To pomeni, da je preostali del postopka isti kot pri avtomatskem uvozu vhodnega cenika.

Osnovni tok:

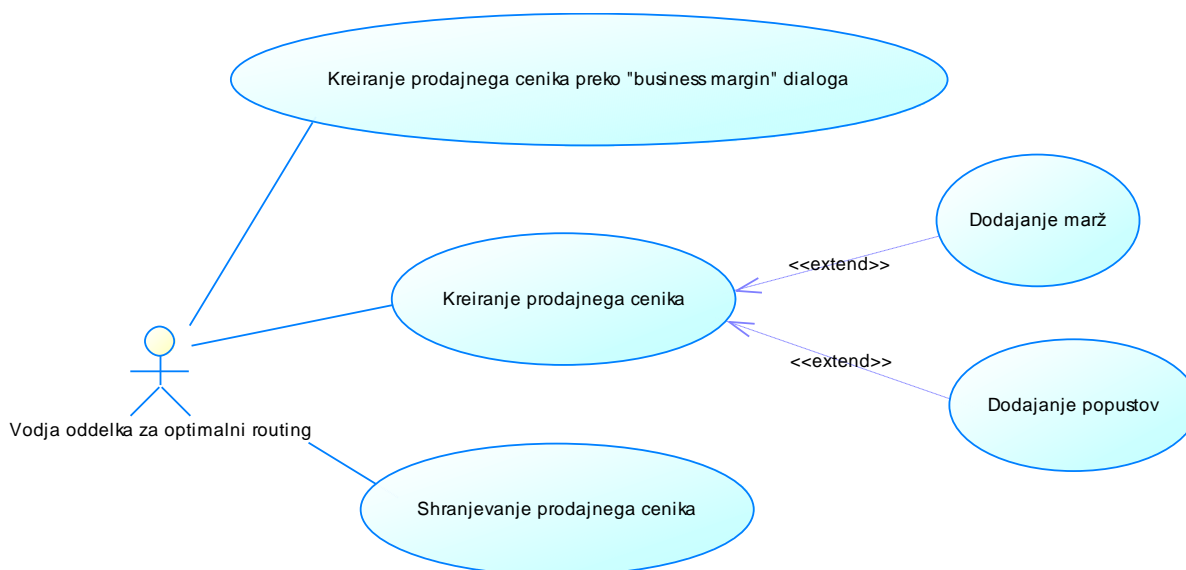
- A. Ob kliku na zavihek `Import` se prikažejo vnosna in izbirna polja. V tem primeru jih ni treba upoštevati. Izberemo ikono `Manual pricelist`.
- B. Uporabnik izbere tip operaterja, operaterja, veljavnost cenika, ime cenika, tip cenika in valuto.
- C. Doda posamezne postavke cenika: destinacija, prefiksi, cena, veljavnost cene. Možnost izbire večih tarif.

Alternative:

Alt. 1: V vhodnem ceniku se pojavi nov prefiks. Postopek uvoza je prekinjen. Izberemo ikono `Manage master destinations` v zavihku `Prefiks`. Kreira se nova destinacija. Postopek uvoza se ponovi.

Alt. 2: V vhodnem ceniku se pojavi nova destinacija. Postopek uvoza je prekinjen. Izberemo ikono `Manage master destinations` v zavihku `Prefiks`. Kreira se nova destinacija. Postopek uvoza se ponovi.

### 4.1.3. Diagram primerov uporabe za kreiranje prodajnega cenika



**Slika 9: Diagram primera uporabe za kreiranje prodajnega cenika**

Diagram primera uporabe na sliki 9 prikazuje interakcijo med uporabnikom in primeri uporabe, pri postopku kreiranja novega prodajnega cenika.

#### 4.1.3.1. Opis toka dogodkov za primer uporabe Kreiranje prodajnega cenika

Primer uporabe kreira novi prodajni cenik z možnostjo upoštevanja marž in popustov.

Osnovni tok:

- A. Ob kliku na zavihek New, izberi ikono Auto-generate selling pricelist.
- B. Odpre se dialog, kjer izbereš parametre algoritem, operater, minimalna cena oziroma marža v procentih.
- C. Izračunajo se cene za izbranega operaterja in vse destinacije.
- D. Kreiran cenik lahko še ročno spreminjamo ter izvajamo razne izračune.

#### 4.1.3.2. Opis toka dogodkov za primer uporabe Kreiranje prodajnega cenika preko "business margin" dialoga

Primer uporabe kreira novi prodajni cenik iz izbranega cenika z upoštevanjem upravljalških marž.

Osnovni tok:

- A. Ob kliku na zavihek `New` izberi ikono `Manager margins`.
- B. Odpre se dialog, kjer se izbereš cenik in upravljalno maržo.
- C. Izračunajo se nove cene.
- D. Kreiran cenik lahko še ročno spreminjamo ter izvajamo razne izračune.

#### **4.1.3.3. Opis toka dogodkov za primer uporabe Shranjevanje prodajnega cenika**

Dialog za shranjevanje prodajnega cenika oziroma ročnih popravkov generiranega prodajnega cenika.

Osnovni tok:

- A. Po uspešno generiranem prodajnem ceniku oziroma po uspešno izvedenih ročnih spremembah, kliknemo ikono `Save pricelist`.
- B. Odpre se dialog, kjer izberemo veljavnost cenika in njegovo ime.
- C. Ob kliku na ikono `Save pricelist` cenik shranimo v bazo. Dialog se zapre.

Alternative:

Alt. 1: Če sprememb v ceniku ne želimo shraniti, kliknemo ikono `Cancel`. Dialog se zapre, spremembe se ne shranijo v podatkovno bazo.

#### **4.1.3.4. Opis toka dogodkov za primer uporabe Dodajanje marž glede na operaterja**

Še preden se generira novi prodajni cenik, lahko posameznemu operaterju in njegovim destinacijam določimo maržo.

Osnovni tok:

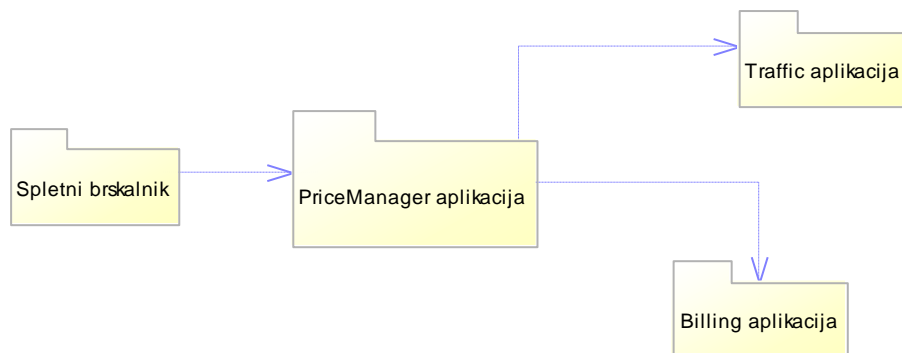
- A. Ob kliku na zavihek `New`, izberi ikono `Profit margins`.
- B. Odpre se dialog, kjer za izbranega operaterja in destinacijo vpišemo minimalno ceno in/ali maržo v %.
- C. Vnos se shrani avtomatsko v podatkovno bazo, ko se premaknemo v novo vrstico.
- D. Za izhod iz dialoga klikni gumb `Close`.

## **4.2. Paketni diagram**

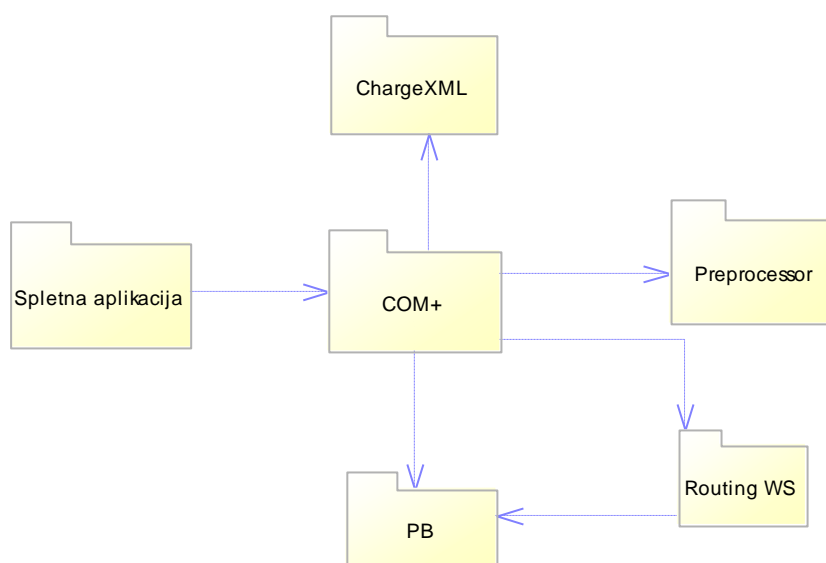
Naloga paketnega diagrama je razčlenitev velikega sistema na manjše dele. Paket predstavlja skupino modelirnih elementov, grupiranih po pomenu, z namenom zmanjšanja kompleksnosti. Idejo grupiranja lahko uporabimo pri vsakem elementu UML modela, ne samo pri razredih.

Paketi nam ne dajo odgovora na vprašanje, kako zmanjšati odvisnosti v sistemu, ampak nam pomagajo ugotoviti, katere odvisnosti v sistemu obstajajo.

Paketni diagrami so pomembno orodje pri vzdrževanju in upravljanju celotne strukture sistema. Uporaba je primerna oziroma nujna, predvsem kadar razrednega diagrama ali diagrama primerov uporabe, ki zajema obsežnejši del sistema, ni mogoče predstaviti v primernem formatu [6].



**Slika 10: Paketni diagram**



**Slika 11: Paketni diagram PriceManager aplikacije**

Spletna aplikacija je množica spletnih strani, ki izvajajo uporabnikove zahteve in vračajo ustrezne rezultate.

COM+ skrbi za izvajanje poslovne logike in za interakcijo s podatkovno bazo (paket PB). Zahteve prihajajo iz paketa Spletna aplikacija. Ta paket vrši tudi zahteve do vseh ostalih vpletenih modulov, kot so Preprocessor, spletne storitve Routing, itd.

Preprocessor skrbi za pretvorbo Excel in Word vhodnih cenikov v standardni format, ki ga uporablja aplikacija.

PB je podatkovna baza naše aplikacije.

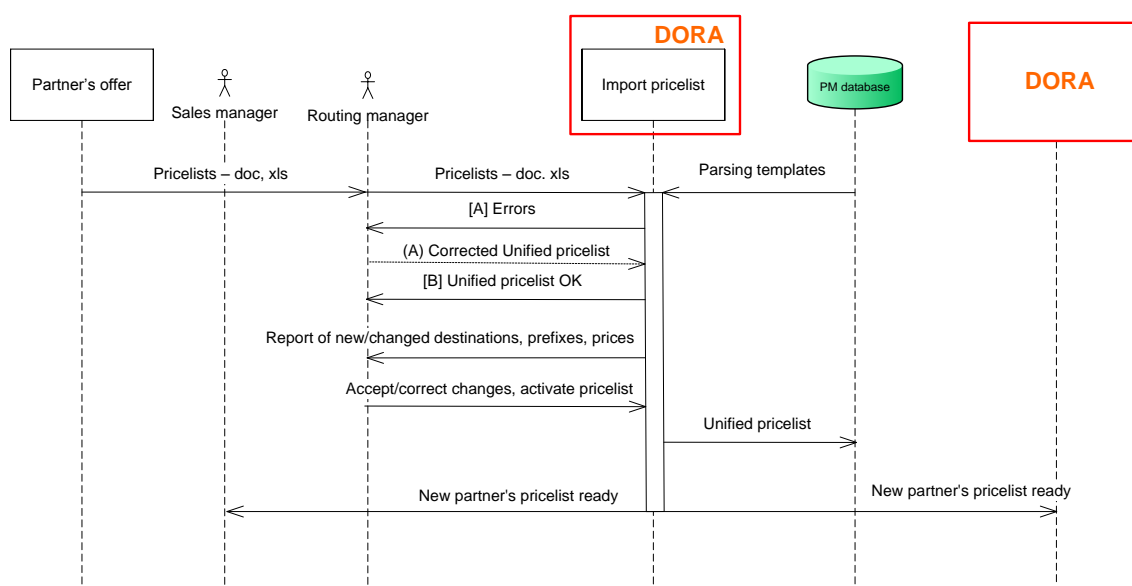
Routing WS so spletni servisi, ki izvajajo routing module kot so moduli za optimizacijo, simulacijo in podobno.

### 4.3. Diagram zaporedja

Diagram zaporedja prikazuje interakcije na osnovi časovnega zaporedja. Prikazuje, kako objekti sodelujejo in izmenjujejo sporočila v interakcijah na osnovi časovnih linij. Diagram ne prikazuje povezav med objekti. Obstaja v splošni obliki, kjer opisuje vsa možna zaporedja, in v obliki primerkov, kjer opisuje konkretno zaporedje, skladno s splošno obliko. V primerih brez zank in vejitev sta obe obliki izomorfni.

Diagram zaporedja ima dve dimenziji: vertikalno, ki prikazuje čas, in horizontalno, ki prikazuje različne objekte. Praviloma čas teče navzdol, vendar to ni obvezujoče. Po navadi so pomembne le časovne sekvence, za aplikacije v realnem času pa je lahko časovna os dejanska metrika. Osi lahko zamenjamo, tako da čas teče horizontalno in so objekti predstavljeni na vertikalni osi. Ob strani ali pri prehodih med objekti lahko dodamo besedilo ali tekstovno oznako. Horizontalno razvrščanje objektov ni pomembno, lahko pa jih združujemo [6].

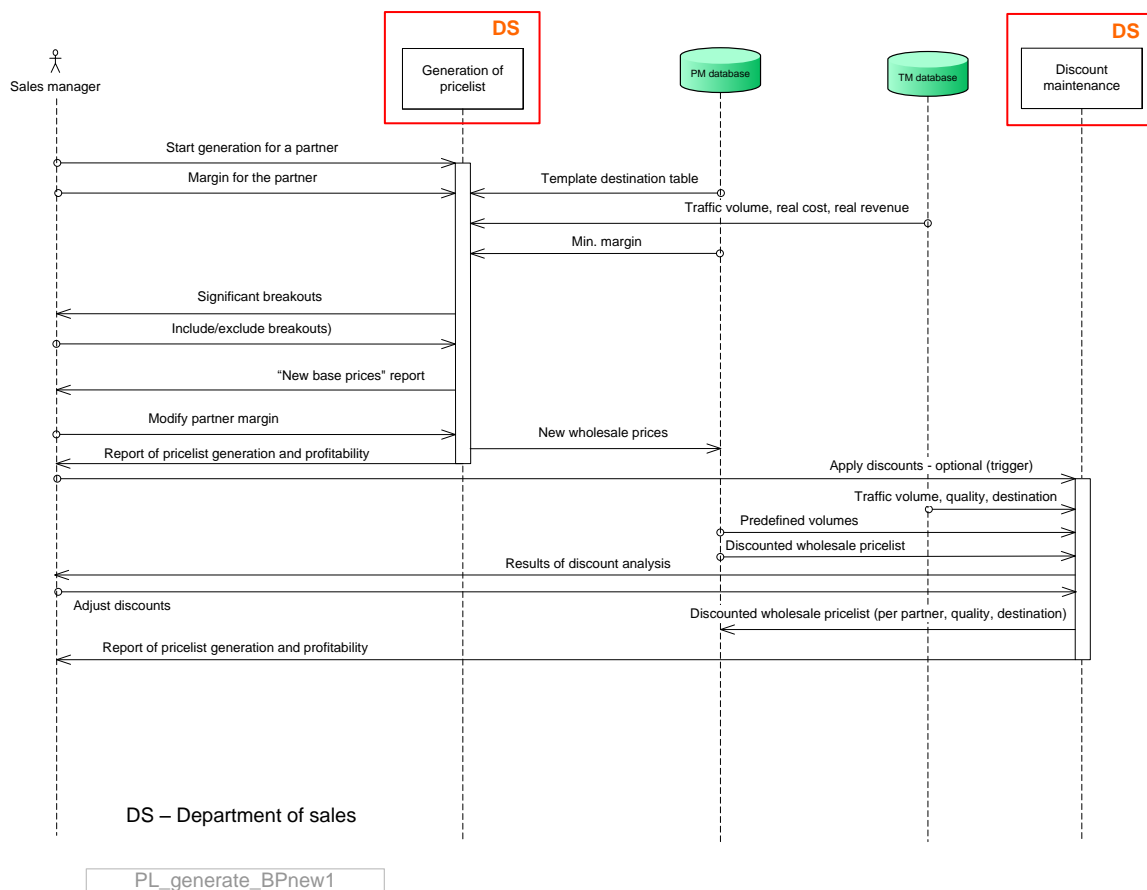
New business process: *Pricelist import*



DORA – Department of optimal routing

PL\_import\_BPnew1

**Slika 12: Diagram zaporedja za uvoz nabavnih cenikov**

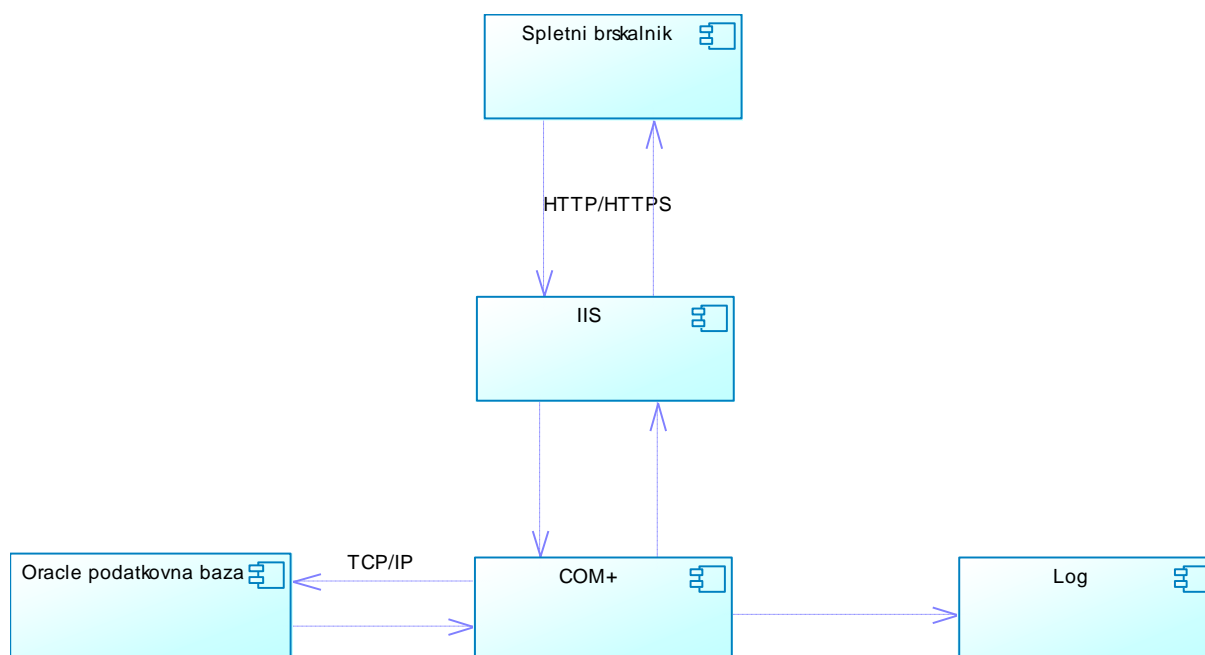


**Slika 13: Diagram zaporedja za kreiranje prodajnih cenikov**

#### 4.4. Komponentni diagram

Komponentni diagram prikazuje odvisnosti med programskimi komponentami. Vključuje izvedljive komponente ter izvorno in binarno kodo komponent. Programski modul predstavimo kot tip komponente, ki vsebuje programsko kodo elementov v sistemu. Nekatere komponente lahko obstajajo v času prevajanja, druge v času povezovanja in tretje v času izvajanja.

Je graf, kjer so vozlišča komponente, povezave pa relacije med njimi. Vozlišča lahko predstavimo z napravami, ki predstavljajo fizične vire. Naprave so lahko na primer: tiskalnik, računalnik, osrednji računalnik, telefon, strežnik, itd. S komponentnim diagramom modeliramo statični pogled na implementacijo sistema. V osnovi so komponentni diagrami razredni diagrami, ki sistem obravnavajo z vidika komponent.



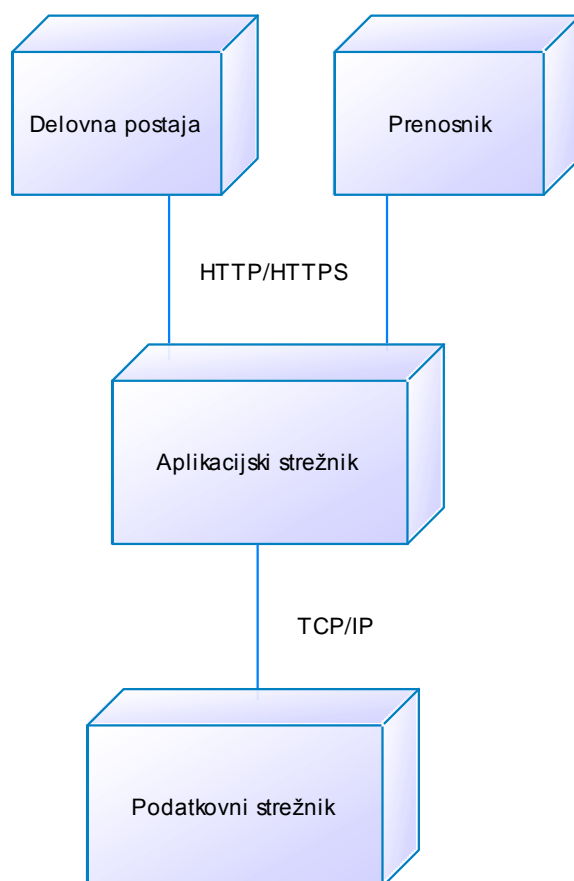
**Slika 14: Komponentni diagram**

#### 4.5. Arhitekturni diagram

Arhitekturni diagram prikazuje konfiguracijo izvajalnega elementa in programskih komponent, procesov in objektov, ki se nahajajo v njem. Primerki programskih komponent predstavljajo izvedljivo programsko kodo enot. Komponente, ki ne obstajajo kot izvedljive entitete, prevajale so se drugje, ne nastopajo v tem diagramu, ampak jih prikažemo na diagramu komponent. S tem diagramom predstavimo fizično arhitekturo sistema.

Vsako vozlišče v diagramu prikazuje enoto neke vrste. V večini primerov je to del strojne opreme. Strojna oprema je lahko preprosta naprava, kot na primer senzor, osrednji računalnik, tiskalnik in telefon.

Je graf vozlišč, ki so povezana s komunikacijsko asociacijo. Vozlišča morajo vsebovati primerke komponent, kar pomeni, da komponente živijo oziroma se izvajajo v vozliščih. Komponente vsebujejo objekte, zato so objekti del komponent. Povezave med komponentami predstavimo s črtno puščico, ki predstavlja odvisnost, po možnosti preko vmesnikov. To pomeni, da ena komponenta uporablja storitve drugih komponent. Premik komponent, od vozlišča do vozlišča, ali objektov, od komponent do komponent, prikažemo z usmerjeno povezavo.



**Slika 15: Arhitekturni diagram**

## 5. Izvedba

### 5.1. Opis nabavnega cenika

Nabavni cenik je cenik, ki ga pošljejo operaterji, preko katerih bomo terminirali klic. Cenik je lahko v Excel ali Word formatu (glej prilogo 1 in 2). Standardni format ni definiran, ampak ga pošilja vsak operater v svojem formatu. To seveda povzroča večje težave pri uvozu cenika, saj se lahko pojavijo nepredvideni problemi. Če so problemi manjše narave, je v večini primerov dovolj, da se izvorni cenik popravi oziroma se predloga temu primerno priredi. Vsak operater ima lahko več svojih predlog za različne tipe formatov cenikov. Predloge so opisane v naslednjem poglavju.

V skrajnem primeru lahko cenik pretvorimo v standardno obliko z Excel makrojem in šele nato uvozimo.

Vsak cenik vsebuje naslednje elemente:

- Partner, ime operaterja
- Destinacije, seznam destinacij, regij
- Seznam prefiksov za vsako destinacijo
- Valuta cenika
- Datum začetka veljavnosti cenika
- Datum zaključka veljavnosti cenika
- Cene, seznam cen za vsako destinacijo
- Smer spremembe cene, dvig ali spust (opcijsko)

Podatki cenika so lahko v večih zavihkih, odvisno od operaterja. Temu primerno je napisana tudi predloga.

### 5.2. Opis predloge za uvoz nabavnega cenika

Predloga je XML konfiguracijska datoteka, namenjena za branje različnih cenikov in pripravo prebranih podatkov za uvoz v aplikacijo. Dovoljuje uvoz različnih formatov cenikov (Excel tabel in Word dokumentov) v njegov interni format. Konfiguracijska datoteka določa, kako bodo ceniki uvoženi. Je preslikava med cenikom v izvornem formatu in cenikom v internem formatu.

Vsak od elementov cenika je lahko tipa:

- Scalar
- List, seznam vrednosti
- Dictionary, seznam parov ključ/vrednost

Elementi tipa Scalar so:

- Partner
- Valuta cenika
- Datum veljavnosti

Element tipa List je seznam destinacij.

Elementa tipa Dictionary sta:

- Seznam prefiksov
- Seznam cen

XML struktura konfiguracijske datoteke:

```
<PriceListPreprocessor>
  <PriceListConfiguration>
    ... Element tags ...
  </PriceListPreprocessor>
</PriceListConfiguration>
```

## 5.2.1. Pravila

Vsak element mora imeti vire, element SOURCES. Ta element opisuje kateri del izvirnega dokumenta, Excel tabela ali Word tabela, je vir podatkov. Elementi imajo lahko element RULES, ki opisuje, kako vhodne podatke pretvoriti v obliko, primerno za vpis v podatkovno bazo. Element RULES se lahko uporablja tudi za filtriranje podatkov.

Prebrani podatki se shranijo v interno strukturo. S temi podatki lahko upravljamo s pomočjo pravil, ki se izvajajo v istem vrstnem redu, kot so zapisana v XMLu. Če predloga ne vsebuje elementa RULES, potem se nad temi podatki ne izvede nobene akcija.

### 5.2.1.1. Tipi pravil

#### 5.2.1.1.1. Distinct

Odstrani podvojene vrstice.

Ime	Opis
applyTo (required)	Definira, na kaj se pravilo asociira. Veljavne vrednosti so Key, Data in All.

### 5.2.1.1.2. Split

Eno samo vrednost razdeli na eno ali več vrednosti glede na izbran separator. Pravilo lahko razdeli celoten niz znakov ali le del.

Ime	Opis
applyTo (required)	Definira, na kaj se pravilo nanaša. Veljavna vrednost je Data, v nasprotnem primeru se pravilo ignorira.
separator	Definira tekstovni separator. Rezultat je seznam razdeljenih tekstovnih znakov.
Action	Definira, kako bo tekstovni niz razdeljen. Veljavne akcije so: <ul style="list-style-type: none"> <li>• SplitAfterText – razdeljen je le tisti del niza, ki stoji za separatorjem, definiranim v atributu action. Primer: če je vrednost atributa actionValue   in je niz 3;5 4;6, bo imel rezultat dve vrednosti 3;54 in 3;56.</li> <li>• SplitBeforeText – razdeljen je le tisti del niza, ki stoji pred separatorjem, definiranim v atributu action. Primer: če je vrednost atributa actionValue   in je niz 3;5 4;6, bo rezultat 34;6 in 54;6.</li> <li>• SplitMultiPartText – razdeljen je le tisti del niza, ki stoji za separatorjem, definiranim v atributu actionValue. Primer: če je actionValue   in je niz 3;5 4;6, bo rezultat 34, 36, 54 in 56.</li> <li>• “” - razdeljen je celotni niz. Primer: če je action ; in je niz 3;5 4;6, bo rezultat 3, 5 4 and 6.</li> <li>•</li> </ul>
rangeSeparator	Namenjen je definiranju območij števil za potrebe definiranja prefiksov. Primer: če je rangeSeparator '-' in so podatki 3-5 potem bo rezultat 3,4 in 5.
actionValue	Separator za atribut action.

### 5.2.1.1.3. RemoveEmpty

Odstrani prazne vrednosti iz seznama oziroma slovarja.

Ime	Opis
applyTo (required)	Definira, na kaj pravilo nanaša. Če je vrednost Data, odstrani vrstico, če je vrednost empty, ne glede na to kaj je v ključu. Če je vrednost Key, pravilo odstrani vrstico, če je ključ empty. Če je vrednost All, se zapis odstrani le če sta ključ in podatek empty.

#### 5.2.1.1.4. Substring

Vrne del tekstovnega niza, ki je definiran med `startText` in `endText`. V rezultatu so odstranjeni tudi odstranjene vsi presledki.

Ime	Opis
<code>applyTo</code> (required)	Definira, na kaj se pravilo nanaša. Če je vrednost <code>Data</code> , pravilo spreminja podatek. Če je vrednost <code>Key</code> , se pravilo aplicira na ključ. In če je vrednost <code>Both</code> , se pravilo aplicira tako na ključ kot na podatek.
<code>startText</code> (required)	Tekst, ki definira začetek izvajanja pravila. Tekst ni vključen v rezultat. Če je niz prazen, se začne na začetku tekstovnega niza.
<code>endText</code> (required)	Tekst, ki definira konec izvajanja pravila. Tekst ni vključen v rezultat. Če je niz prazen, se vzame konec niza.
<code>startIndex</code> (optional)	Pozicija prvega znaka. Njegova vrednost je celo število ali prazen niz (privzeto).
<code>endIndex</code> (optional)	Pozicija zadnjega znaka. Njegova vrednost je celo število ali prazen niz (privzeto).

#### 5.2.1.1.5. Contain

Glede na vrednost atributa `action`, odstrani tekst, ki vsebuje oziroma ne vsebuje danega teksta v ključu ali/in podatku.

Ime	Opis
<code>applyTo</code> (required)	Definira, na kaj se pravilo nanaša. Če je vrednost <code>Data</code> , bo pravilo preverjalo samo podatek. Če je vrednost <code>Key</code> , bo preverjalo vrednost v ključu. Če pa je vrednost <code>All</code> , bo pa preverjalo oboje.
<code>action</code>	Če je vrednost atributa <code>keep</code> , potem bodo vrstice, ki vsebujejo vrednost atributa <code>Text</code> , ohranjene, v nasprotnem primeru bodo odstranjene. Če je vrednost <code>remove</code> , bodo odstranjene in ostale bodo ohranjene za nadaljne procesiranje.
<code>Text</code>	Tekstovni niz za preverjanje.

#### 5.2.1.1.6. Replace

Išče in zamenja tekstovni niz.

Ime	Opis
<code>applyTo</code> (required)	Definira, na kaj se pravilo nanaša. Če je vrednost <code>Data</code> , potem se pravilo nanaša na podatkovni del. Če je vrednost <code>Key</code> , potem se pravilo nanaša na ključ. Če je vrednost <code>All</code> , se pravilo nanaša na obe vrednosti.

oldText	Iskan tekstovni niz.
newText	Tekstovni niz, ki bo zamenjal iskan tekstovni niz.

### 5.2.1.1.7. Trim

Odstrani presledke in ostale znake v začetku in koncu tekstovnega niza.

Ime	Obvezno	Opis
applyTo	Da	Definira, na kaj se pravilo nanaša. Če je vrednost <code>Data</code> , potem se pravilo nanaša na podatkovni del. Če je vrednost <code>Key</code> , se nanaša na ključ. Če pa je vrednost <code>All</code> , se nanaša na oboje.
from	Da	Definira, nad katerim delom tekstovnega niza se izvede pravilo. Veljavne vrednosti so: <ul style="list-style-type: none"> <li>• <code>Start</code>: niz definiran v atributu <code>text</code> bo odstranjen iz začetka danega niza. Ostali deli danega niza ostanejo nedotaknjeni.</li> <li>• <code>End</code>: niz definiran v atributu <code>text</code> bo odstranjen s konca danega niza. Ostali deli danega niza ostanejo nedotaknjeni.</li> <li>• <code>Empty ("")</code>: niz je odstranjen na obeh koncih danega niza.</li> </ul>
text	Da	Seznam odstranjenih znakov.

### 5.2.1.1.8. Format

Spremeni obliko vrednosti, predstavljene v tekstovnem, numeričnem ali datumskem formatu. Primer, lahko spremeni obliko datum iz oblike `dd.mm.yyyy` v obliko `mm/dd/yy` in obratno.

Ime	Obvezno	Opis
applyTo	Da	Definira, na kaj se pravilo nanaša. Če je vrednost <code>Data</code> , se pravilo izvede nad podatkovnim delom. Če je vrednost <code>Key</code> , se pravilo izvede nad ključem. In če je vrednost <code>All</code> , se izvede nad obema.
valueType	Da	Tip vrednosti. Veljavne vrednosti so: <ul style="list-style-type: none"> <li>• <code>String</code>: uporabljen za spreminjanje tekstovnih nizov. Primer: lahko oviješ tekstovni niz okrog vrednosti. Prvotna vrednost je dosegljiva z <code>{0}</code>. Primer: <code>valueType = "***{0}**"</code> bi pretvoril vrednost <code>ICE</code> v <code>**ICE**</code>.</li> <li>• <code>DateTime</code>: uporabljen za pretvarjanje datuma</li> </ul>

		<p>oziroma dela datuma v <code>date/time</code> nizu. Preprocessor javi napako, če vrednost ne ustreza obliki, definirani v <code>oldFormat</code> atributu. Uvažajo se lahko ceniki z datumom v obliki <code>dd.MM.yyyy</code>.</p> <ul style="list-style-type: none"> <li>• <code>Int</code>: vrednost je pretvorjena v celo število in potem nazaj v tekstovni niz. <code>oldFormat</code> parameter se ignorira. <code>newFormat</code> parameter se uporabi za formatiranje izhodnega rezultata. Metoda lahko pretvarja tudi šestnajstiške vrednosti. Preprocessor javi napako, če vrednost ni celoštevilskega tipa.</li> <li>• <code>Decimal</code>: vrednost se pretvori v decimalno število in potem nazaj v tekstovni format upoštevajoč parameter <code>newFormat</code>. Preprocessor javi napako, če vrednost ni ustreznega tipa.</li> </ul>
<code>oldFormat</code>	Da	Vhodni format. Za več informacij glej opis <code>valueType</code> .
<code>oldCulture</code>	Ne	Definira jezikovne nastavitve vhodnega toka podatkov.
<code>newFormat</code>	Da	Izhodni format. Za več informacij glej opis <code>valueType</code> .
<code>newCulture</code>	Ne	Definira jezikovne nastavitve vhodnega toka podatkov.

### 5.2.1.1.9. FormatDate

Formatira datum glede na eno/več mask. Če je polje že v datumskem formatu, se uporabi vrednost v polju brez nadaljnega formatiranja.

#### 5.2.1.1.9.1. Atributi

Ime	Opis
<code>applyTo</code>	Definira, na kaj se pravilo nanaša. Veljavne vrednosti so <code>Data</code> (vsi podatkovni elementi) ali <code>Key</code> (podatkovni elementi tipa <code>Dictionary</code> ).
<code>outputFormat</code> (optional)	Izhodna format maska. Če maska ni določena, se uporabi <code>dd.MM.yyyy</code> maska.
<code>outputCulture</code> (optional)	Privzeta vrednost je prazni niz.
<code>allowEmpty</code> (optional)	Veljavni vrednosti sta <code>Yes</code> in <code>No</code> . Če je vrednost <code>Yes</code> , potem je prazen niz dovoljen (preprocessor ne javi napake). Privzeta vrednost je <code>No</code> .
<code>clearIfNoMatch</code> (optional)	Veljavni vrednosti sta <code>Yes</code> in <code>No</code> . Če <code>Yes</code> , preprocessor zamenja trenutno vrednost s praznim nizom, če se maska ne ujema. V nasprotnem primeru preprocessor javi napako.

	Privzeta vrednost je No.
--	--------------------------

### 5.2.1.1.9.2. Podelamenti

#### **masks**

Vsebuje enega ali več mask elementov. Predstavljajo veljavne format maske. Element mask vsebuje naslednje attribute:

- Format
- Culture – (neobvezno). Koda jezika. Privzeta vrednost je prazni niz.

#### **dateValidations**

Vsebuje enega ali več elementov za preverjanje datuma. Ti elementi predstavljajo posebna pravila (dan je lahko 1. in 15. in podobno). Pravila so opisana z dateValidation elementom.

Ime	Opis
what	Določa predmet validacije. Veljavne vrednosti so day, month ali year.
validValues (optional*)	Določa seznam veljavnih vrednosti, ločenih z vejico ali '-'.
invalidValues (optional*)	Določa seznam neveljavnih vrednosti, ločenih z vejico ali '-'.
strong (optional)	

\*Vsaj eden od obeh atributov mora biti prisoten.

## 5.2.2. Elementi

### 5.2.2.1. SOURCES

Kot je že bilo omenjeno zgoraj, element opisuje del datoteke, ki vsebuje vhodne podatke. Ker so podatki lahko razpršeni po datoteki, ima določen element SOURCES več SOURCE elementov, ki so med seboj povezani.

Atribut	Obvezen	Opis
Type	Da	Tip vira.
dataLinkingFormat	Da	Opisuje, kako so različni viri med seboj povezani. Številke v zavutih oklepajih so zamenjane s prebranimi vrednostmi iz vhodne datoteke. Ostali znaki se

		kopirajo. Deluje na enak način kot .NET funkcija <code>String.Format(linkingFormat, source1, source2,...)</code> .
<code>keyLinkingFormat</code>	Da	Uporabljen le v tipu <code>Dictionary</code> in opisuje relacijo ključa. Deluje na enak način kot <code>dataLinkingFormat</code> .

### 5.2.2.2. SOURCE

Atribut	Obvezen	Opis
<code>id</code>	Da	ID vira. Mora se začeti z 0 in nadaljevati z 1, 2, 3, ...
<code>resource</code>	Da	Določa ali je vir ime datoteke ali vsebina datoteke. Veljavni vrednosti sta <code>FileContent</code> in <code>FileName</code> .
<code>table</code>	Ko ima <code>resource</code> atribut vrednost <code>FileContent</code>	Ime zavijka v Excel datoteki, številka tabele v MS Word datoteki.
<code>range</code>	Ko ima <code>resource</code> atribut vrednost <code>FileContent</code>	Definira območje, del tabele. <i>Glej definicijo range atributa.</i>

#### 5.2.2.2.1. Atributi

##### Range

Območja so definirana z atributom `range`, ki ima naslednji format:

```
startRow: startColumn: endRow: endColumn
```

`startRow` lahko zavzame naslednje vrednosti:

- Numerično vrednost
- Vrednost `FromText`
  - Preprocesor išče prvo najdeno besedo, ki je navedena v 1. stolpcu
    - `FromText:1:1000:1::Text to search`
- Vrednost `FromTextEx`
  - Podobno kot `FromText`, vendar preprocesor išče n-to besedo in začne iskati na x-ti poziciji
    - `FromTextEx:1:1000:1::Labrador:x:n`

startColumn, endRow in endColumn lahko zavzamejo naslednje vrednosti:

- Numerično vrednost za fiksno stolpec/vrstica pozicijo
- Last
  - Išče zadnjo celico, ki vsebuje podatek
- FirstEmpty
  - Išče prvo prazno celico
- UntilText
  - Išče toliko časa, dokler ne najde celice s tekstom definiranim v 1. stolpcu
  - Primer:
    - *2: 1: UntilText: 1: Total*, začne v drugi vrstici in nadaljuje do vrstice v kateri najde Total.

Iskanje je omejeno, in sicer podatki ne smejo biti v obliki matrice, ampak morajo biti v enodimenzionalni obliki.

#### 5.2.2.2. Pravilo

##### Zamenjava vrednosti (podprto samo v Excelu)

Vsak element SOURCE lahko zamenja vrednosti, prebrane iz polj z drugimi vrednostmi. To se izvede z elementom REPLACE, ki je podelement elementa SOURCE.

Atribut	Obvezen	Opis
What	Ne	Zamenjani tekst. Če je atribut izpuščen, se uporabijo vse vrednosti.
TableName	Ne	Ime tabele, kjer se vrednosti polj zamenjajo. Če je atribut izpuščen, se uporabi ista tabela kot v elementu SOURCE.
Cell	Da	Naslov polja, iz katerega se vzame vrednost za zamenjavo. Naslov je lahko fiksni v obliki RowNo: ColNo ali v obliki SearchText: ColNo: Text to Find: StartRow: NumberOfOccurrences, kjer se išče Text to Find.

### 5.2.3. Primer strukture XML konfiguracijske datoteke

XML konfiguracijska datoteka oziroma predloga se uporablja pri postopku uvoza cenika v aplikacijo. Namenjena je pretvorbi Excel, Word cenika v standardno obliko, ki omogoča aplikaciji hitrejše in učinkovitejše upravljanje s cenikom. Osnovna struktura XML predloge je prikazana v prilogi 3.

## 5.3. Opis izhodne, ChargeXML datoteke

Generiranje ChargeXML datoteke je zadnji korak v postopku uvoza cenika. Generiramo jo z modulom ChargeXML.exe. Vsebuje tarife, ki se uporabljajo za zaračunavanje izhodnih stroškov v aplikaciji Traffic<sup>5</sup>.

### 5.3.1. Struktura

Če poenostavljeno opišem strukturo ChargeXML datoteke, ta vsebuje območja (element Zones), sheme (ChargeSchemes) po katerih se zaračunava klice ter relacije (ChargeRelations) med območji s pripadajočo shemo. Torej, klic na določeni relaciji se zaračuna po veljavni shemi.

Glej prilogo 4.

#### 5.3.1.1. Elementi

Pomembnejši elementi ChargeXML datoteke in njihovi atributi:

- TimeInterval
  - StartDay
  - StartTime
  - StopDay
  - StopTime

Atributa StartDay in StopDay lahko zavzemata bitne vrednosti iz množice {nedelja = 1, ponedeljek = 2, torek = 4, sreda = 8, četrtek = 16, petek = 32, sobota = 64}.

Atributa StartTime in StopTime pa lahko zavzemata celoštevilске vrednosti iz intervala [1, 96], kjer je 1 petnajstminutni interval od 00:00 do 00:15, 2 je interval od 00:15 do 00:30 in tako naprej, vse do zadnje vrednosti, 96, ki zajema časovni interval 23:45 do 00:00.

- SubTar1, SubTar2, SubTar3, SubTar4
  - CurrFactor
  - CurrScale

---

<sup>5</sup> Aplikacija Traffic uporablja produkte aplikacije PriceManager. To so nabavni in prodajni ceniki za zaračunavanje klicev.

Na podlagi vrednosti teh dveh atributov, se računa cena klica na določeni časovni interval (atribut CurrInterval).

Primer:

CurrInterval = "60", minutni interval

CurrFactor = "3000000"

CurrScale = "249"

Sledi, da je cena =  $3000000 * 10^{(-7)} = 3\text{€}$  na minuto.

## 6. Grafični prikaz rešitve

### 6.1. Postopek uvoza nabavnega cenika

Predpogoj za začetek postopka je, da vhodna Excel oziroma Word datoteka obstaja. Obstajati mora tudi predloga, preko katere bomo transformirali vhodno datoteko.

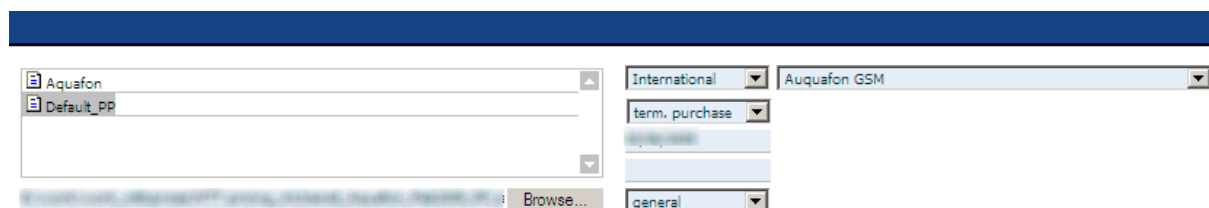
#### Naloga 1: Uvoz nabavnega cenika

Naloga opisuje postopek začetka uvoza nabavnega cenika. Izbrati je potrebno parametre prikazane na slikah 11 in 12<sup>6</sup>:

1. Klikni zavihek Import.
2. Izberi operaterja.
3. Izberi tip cenika (nabavni/prodajni)
4. Izberi splošni ali akcijski cenik.
5. Iz seznama izberi predlogo.
6. Klikni gumb Browse in izberi datoteko v dialogu Choose file.
7. Klikni ikono Read pricelist.



Slika 16: Izbira ikone

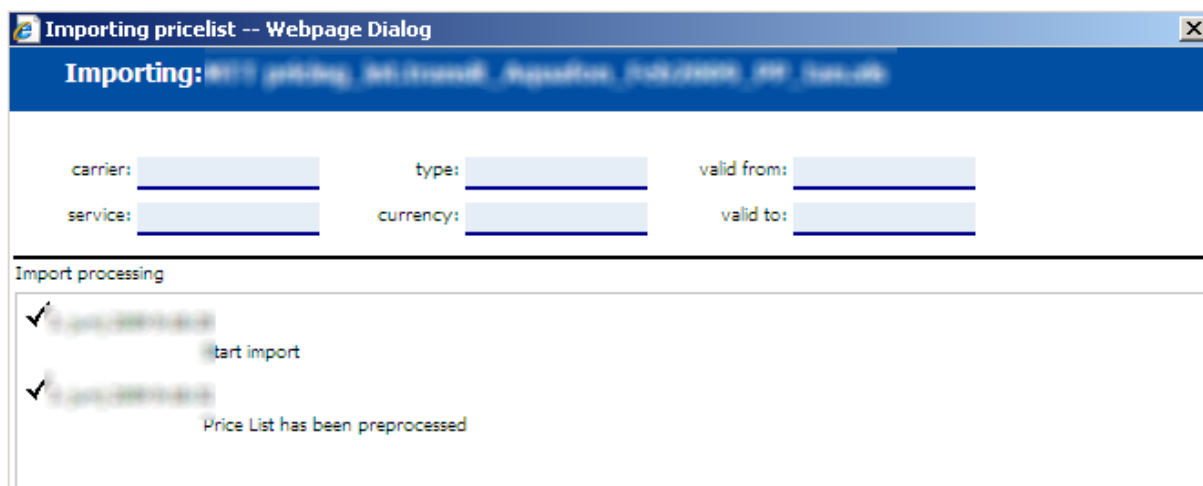


Slika 17: Polja za vnos parametrov nabavnega cenika

#### Pričakovan rezultat:

Slika 18 prikazuje situacijo, ko se vhodni cenik uspešno preslika preko XML predloge v interni format cenika.

<sup>6</sup> Besedilo na tej sliki in še na nekaterih drugih je namerno prekrito.



**Slika 18: Vhodni cenik je uspešno preslikan preko XML predloge**

## Naloga 2: Preverjanje vhodnega cenika

V postopek je vključeno tudi preverjanje vhodnega cenika, in sicer se preverja podvojene prefikse, podvojene destinacije in druge parametre glede na cenik, ki je že v aplikaciji.

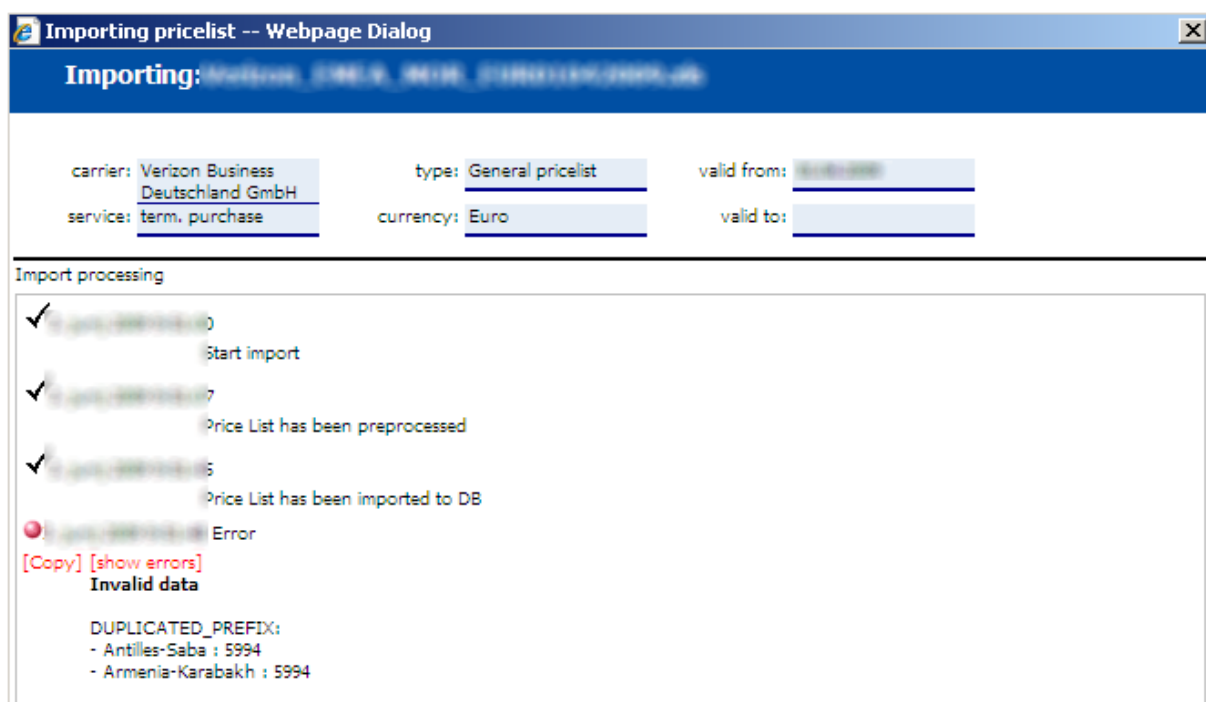
### Pričakovan rezultat:

Postopek se nadaljuje.

### Nepričakovan rezultat:

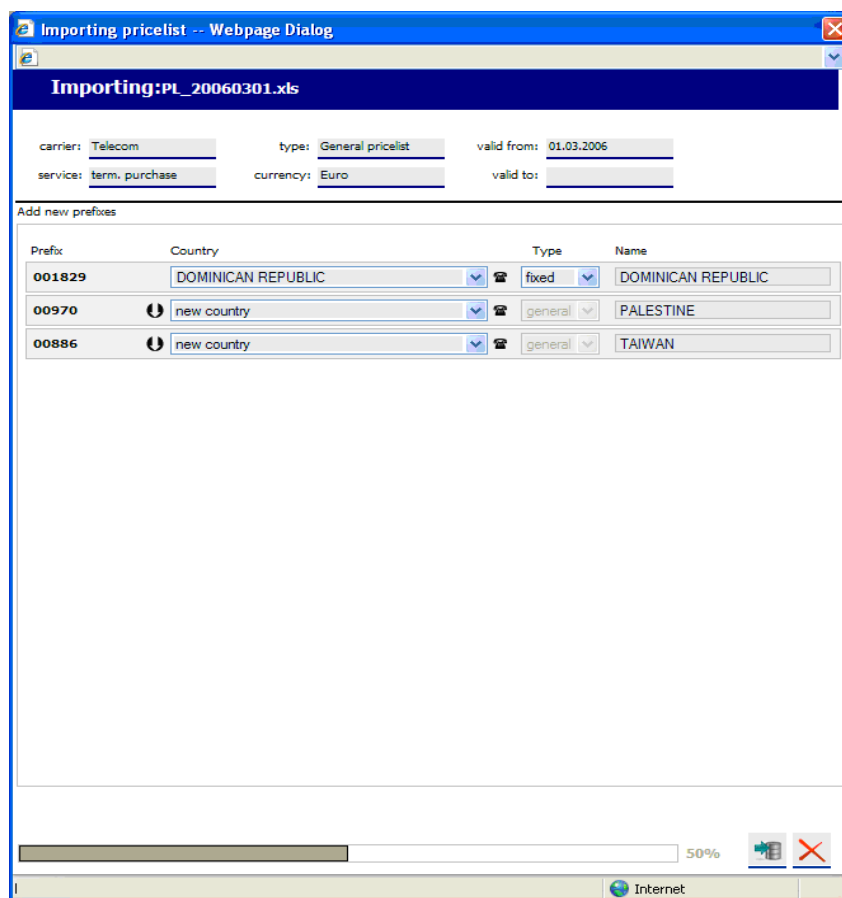
V primeru nekonsistentnosti v podatkih, se prikaže dialog na sliki 19. Napake v vhodnem ceniku je potrebno odpraviti in postopek uvoza ponoviti.

Dialog omogoča, da se opis napake kopira ([Copy]) v odložišče za lažje analiziranje napake. Omogoča tudi lociranje ([show errors]) napak v vhodnem ceniku. Celice, kjer so nekonsistentni podatki, obarva. Ceniki so zelo veliki, zato je s tem omogočeno lažje iskanje in preverjanje pri postopku analize cenika.




Slika 19: Napaka - vhodni cenik je nekonsistenten

### Naloga 3: Razreševanje novih prefiksov



Slika 20: Dialog novih prefiksov

Kot prikazuje slika 20, sta bila najdena dva nova prefiksa, ki ju še ni v sistemu. Problem mora biti rešen pred nadaljevanjem postopka uvoza.

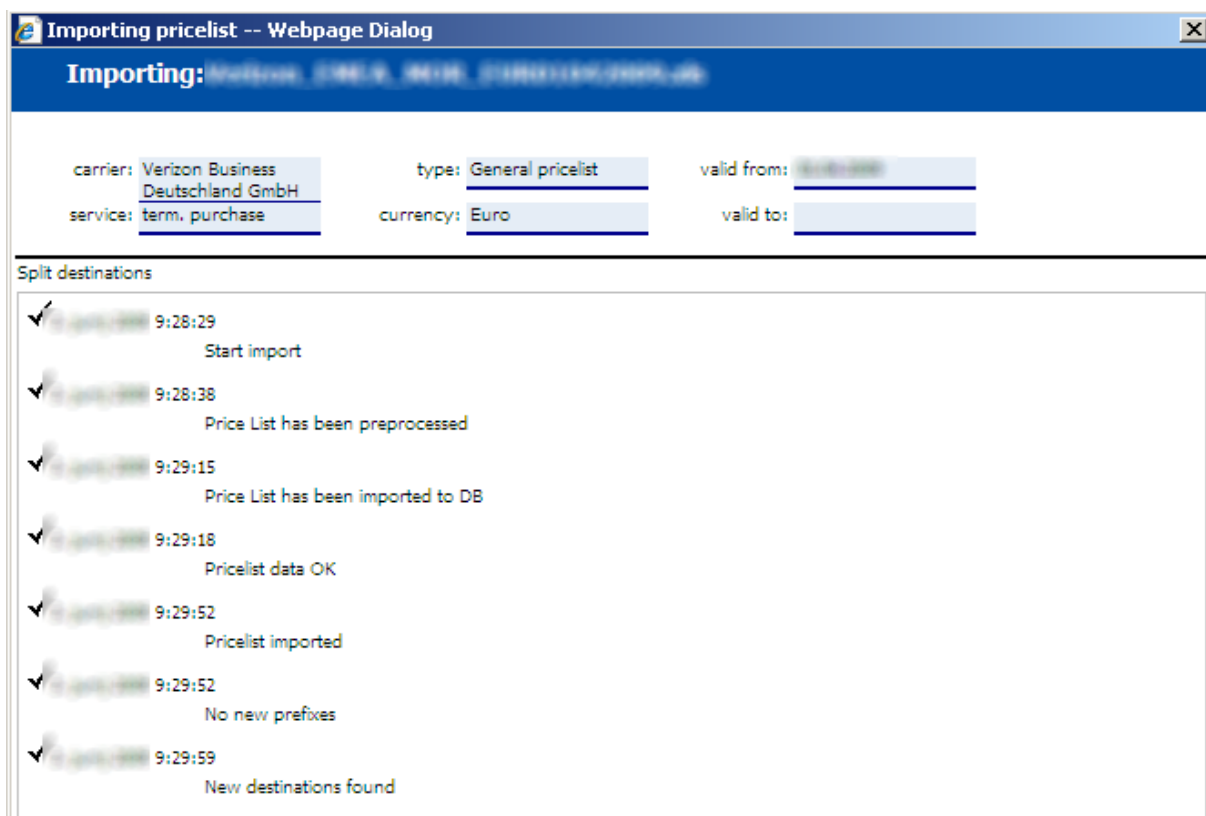
1. Prekini branje s pritiskom na gumb .
2. Klikni zavihek Prefixes, nato ikono Manage master destinations.
3. Kreiraj nove destinacije.
4. Ponovi postopek uvoza.

### Pričakovan rezultat:

Razrešeni prefiksi se ne pojavijo več v dialogu.

### Naloga 4: V vhodnem ceniku se najdejo nove destinacije

Med samim postopkom uvoza lahko pride tudi do izjem, slika 21, ko so v vhodnem ceniku najdene destinacije, ki jih v obstoječih cenikih še ni.



**Slika 21: V vhodnem ceniku so najdene nove destinacije**

### Naloga 5: Razreševanje novih destinacij

Importing pricelist -- Webpage Dialog

**Importing:**

carrier: Verizon Business  
Deutschland GmbH

type: General pricelist

valid from: 2010-01-01

service: term, purchase

currency: Euro

valid to:

Split destinations

Master table region	Pricelist prefix	Type	Price	Templates
AFGHANISTAN-CELL- AFGHANISTAN-MOBILE (AT)	Afghanistan-Mobile-Roshan	mobile	Price: 0.1532	

**Slika 22: Dialog novih destinacij**

Na sliki 22 je prikazana nova destinacija, ki se je pojavila v vhodnem ceniku:

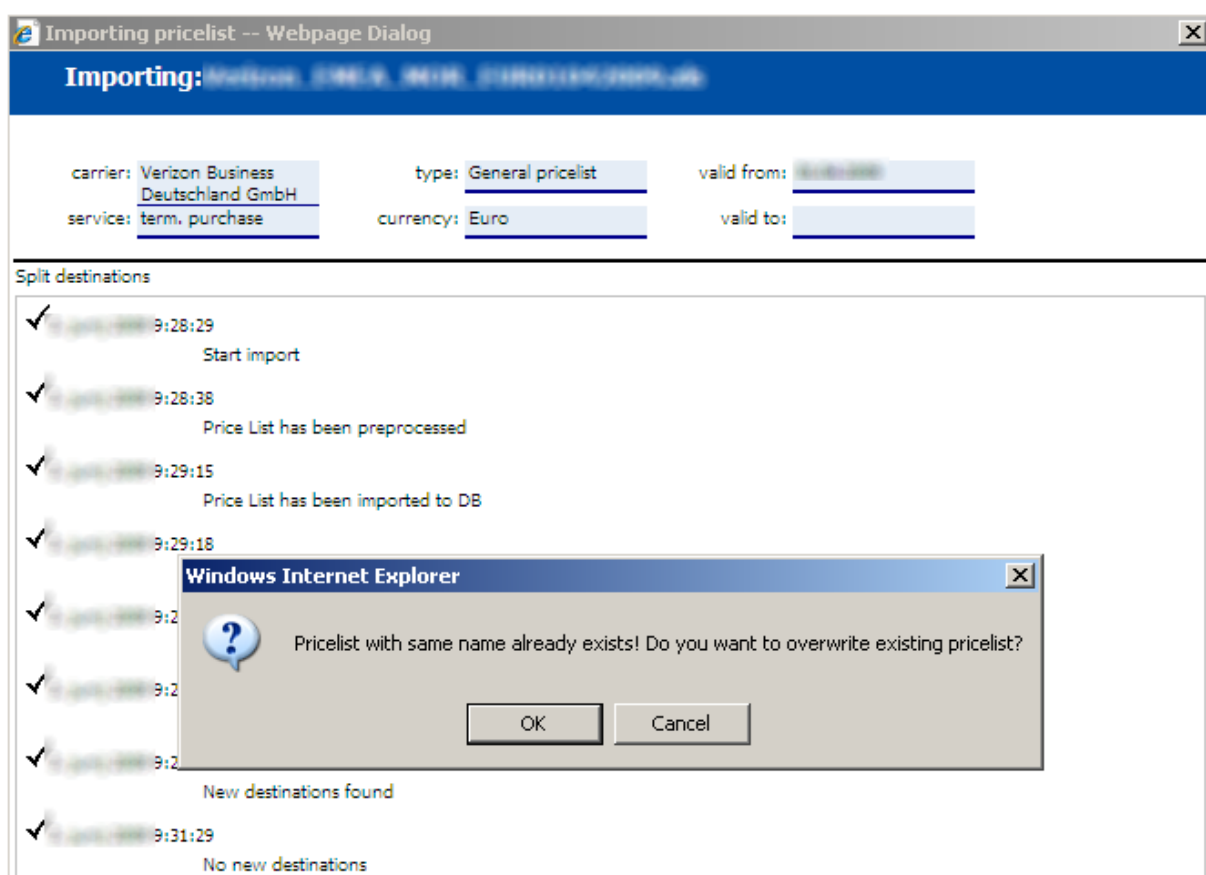
1. Če je tip destinacije nerazrešen, seznam tipov je obarvan rdeče, je potrebno izbrati ustrezen tip ročno.
2. Klikni ikono Read pricelist v spodnjem, desnem kotu.

### Pričakovan rezultat:

Postopek uvoza se nadaljuje.

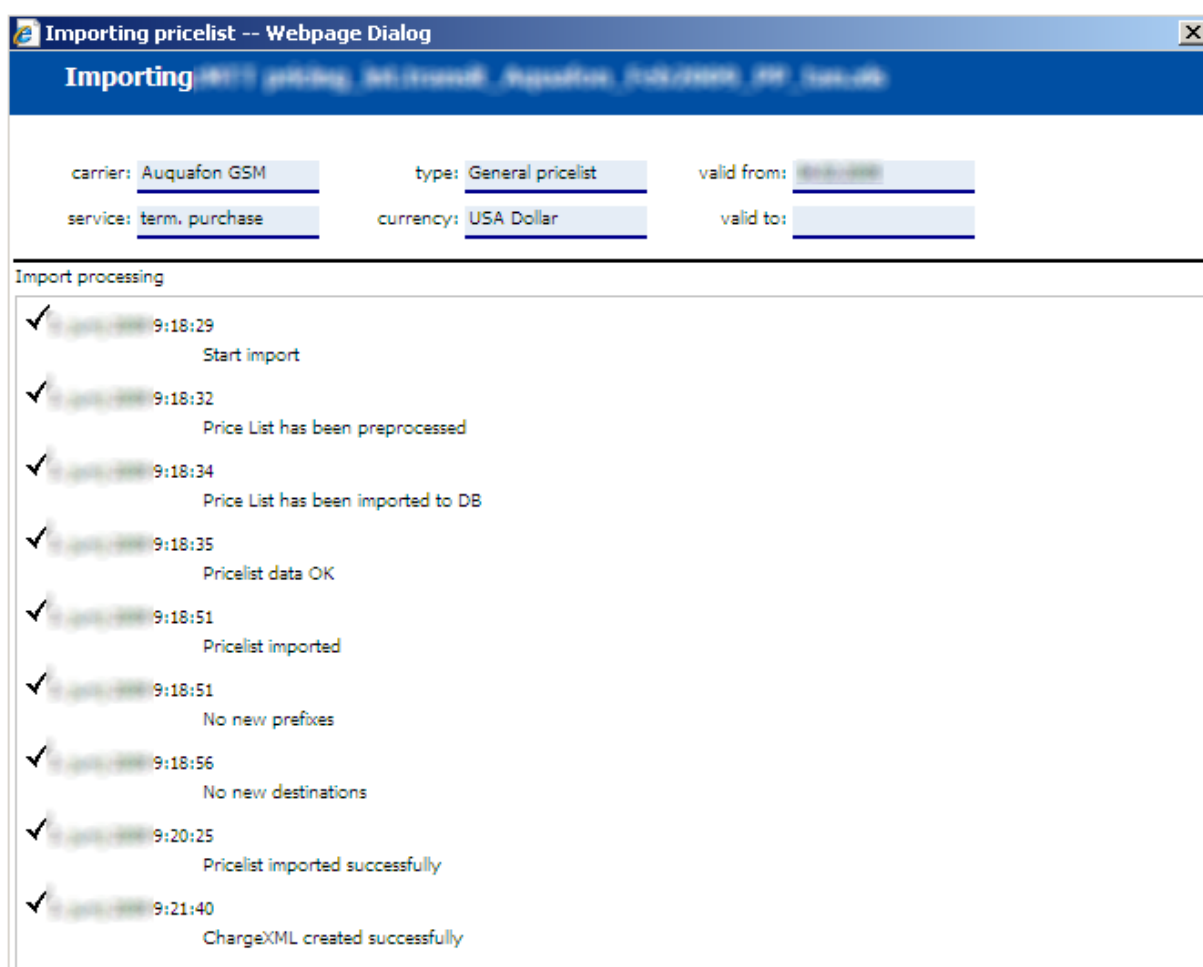
### Naloga 6: Zaključek postopka uvažanja cenika

Če cenik z istim imenom že obstaja, ga lahko prepíšemo, slika 23.



**Slika 23: Cenik s tem imenom že obstaja**

Ko je cenik uspešno shranjen, ostane samo še korak kreiranja ChargeXML datoteke, slika 24. S tem korakom je postopek uvoza cenika zaključen.



**Slika 24: Cenik je uspešno uvožen**

### Naloga 7: Pregledovanje uvoženega cenika

Po uspešno končanem uvozu, klikni ikono `Read pricelist` v spodnjem, desnem kotu dialoga.

### Pričakovan rezultat:

Na zaslону se prikaže pravkar uvoženi cenik, slika 25.

destination	price	valid from	valid to
1	ALBANIA-CELL-Albania - Mobile	0,052	28.11.2006
2	ALBANIA-CELL-ALBANIA MOBILE	0,052	28.11.2006
3	ALBANIA-CELL-ALBANIA_MOB	0,1201	28.11.2006
4	ALBANIA-CELL-ALBANIA_MOB-35568	0,1201	28.11.2006
5	ALBANIA-FIX-ALBANIA	0,052	28.11.2006
6	ALBANIA-FIX-Albania - Fixed	0,052	28.11.2006
7	ALBANIA-FIX-ALBANIA - Tirana	0,0295	28.11.2006
8	ALBANIA-FIX-Albania Tirana	0,0295	28.11.2006
9	ALBANIA-FIX-ALBANIA TIRANA (2)	0,0295	28.11.2006
10	ALBANIA-FIX-ALBANIA-TIRANA	0,0295	28.11.2006
11	ALBANIA-general-default	0,052	28.11.2006
12	ALGERIA-CELL-Algeria - Mob Other	0,086	28.11.2006
13	ALGERIA-CELL-Algeria - Mobile - Mobilis	0,0899	28.11.2006
14	ALGERIA-CELL-Algeria - Mobile Orascom	0,0797	28.11.2006
15	ALGERIA-CELL-Algeria - Mobile Orascom...	0,097	28.11.2006

Slika 25: Prikaz uvoženega nabavnega cenika

## 6.2. Uvoz nabavnega cenika ročno

Za ročni uvoz ne potrebujemo Excel oziroma Word datoteke, saj podatke za uvoz vnesemo ročno. Od običajnega avtomatskega uvoza se razlikuje le v formatu danih vhodnih podatkov.

Kot kaže slika 26, pri ročnem vnosu cenika lahko vnašamo poljubne destinacije in prefikse, ki pripadajo tej destinaciji.

Region	Prefix	Valid from	Valid to	1.price	1.tariff	2.price	2.tariff	3.price
Test	386	05.05.2009		0.12	Common 24h tariff			
		05.05.2009						

Slika 26: Dialog za ročni vnos nabavnega cenika

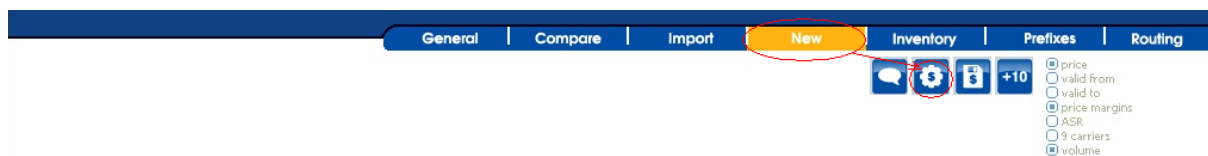
## 6.3. Postopek kreiranja prodajnega cenika

### Naloga 1: Zagon čarovnika za kreiranje cenika

Slika 27 prikazuje lokacijo čarovnika za kreiranje prodajnega cenika:

1. Klikni zavihek New.

2. Klikni ikono Auto generate selling pricelist.



**Slika 27: Lokacija ikone za kreiranje prodajnega cenika**

### Pričakovan rezultat:

Na sliki 28 je prikazan dialog za kreiranje prodajnega cenika. Uporabnik mora najprej izbrati opcije v področju dialoga general, potem sledijo opcije iz področja advanced calculation in kot zadnje opcije carrier related calculation, s katerimi prilagodimo končno formulo za izračun cen v ceniku.



**Slika 28: Dialog za kreiranje prodajnega cenika**

### Naloga 2: Kreiranje prodajnega cenika

Najprej iz seznama izberemo vrsto algoritma za izračun cen. Informacija o pozicijah operaterjev je dosegljiva v usmerjevalni tabeli. Aplikacija jemlje cene iz destinacij, ki so shranjene v korenskem direktoriju in ignorira vsak umetno narejen poddirektorij. Na voljo so naslednji algoritmi:

- |                                   |  |
|-----------------------------------|--|
| • <i>Max from routing</i>         | maksimalna cena operaterjev            |
| • <i>Max first 3 from routing</i> | maksimalna cena prvih treh operaterjev |
| • <i>Max first from routing</i>   | prva maksimalna cena operaterjev       |
| • <i>Min first from routing</i>   | prva minimalna cena operaterjev        |
| • <i>Empty pricelist</i>          | obstoječe cene niso uporabljene        |

V drugem koraku se, s pomočjo področja advanced calculations, osnovna cena spremeni glede na formulo (8):

$$cena = osnovna\_cena * (1 + \% / 100) + E \quad (8)$$

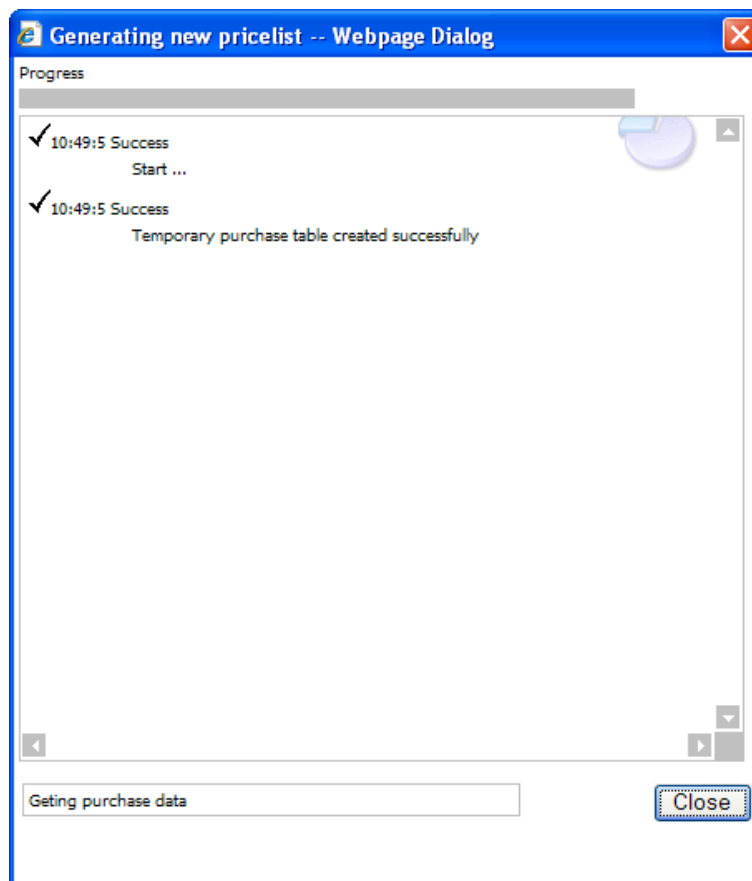
Primer:  $osnovna\_cena = 1$ ,  $\% = -5$ ,  $E = 0,2$ ;  $cena = 1,15$ .

Kot tretje, se lahko pri izračunu cene upošteva tudi kvaliteta izbranega operaterja, kar je nastavljivo na področju carrier related calculations z izbiro potrditvenih polj use minimum price, use profit margin.

Ko so vse opcije izbrane, klikni na gumb Create new pricelist .

### Pričakovan rezultat:

Slika 29 prikazuje dialog za spremljanje trenutnega koraka pri postopku kreiranja prodajnega cenika. Ko se kreiranje uspešno zaključi, se gumb **Close** na dialogu spremeni v gumb **OK**.



**Slika 29: Dialog napredka postopka kreiranja cenika**

### Naloga 3: Prikaz kreiranega cenika

Klikni na gumb **OK** v spodnjem, desnem kotu dialoga.

## Pričakovan rezultat:

Slika 30 prikazuje kreiran prodajni cenik. Cenik vsebuje dve vrstici glave cenika in eno ali več postavk cenika. Zgornji vrstica glave cenika vsebuje nekaj ikon, ime cenika in seznam tarif. Ikona za filtriranje, destination tekst in oznake stolpcev cenika so v spodnji vrstici glave cenika. Vsaka vrstica postavke je sestavljena iz zaporedne številke, imena destinacije, cene, začetnega datuma in končnega datuma veljavnosti cenika.

destination	price(USD)	last price	base price	last - base	1st choice	mult.(%)	add	calc(%)	valid from	volume	cost
1 Afghanistan	0,1855	0,2019	0,1669	0,0384	0,1635	10	0,005	10		671	10
2 Afghanistan mobile_AT	0,2123	0,2222	0,1911	0,0325	0,1897	10	0,005	10		19	
3 Afghanistan mobile_AWCC	0,21	0,2192	0,189	0,0304	0,1888	10	0,005	10		865	16
4 Afghanistan mobile_Etisalat	0,2145	0,2192	0,193	0,0247	0,1945	10	0,005	10		423	8
5 Afghanistan mobile_MTN	0,2122	0,2268	0,191	0,0358	0,191	10	0,005	10		40.534	7.74
6 Afghanistan mobile_Others	0,1875	0,2268	0,1688	0,2268	0,1635	10	0,005	10		0	
7 Afghanistan mobile_Roshan	0,2284	0,2268	0,2055	0,0213	0,2059	10	0,005	10		24.981	5.13
8 Afghanistan mobile_Sveta1	0,1657	0	0,1657	0	0,1635	0	0	0		0	
9 Albania	0,0511	0,0598	0,046	0,0138	0,046	10	0,005	11		379	1
10 Albania mobile	0,247	0,2525	0,2223	0,0302	0,228	10	0,005	10		46.720	10.38
11 Albania mobile_roaming	0,2404	0,2525	0,2164	0,2525	0,2142	10	0,005	10		0	
12 Algeria	0,0643	0,0829	0,0578	0,0252	0,0577	10	0,005	10		1.449	6
13 Algeria mobile_Algeria Telecom	0,1327	0,1375	0,1194	0,018	0,1195	10	0,005	10		720	6
14 Algeria mobile_Orascom	0,146	0,1472	0,1314	0,0184	0,1314	10	0,005	10		1.369	17
15 Algeria mobile_Others	0,0643	0,1472	0,0578	0,1472	0,0577	10	0,005	10		0	
16 Algeria mobile_Wataniya	0,1367	0,1312	0,123	0,0082	0,123	10	0,005	10		62.515	7.6
17 American Samoa	0,0445	0,0694	0,0395	0,0338	0,0356	10	0,005	14		57	

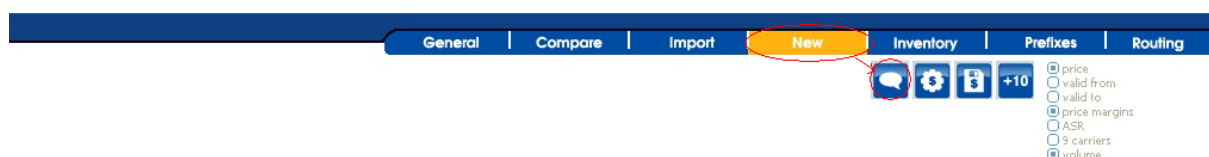
Slika 30: Kreiran prodajni cenik

## 6.4. Postopek kreiranja prodajnega cenika preko dialoga za upravljalno maržo

### Naloga 1: Zagon čarovnika za dodajanje upravljalne marže

Slika 31 prikazuje lokacijo čarovnika za kreiranje prodajnega cenika z dodajanjem upravljalne marže:

1. Klikni zavihek New.
2. Klikni ikono Invoke Manager margins.



Slika 31: Lokacija dialoga za dodajanje upravljalne marže


## Pričakovan rezultat:

Slika 32 prikazuje dialog za kreiranje prodajnega cenika z dodajanjem upravljalke marže.



Slika 32: Dialog za dodajanje upravljalke marže

## Naloga 2: Dodajanje upravljalke marže

1. Izberi operaterja.
2. Izberi enega od operaterjevih obstoječih cenikov.
3. Vnesi vrednosti za formulo (8).
4. Klikni 

## Pričakovan rezultat:

Rezultat, na sliki 33, je prodajni cenik .

Common 24h tariff													1,276
destination	price(USD)	last price	base price	buy cost	base(%)	buy(%)	base(add)	buy(add)	calc.(%)	Σvolume	cost		
1 Afghanistan	0,2221	0	0,2019	0,1635	10	0	0	0	26	736	12		
2 Afghanistan mobile_AT	0,2444	0	0,2222	0,1965	10	0	0	0	20	43			
3 Afghanistan mobile_AWCC	0,2411	0	0,2192	0,1828	10	0	0	0	24	1.876	34		
4 Afghanistan mobile_Etisalat	0,2411	0	0,2192	0,1877	10	0	0	0	22	460	8		
5 Afghanistan mobile_MTN	0,2495	0	0,2268	0,1967	10	0	0	0	21	2.499	45		
6 Afghanistan mobile_Others	0,2495	0	0,2268	0,1955	10	0	0	0	22	1			
7 Afghanistan mobile_Roshan	0,2495	0	0,2268	0,1969	10	0	0	0	21	3.886	7		
8 Albania	0,0658	0	0,0598	0,0488	10	0	0	0	26	125			
9 Albania mobile	0,2778	0	0,2525	0,2162	10	0	0	0	22	4.665	1.00		
10 Albania mobile_roaming	0,2778	0	0,2525	0,2155	10	0	0	0	22	8			
11 Algeria	0,0912	0	0,0829	0,0561	10	0	0	0	39	1.014	5		
12 Algeria mobile_Algeria Telecom	0,1488	0	0,1353	0,1204	10	0	0	0	19	738	6		
13 Algeria mobile_Orascom	0,1591	0	0,1446	0,1257	10	0	0	0	21	1.016	12		
14 Algeria mobile_Others	0,1591	0	0,1446	0	10	0	0	0	0	0			
15 Algeria mobile_Wataniya	0,1443	0	0,1312	0,1184	10	0	0	0	18	1.023	12		
16 American Samoa	0,0763	0	0,0694	0,0341	10	0	0	0	55	81			

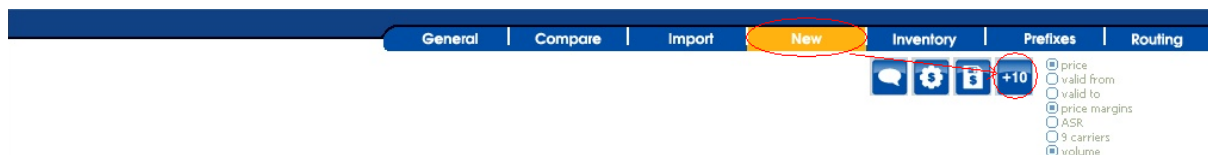
Slika 33: Prodajni cenik z dodano upravljalno maržo

## 6.5. Postopek dodajanja marže

### Naloga 1: Zagon čarovnika za dodajanje marže

Slika 34 prikazuje lokacijo dialoga za dodajanje marže za posameznega operaterja in destinacijo:

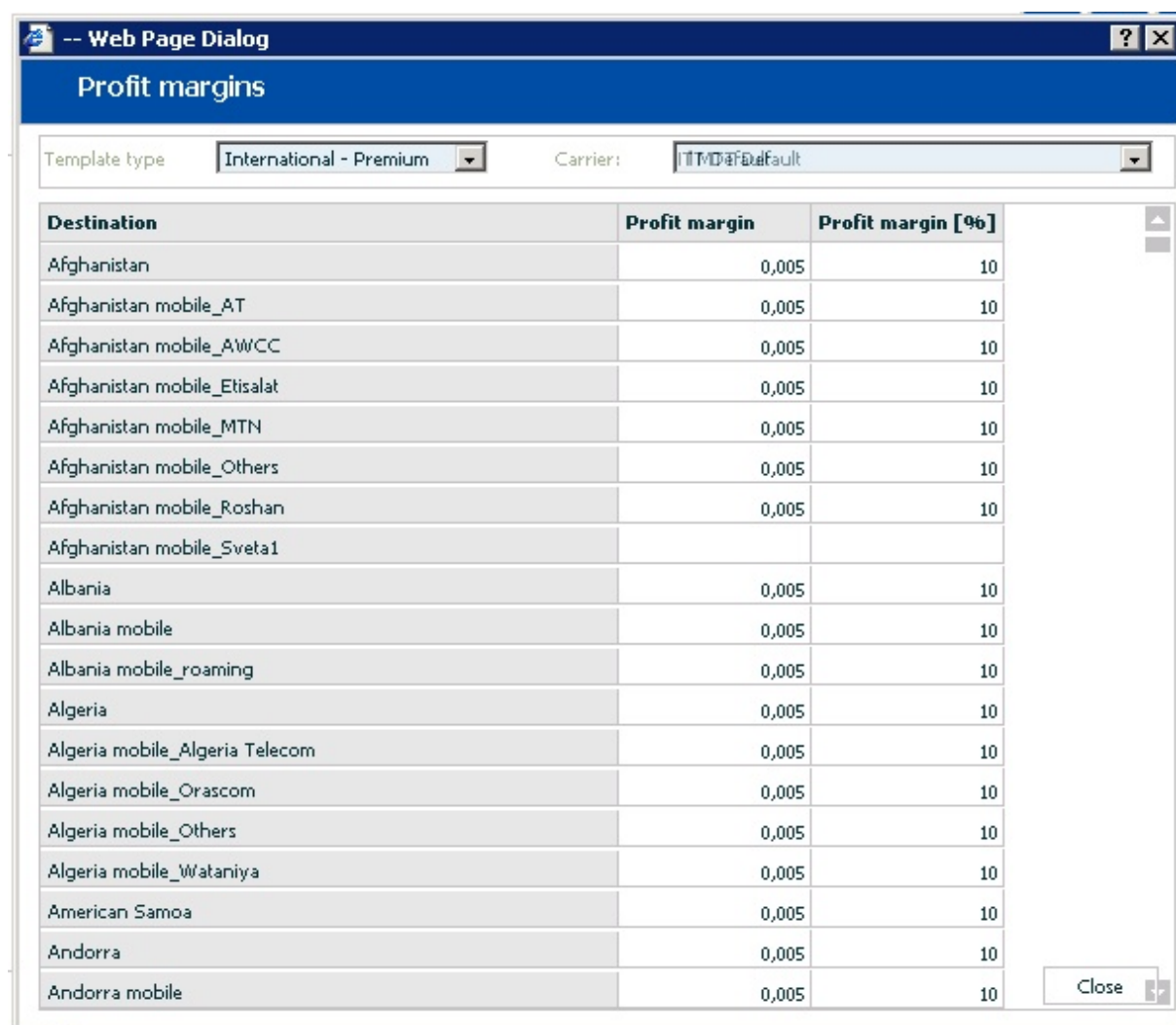
1. Klikni zavihek `New`.
2. Klikni ikono `Profit margins`.



**Slika 34: Lokacija dialoga za dodajanje marže**

### Pričakovan rezultat:

Za poljuben trojček, tip predloge, operater in destinacija, lahko dodamo maržo, ki jo potem upoštevamo pri izračunu prodajne cene. Na dialogu, slika 35, je potrebno izbrati operaterja ter določenim destinacijam dodati maržo. Marža lahko zavzame dve vrednosti, absolutno, polje `Profit margin` ali relativno, polje `Profit margin[%]`. V primeru, da ima določena destinacija obe vrednosti, se vzame maksimalna vrednost.



Web Page Dialog

Profit margins

Template type: International - Premium Carrier: TTMDFDfault

Destination	Profit margin	Profit margin [%]
Afghanistan	0,005	10
Afghanistan mobile_AT	0,005	10
Afghanistan mobile_AWCC	0,005	10
Afghanistan mobile_Etisalat	0,005	10
Afghanistan mobile_MTN	0,005	10
Afghanistan mobile_Others	0,005	10
Afghanistan mobile_Roshan	0,005	10
Afghanistan mobile_Sveta1		
Albania	0,005	10
Albania mobile	0,005	10
Albania mobile_roaming	0,005	10
Algeria	0,005	10
Algeria mobile_Algeria Telecom	0,005	10
Algeria mobile_Orascom	0,005	10
Algeria mobile_Others	0,005	10
Algeria mobile_Wataniya	0,005	10
American Samoa	0,005	10
Andorra	0,005	10
Andorra mobile	0,005	10

Close

Slika 35: Dialog za dodajanje marže

## 7. Sklepne ugotovitve

V nalogi je opisan le del celotne aplikacije za upravljanje s ceniki. Aplikacija je namenjena tudi optimizaciji prodajnih cen in podpira še druge funkcije, ki so nujne za uspešno delovanje glavnih funkcij. Tehnično gradivo na to temo je zelo težko dosegljivo oziroma ni dosegljivo v okviru širše javnosti. Podjetja, ki razvijajo te vrste aplikacij, jih razvijajo po svojem modelu in specifikacijah. Svojih ugotovitev in rešitev pa ne objavljajo v gradivu ali študijah, dostopnih širši strokovni javnosti. K temu pripomore tudi sam naročnik, s svojimi specifičnimi zahtevami in željami, da te informacije ostanejo zaupne. Iz teh razlogov, sem veliko uporabnih informacij uspel poiskati na spletu oziroma v podjetju, kjer smo razvijali aplikacijo.

Če se osredotočim na samo izvedbo postopka uvoza in generiranja cenikov, lahko ugotavljam, da bi bilo možno ta postopek še optimizirati in s tem pohitriti. Še posebej postopek uvoza, ki je narejen na podlagi stare arhitekture, za nekega drugega naročnika in trenutno še ni popolnoma optimiziran.

Funkcionalno je aplikacija zelo napredna, saj vsebuje zelo širok nabor pregledov in ostalih konfiguracijskih obrazcev, ki jih, po mojih izkušnjah, druge aplikacije nimajo.

## 8. Dodatek A

### 8.1. Priloge

#### 8.1.1. Priloga 1: Nabavni cenik v Word formatu

Effective date: February 01' 2008

Professional Communications LTD rates for MTT

<i>Destination</i>	Codes	Rate,USD/min
Azerbaijan	994	0,115
Azerbaijan, Baku	994 12	0,056
Azerbaijan, mobile (меньше 1 500 000 мин.)	994: 50,55,51,40,70, 994 60 540 - new	0,135
Azerbaijan, mobile (больше 1 500 000 мин.)	994: 50,55,51,40,70, 994 60 540 - new	0,117

#### 8.1.2. Priloga 2: Del nabavnega cenika v Excel formatu

**BT Germany monthly offer:** APR 2008  
**Valid from:** 01/04/08  
**Monthly offer to:** Russia Mtt

##### Customer Data

Customer: Russia Mtt  
 Voice Specialist: Fabrizio Bolognesi  
 Rate effective from: 01/04/08

Price for all mobile destinations are subjected to change in accordance with the agreed service terms and conditions which are available from the sender on request.

Price for all other destinations are subjected to change in accordance with the agreed service terms and conditions which are available from the sender on request.

All prices are in EURO and are exclusive of any local taxes.

Traffic to any unspecified destinations, including non-geographic number ranges, will be treated as unauthorised traffic and billed accordingly. Unless otherwise agreed the standard billing rate for unauthorised traffic is Euro 5.00/minute. Please contact the sender if you cannot see the destination you need.

The prices offered are for IDD service only and exclude 64k clear channel ISDN, operator and collect calls.

All tariffs are strictly  
confidential

**International  
Destinations**

<b>Destinations</b>	<b>Time of Day</b>	<b>Dial Code</b>	<b>Effective Date</b>	<b>New Offer EURO/permin</b>	<b>Last Offer EURO/permin</b>	<b>Change</b>
AFGHANISTAN	24hr	93	01/04/2008	0,1189	0,1386	Decrease
AFGHANISTAN						
AREEBA MOBILE	24hr	9377	01/04/2008	0,1149	0,1261	Decrease
AFGHANISTAN						
MOBILE	24hr	9370	01/04/2008	0,1296	0,1420	Decrease
AFGHANISTAN						
MOBILE	24hr	9371	01/04/2008	0,1296	0,1420	Decrease
AFGHANISTAN						
MOBILE	24hr	9372	01/04/2008	0,1296	0,1420	Decrease
AFGHANISTAN						
MOBILE	24hr	9373	01/04/2008	0,1296	0,1420	Decrease
AFGHANISTAN						
MOBILE	24hr	9374	01/04/2008	0,1296	0,1420	Decrease
AFGHANISTAN						
MOBILE	24hr	9375	01/04/2008	0,1296	0,1420	Decrease
AFGHANISTAN						
MOBILE	24hr	9376	01/04/2008	0,1296	0,1420	Decrease
AFGHANISTAN						
MOBILE	24hr	9378	01/04/2008	0,1296	0,1420	Decrease
AFGHANISTAN						
ROSHAN MOBILE	24hr	9379	01/04/2008	0,1347	0,1510	Decrease
ALASKA	24hr	1907	01/04/2008	0,0165	0,0177	Decrease
ALBANIA	24hr	355	01/04/2008	0,0362	0,0456	Decrease
ALBANIA OTHER						
MOBILE	24hr	35567	01/04/2008	0,1550	0,1434	Increase
ALBANIA OTHER						
MOBILE	24hr	35568	01/04/2008	0,1550	0,1434	Increase
ALBANIA						
TIRANA	24hr	3554	01/04/2008	0,0344	0,0371	Decrease
ALBANIA						
VODAFONE						
MOBILE	24hr	35569	01/04/2008	0,1550	0,1518	Increase
ALGERIA	24hr	213	01/04/2008	0,0885	0,0953	Decrease
ALGERIA						
DJEZZY MOBILE	24hr	2137	01/04/2008	0,0980	0,0940	Increase
ALGERIA OTHER						
MOBILE	24hr	2136	01/04/2008	0,0900	0,0969	Decrease
ALGERIA OTHER						
MOBILE	24hr	21396	01/04/2008	0,0900	0,0969	Decrease
ALGERIA						
WATANIYA						
MOBILE	24hr	2135	01/04/2008	0,0972	0,1046	Decrease
AMERICAN						
SAMOA	24hr	1684	01/04/2008	0,0316	0,0532	Decrease
ANDORRA	24hr	376	01/04/2008	0,0130	0,0140	Decrease
ANDORRA						
MOBILE	24hr	3763	01/04/2008	0,1496	0,1669	Decrease
ANDORRA						
MOBILE	24hr	3764	01/04/2008	0,1496	0,1669	Decrease
ANDORRA						
MOBILE	24hr	3765	01/04/2008	0,1496	0,1669	Decrease
ANDORRA						
MOBILE	24hr	3766	01/04/2008	0,1496	0,1669	Decrease
ANGOLA	24hr	244	01/04/2008	0,0646	0,0696	Decrease

### 8.1.3. Priloga 3: Osnovna struktura XML predloge

```

<PriceListPreprocessor>
<PriceListConfiguration>
<Destinations>
  <Sources type="List" dataLinkingFormat="{0}">
  </Sources>
  <Rules>
  </Rules>
</Destinations>
<Prefixes>
  <Sources type="Dictionary" keyLinkingFormat="{0}" dataLinkingFormat="{1}">
  </Sources>
  <Rules>
  </Rules>
</Prefixes>
<Prices>
  <Price tariffId="7">
  <TariffPrice>
    <Sources type="Dictionary" keyLinkingFormat="{0}"
dataLinkingFormat="T{2}P{1}:T{4}P{3}:T{6}P{5}">
    </Sources>
    <Rules>
    </Rules>
  </TariffPrice>
  <TariffValidFrom>
<Sources type="Dictionary" keyLinkingFormat="{0}" dataLinkingFormat="{1}">
</Sources>
    <Rules>
    </Rules>
  </TariffValidFrom>
  <TariffValidTo>
<Sources type="Dictionary" keyLinkingFormat="{0}" dataLinkingFormat="{1}">
</Sources>
    <Rules>
    </Rules>
  </TariffValidTo>
</Price>
</Prices>
<Partner>
<Sources type="Scalar" dataLinkingFormat="{0}">
  </Sources>
  <Rules>
  </Rules>
</Partner>
<Currency>
  <Sources type="Scalar" dataLinkingFormat="{0}">
  </Sources>
  <Rules>
  </Rules>
</Currency>
<ValidFrom>

```

```

        <Sources type="Scalar" dataLinkingFormat="{0}">
        </Sources>
        <Rules>
        </Rules>
</ValidFrom>
<ValidTo>
        <Sources type="Scalar" dataLinkingFormat="{0}">
        </Sources>
        <Rules>
        </Rules>
</ValidTo>
</PriceListConfiguration>
</PriceListPreprocessor>

```

#### 8.1.4. Priloga 4: Osnovna struktura ChargeXML datoteke

```

<Root_Element>
<Zones>
<Zone Id="0" Name="Default" Mask=""/>
<Zone Id="1" Name="DefaultSame" Mask=""/>
<Zone Id="2" Name="DefaultDiff" Mask=""/>
</Zones>
<ChargeSchemes>
<Scheme Id="0" Name="Zone ID not found" Version="1" CurrUnit="8"
ChargeInterval="60">
<Record Id="0" TariffControl="1" CurrUnit="8" VoiceId="652" BillingInfo="0"
Name="">
<TimeInterval StartDay="1" StartTime="1" StopDay="64" StopTime="96"/>
<FreeMinute Type="0" Lower="0" Upper="0"/>
<CallSetup CurrFactor="0" CurrScale="0"/>
<CallAttempt CurrFactor="0" CurrScale="0"/>
<SubTar1 CurrFactor="0" CurrScale="0" Duration="0" TariffControl="0"/>
<SubTar2 CurrFactor="0" CurrScale="0" Duration="0" TariffControl="0"/>
<SubTar3 CurrFactor="0" CurrScale="0" Duration="0" TariffControl="0"/>
<SubTar4 CurrFactor="0" CurrScale="0" Duration="0" TariffControl="0"/>
</Record>
</Scheme>
<Scheme Id="1" Name="No Relation" Version="1" CurrUnit="8" ChargeInterval="60">
<Record Id="0" TariffControl="1" CurrUnit="8" VoiceId="200" BillingInfo="0"
Name="">
<TimeInterval StartDay="1" StartTime="1" StopDay="64" StopTime="96"/>
<FreeMinute Type="0" Lower="0" Upper="0"/>
<CallSetup CurrFactor="0" CurrScale="0"/>
<CallAttempt CurrFactor="0" CurrScale="0"/>
<SubTar1 CurrFactor="0" CurrScale="0" Duration="0" TariffControl="0"/>
<SubTar2 CurrFactor="0" CurrScale="0" Duration="0" TariffControl="0"/>
<SubTar3 CurrFactor="0" CurrScale="0" Duration="0" TariffControl="0"/>
<SubTar4 CurrFactor="0" CurrScale="0" Duration="0" TariffControl="0"/>
</Record>
</Scheme>
<Scheme Id="2" Name="No Relation" Version="1" CurrUnit="8" ChargeInterval="60">

```

```
<Record Id="0" TariffControl="1" CurrUnit="8" VoiceId="700" BillingInfo="0"
Name="">
<TimeInterval StartDay="1" StartTime="1" StopDay="64" StopTime="96"/>
<FreeMinute Type="0" Lower="0" Upper="0"/>
<CallSetup CurrFactor="0" CurrScale="0"/>
<CallAttempt CurrFactor="0" CurrScale="0"/>
<SubTar1 CurrFactor="0" CurrScale="0" Duration="0" TariffControl="0"/>
<SubTar2 CurrFactor="0" CurrScale="0" Duration="0" TariffControl="0"/>
<SubTar3 CurrFactor="0" CurrScale="0" Duration="0" TariffControl="0"/>
<SubTar4 CurrFactor="0" CurrScale="0" Duration="0" TariffControl="0"/>
</Record>
</Scheme>
</ChargeSchemes>
<ChargeRelations>
<RelationsCategory Id="0">
<Relation Id="0" Name="" ZoneA="0" ZoneB="X" SchemeId="Y"/>
</RelationsCategory>
</ChargeRelations>
<FreeMinutes LocalQuota="0" FreeMinuteCalls="0"/>
<HOLIDAYS/>
</Root_Element>
```

## 9. Literatura

- [1] O. Boylaud, G. Nicoletti, Regulation, market structure and performance in telecommunications, Pariz, 2000, str.21
- [2] Gartner, "Pricing optimization boosts revenue and earnings", v raziskavi z zaporedno številko G00143284, 22.september 2006
- [3] J.Holdershaw, P.Gendall, R.Garland, "The widespread use of odd pricing in the retail sector", v publikaciji Marketing Bulletin, 1997, številka 8, str. 53-58.
- [4] M. V. Marn, E. V. Roegner, C. C. Zawada, The price advantage, New Jersey: John Wiley & Sons, 2004
- [5] M.Mihelčič, Ekonomika poslovanja za inženirje, Ljubljana: Fakulteta za računalništvo in informatiko, 2005, pogl. 7.3.
- [6] M.Silič, M.Colnar, M.Krisper, EMRIS – Enotna metodologija razvoja informacijskih sistemov, Ljubljana: Vlada Republike Slovenije, Center Vlade RS za informatiko, 2000, 4.zvezek
- [7] B. Valančič, Liberalizacija in privatizacija telekomunikacijskega sektorja v Sloveniji: magistrsko delo, Ljubljana, 1998, str. 26-27
- [8] 2.člen direktive Evropske komisije o medomrežnem povezovanju in dostopu z dne 7.2.2002
- [9] Prevedeno iz:  
[http://en.wikipedia.org/wiki/Unified\\_Modeling\\_Language](http://en.wikipedia.org/wiki/Unified_Modeling_Language)
- [10] Povzeto iz:  
<http://en.allexperts.com/q/Marketing-Research-2308/Psychological-pricing.htm>