Univerzitet Singidunum

Vladislav Miškovic

SISTEMI ZA PODRŠKU ODLUČIVANJU

DECISION SUPPORT SYSTEMS



Beograd, 2024.

UNIVERZITET SINGIDUNUM

Vladislav Miškovic

SISTEMI ZA PODRŠKU ODLUČIVANJU

Treće izdanje

Beograd, 2024.

SISTEMI ZA PODRŠKU ODLUČIVANJU

Autor:
dr Vladislav Miškovic

Recenzenti:
dr Milan Milosavljević
dr Mladen Veniović
dr Boško Nikolić

Izdavač:
UNIVERZITET SINGIDUNUM
Beograd, Danijelova 32
www.singidunum.ac.rs

Za izdavača:
dr Milovan Stanišić

Tehnička obrada:
Vladislav Miškovic

Dizajn korica:
Aleksandar Mihajlović

Godina izdanja:
2024.

Tiraž:
300 primeraka

Štampa:
Caligraph, Beograd

ISBN: 978-86-7912-640-5

Copyright:
© 2024. Univerzitet Singidunum
Izdavač zadržava sva prava.
Reprodukcija pojedinih delova ili celine ove publikacije nije dozvoljeno.

Sadržaj

1 Uvod u sisteme za podršku odlučivanju 1
2 Osnove teorije odlučivanja 27
3 Modeliranje i analiza procesa donošenja odluka 54
4 Računarska podrška odlučivanju: koncept i tehnologija sistema 88
5 Poslovna inteligencija, skladištenje podataka i otkrivanje znanja 100
6 Složeno odlučivanje: grupno donošenje odluka 113
7 Inteligentni sistemi: podrška donošenju složenih odluka 127
8 Ekspertni sistemi 139
9 Metodi i tehnike mašinskog učenja 154
10 Inteligentna podrška elektronskoj trgovini 186
11 Implementacija sistema za podršku odlučivanju 207
Literatura 241
Prilozi 1

Sadržaj

1 Uvod u sisteme za podršku odlučivanju 1
1.1 Osnovni pojmovi 1
1.2 Proces donošenja poslovnih odluka 10
1.3 Modeliranje procesa donošenja odluka 14
1.4 Automatizacija podrške odlučivanju 15
1.4.1 Računarska podrška strukturiranom odlučivanju 16
1.4.2 Računarska podrška nestrukturiranom odlučivanju 19
1.4.3 Računarska podrška delimično strukturiranom odlučivanju 21
1.4.4 Softverski alati za razvoj sistema za podršku odlučivanju 22
1.5 Nastanak i razvoj sistema za podršku odlučivanju 24
2 Osnove teorije odlučivanja 27
2.1 Osnovni pojmovi 27
2.2 Modeli odlučivanja 30
2.3 Proces donošenja odluka 32
2.4 Statistička teorija odlučivanja 36
2.5 Teorija verovatnoće i matematička logika 37
2.6 Alternativne teorije odlučivanja 41
2.6.1 Faktori pouzdanosti 41
2.6.2 Teorija fazi skupova 45
3 Modeliranje i analiza procesa donošenja odluka 54
3.1 Modeli i modeliranje 54
3.2 Matematički modeli za podršku odlučivanju 56
3.2.1 Optimizacija i linearno programiranje 56
3.2.2 Tabele i stabla odlučivanja 70
3.2.3 Modeli redova čekanja 71
3.2.4 Opšti metodi rešavanja problema 73
3.2.5 Simulacioni modeli 76
3.3 Softverska podrška modeliranju 78
3.4 Primeri rešavanja strukturiranih problema 78
4 Računarska podrška odlučivanju: koncept i tehnologija sistema 88
4.1 Koncept i arhitektura sistema za podršku odlučivanju 88
4.2 Komponente sistema za podršku odlučivanju 90
4.2.1 Podsistem za upravljanje podacima 91
4.2.2 Podsistem za upravljanje modelima 91
4.2.3 Podsistem za upravljanje znanjem 93
4.2.4 Podsistem korisničkog interfejsa 93

4.3 Klasifikacija sistema za podršku odlučivanju 95
4.4 Prostorni sistemi za podršku odlučivanju 96
4.5 Primeri sistema za podršku odlučivanju 98
5 Poslovna inteligencija, skladištenje podataka i otkrivanje znanja 100
5.1 Osnovni pojmovi 100
5.2 Poslovna inteligencija 101
5.3 Skladištenje podataka 102
5.3.1 Razvoj skladišta podataka 105
5.3.2 Struktura skladišta (zvezda) 107
5.3.3 Softver za skladištenje 108
5.4 Istraživanje podataka 108
5.5 Upravljanje znanjem 111
5.6 Primeri sistema poslovne inteligencije 112
6 Složeno odlučivanje: grupno donošenje odluka 113
6.1 Proces grupnog odlučivanja 113
6.2 Modeli grupnog odlučivanja 114
6.2.1 Model višekriterijumskog odlučivanja 116
6.3 Sistemi za podršku saradnji i grupnom odlučivanju 120
6.3.1 Sistemi za indirektnu podršku grupnom odlučivanju 120
6.3.2 Integrisani alati za podršku grupnom odlučivanju 120
6.3.3 Sistemi za direktnu podršku grupnom odlučivanju 121
6.3.4 Podrška kreativnosti i stvaranju ideja 121
6.3.5 Primeri 121
6.4 Primeri sistema za grupni rad i odlučivanje 122
7 Inteligentni sistemi: podrška donošenju složenih odluka 127
7.1 Osnovni pojmovi veštačke inteligencije 127
7.2 Inteligentni sistemi 131
7.3 Istraživanje podataka i mašinsko učenje 133
7.4 Inteligentni interfejsi 134
7.5 Inteligentni agenti 135
8 Ekspertni sistemi 139
8.1 Nastanak ekspertnih sistema 139
8.2 Struktura ekspertnih sistema 142
8.2.1 Baza znanja (Knowledga Base) 143
8.2.2 Mehanizam zaključivanja (Inference Engine) 144
8.2.3 Algoritam RETE 149
8.2.4 Primena Bajesovog pravila 149
8.3 Vrste ekspertnih sistema 150

8.4 Alati za razvoj ekspertnih sistema ... 151
8.5 Primeri ekspertnih sistema za podršku odlučivanju ... 151
9 Metodi i tehnike mašinskog učenja ... 154
9.1 Pojam mašinskog učenja ... 154
9.2 Produkciona pravila i stabla odlučivanja ... 157
9.2.1 Metodi induktivnog učenje stabala odlučivanja ... 157
9.2.2 Metodi induktivnog učenja pravila odlučivanja ... 160
9.3 Induktivno učenje i procena tačnosti naučenog znanja ... 162
9.4 Veštačke neuronske mreže (ANN) ... 166
9.5 Metod nosećih vektora (SVM) ... 173
9.6 Zaključivanje po analogiji (CBR) ... 174
9.7 Programski alati mašinskog učenja i istraživanja podataka ... 179
10 Inteligentna podrška elektronskoj trgovini ... 186
10.1 Elektronska trgovina ... 186
10.2 Inteligentni sistemi u e-trgovini ... 192
10.3 Inteligentni interfejsi u e-trgovini ... 201
10.4 Primeri inteligentne podrške e-trgovini ... 203
11 Implementacija sistema za podršku odlučivanju ... 207
11.1 Uvod u implementaciju sistema za podršku odlučivanju ... 207
11.2 Softverske tehnologije i alati za implementaciju ... 213
11.3 Sistemi zasnovani na elektronskim tabelama ... 214
11.4 Implementacija ekspertnih sistema ... 214
11.5 Sistemi mašinskog učenja ... 220
11.6 Menadžment razvoja sistema za podršku odlučivanju ... 230
11.6.1 Podsetnik: projekt razvoja softvera ... 230
11.6.2 Faze razvoja sistema za podršku odlučivanju ... 234
11.6.3 Kriterijumi za ocenjivanje sistema za podršku odlučivanju ... 238
Literatura ... 241
Prilozi ... 1

Popis slika

Sl. 1. Opšta struktura sistema za podršku odlučivanju 6
Sl. 2. Primeri finansijskih kalkulatora..... 7
Sl. 4. Predlog odluke o načinu kupovine automobila (polovni)..... 8
Sl. 5. Komponente sistema za podršku zasnovanog na elektronskim tabelama..... 9
Sl. 6. Primer sistema za podršku odlučivanju (DSS) zasnovanog na Excel tabelama .. 10
Sl. 7. Faze procesa odlučivanja..... 12
Sl. 8. Faze procesa donošenja odluke (Sajmon, 1960)..... 13
Sl. 9. Problem linearnog programiranja sa dve promenljive..... 17
Sl. 10. Primer transportnog problema..... 17
Sl. 11. Deo tekstualnog dijaloga lekara i ekspertnog sistema MYCIN..... 21
Sl. 12. Sistem poslovne inteligencije Pentaho 22
Sl. 13. Radni list sistema Excel i funkcije dodatka Palisade@RISK..... 23
Sl. 14. Ilustracija suviše malog izbora (a) i suviše velikog izbora (b) 29
Sl. 15. Mogućnost izbora i zadovoljstvo potrošača 30
Sl. 16. Relacije između faza i rutina procesa odlučivanja..... 31
Sl. 17. Poređenje dekompozicije procesa odlučivanja različitih autora..... 31
Sl. 18. Primer (a) opšte matrice odlučivanja i (b) matrice koristi 33
Sl. 19. Vrednosti istinitosti u (a) Bulovoj logici i (b) višeznačnoj logici..... 47
Sl. 20. Funkcija pripadnosti skupu visok čovek za (a) klasični i (b) fazi skup 48
Sl. 21. Skupovi nizak, prosečan i visok čovek za (a) klasične i (b) fazi skupove 49
Sl. 22. Fazi skupovi s fazi kvalifikatorom veoma..... 50
Sl. 23. Primer klasičnih produkcionih pravila i ekvivalentnih fazi pravila..... 51
Sl. 24. Fazi skup speed 51
Sl. 25. Opšta struktura kvantitativnih modela i rešavanje problema..... 56
Sl. 26. Struktura kvantitativnih modela i rešavanje problema 57
Sl. 27. Grafičko rešavanje problema linearnog programiranja 60
Sl. 28. Prvi deo računarskog modela: Excel tabela..... 65
Sl. 30. Rešenje problema linearnog programiranja metodom Solver..... 66
Sl. 29. Drugi deo računarskog modela: Solver model 65
Sl. 31. Detaljni izveštaj sistema Solver..... 66
Sl. 32. Moguća osnovna dopustiva rešenja za problem proizvodnje kada (1-5)..... 68
Sl. 33. Ilustracija veze između osnovnih dopustivih rešenja i ekstremnih tački 68
Sl. 34. Prikaz Simpleks algoritma..... 70
Sl. 35. Osnovni elementi reda čekanja..... 72
Sl. 36. Prostor stanja (a) i vrh stabla pretraživanja (b) optimalnog puta..... 75
Sl. 37. Monte Karlo metod računanja integrala empirijske krive 77
Sl. 38. Struktura sistema za podršku odlučivanju 90
Sl. 39. Struktura podsistema za upravljanje podacima 91
Sl. 40. Struktura podsistema za upravljanje modelima..... 92
Sl. 41. Struktura podsistema korisničkog interfejsa..... 94
Sl. 42. Primer komunikacije u prirodnom jeziku s inteligentnim agentom..... 95
Sl. 43. Struktura prostornog DSS..... 96
Sl. 44. Funkcionisanje GIS 97
Sl. 45 Osnovne komponente sistema poslovne inteligencije 101
Sl. 46. Struktura i komponente skladišta podataka 103
Sl. 47. Različite arhitekture skladištenja..... 104

Sl. 48. Proces ekstrakcije, transformacije i punjenja (ETL proces) 105
Sl. 49. SSIS - SQL Server Integration Services 106
Sl. 50. Pprimer strukture zvezde 107
Sl. 51. Proces poslovne inteligencije 109
Sl. 52. Faze standardnog procesa istraživanja podataka (CRISP-DM) 109
Sl. 53. Podatak, informacija i znanje 111
Sl. 54. Primer hijerarhijskog modela višekriterijumskog odlučivanja 118
Sl. 55. Primer diskretne tabelarno zadane funkcije 119
Sl. 56. Pregled rezultata Web glasanja 124
Sl. 58. Klasifikacija agenata prema nameni 136
Sl. 59. Svojstva agenata 137
Sl. 60. Osnovni ciklus mehanizma zaključivanja 145
Sl. 61. Primer male baze znanja za pomoć u kupovini prenosnih računara 145
Sl. 62. Primer lanca zaključivanja 146
Sl. 63. Primeri zaključivanja unapred i unazad 148
Sl. 64. Primer funkcionisanja algoritma RETE 149
Sl. 65. Primer induktivnog učenja razumljivog znanja 157
Sl. 66. Primer IRIS - induktivno učenje pravila i stabala odlučivanja 161
Sl. 67. Model induktivnog učenja na osnovu primera 162
Sl. 68. Struktura nervne ćelije 166
Sl. 69. Model veštačkog neurona 167
Sl. 70. Mreža veštačkih neurona 168
Sl. 71. Algoritam propagacije greške 169
Sl. 72. Tipične arhitekture veštačkih neuronskih mreža 169
Sl. 73. Princip rada metoda nosećih vektora 174
Sl. 74. Primena metoda zaključivanja na osnovu slučajeva 175
Sl. 75. Klasifikacije novih primera metoda zaključivanja na osnovu slučajeva 176
Sl. 76. Postupak zaključivanja na osnovu slučajeva 177
Sl. 77. Paleta alatki Microsoft Excel Data Mining dodatka 181
Sl. 78. Popularnost alata za istraživanje podataka prema KDNuggets.com 184
Sl. 79. Popularnost alata za istraživanje podataka prema Rexer Analytics 184
Sl. 80. Princip induktivnog učenja i upotreba profila korisnika 193
Sl. 81. Parametri rada inteligentnog asistenta za rukovanje neželjenom poštom 202
Sl. 82. Struktura kataloga sajta prodavnice Amazon 204
Sl. 83. Princip funkcionisanja inteligentnog pomoćnika u kupovini 205
Sl. 84. Softverski paketi poslovne inteligencije i analitike 209
Sl. 85. Arhitektura softvera za podršku menadžmentu kompanije Microstrategy 210
Sl. 86. Načini integracije sistema za podršku odlučivanju i ekspertnih sistema 211
Sl. 87. Uloga članova razvojnog tima ekspertnog sistema 215
Sl. 88. Rezultat učenja stabala odlučivanja (vote.arff) 221
Sl. 89. Rezultat učenja pravila odlučivanja (vote.arff) 222
Sl. 90. Stablo odlučivanja u sistemu SQL Server BI 226
Sl. 91. Stablo regresije u dodatku Excel Data Mining 227
Sl. 92. Grafički orijentisani interfejs R Studio 228
Sl. 93. Microsoft Visual Studio dodatak Python Tools for Visual Studio 229
Sl. 94. Grafički orijentisani interfejs sistema Matlab 229
Sl. 95. Životni ciklus softverskog sistema 230

Sl. 96. Metodologija vodopada 231
Sl. 97. Spiralna metodologija razvoja softvera 232
Sl. 98. RUP metodologija procesa razvoja objektno orijentisanog softvera 233
Sl. 99. Proces prototipskog razvoja softvera 235
Sl. 100. Varijante organizacije izgradnje i funkcionisanja sistema 236
Sl. 101. Proces evaluacije projekata DSS 238
Sl. 102. Analizirani model sistema za preporučivanje 239
Sl. 103. Višekriterijumski model evaluacije sistema za preporučivanje 240

Predgovor

Udžbenik se koristi kao osnovna literatura za predmet *Sistemi za podršku odlučivanju (DSS)* na Fakultetu za informatiku računarstvo i Tehničkom fakultetu Univerziteta Singidunum u Beogradu.

Osnovni cilj materijala je izlaganje metoda i odgovarajućih savremenih softverskih alata za podršku odlučivanju, koji obuhvataju sisteme za podršku donošnju odluka zasnovane na kvantitativnim modelima i inteligentne sisteme, kao što su npr. ekspertni sistemi, sistemi za preporučivanje i inteligentni asistenti. Služe za rešavanje veoma različitih praktičnih problema, koji se načelno mogu podeliti na strukturirane i nestrukturirane.

Računarska podrška u rešavanju *strukturiranih* problema izlaže se kroz izabrane metode nauke o menadžmentu (ranije operacionih istraživanja), kao što su metodi linearnog programiranja, višekriterijumskog odlučivanja, mreža repova čekanja i računarske simulacije.

Računarska podrška odlučivanju u rešavanju *nestrukturiranih* problema razmatra se kroz izgradnju *inteligentnih sistema*, kao što su ekspertni sistemi i sistemi za preporučivanje u elektronskoj trgovini, odnosno kroz primenu savremenih metoda *poslovne analitike*, pre svega istraživanja podataka metodima mašinskog učenja.

Zajedničke osnove razmatranih metoda rešavanja nestrukturiranih problema predstavljaju metodi mašinskog učenja, koji su ukratko izloženi u posebnom poglavlju. Neophodni su za izgradnji inteligentnih sistema i predstavljaju osnovne metode istraživanja podataka, odnosno poslovne analitike u podrški odlučivanju.
Većina tema izlaže se uz kratki teorijski uvod i ilustrativne primere, nakon čega se daju dodatna uputstva za praktičnu realizaciju primera jednostavnih sistema i samostalne eksperimente, zbog čega se prilikom izbora softverskih alata prednost daje savremenim besplatnim alatima, odnosno alatima otvorenog koda.

Beograd

19.09.2016. godine

Predgovor prvom izdanju

Knjiga je nastala kao osnovni tekst za predmet *Sistemi za podršku odlučivanju (DSS)* na Fakultetu za informatiku računarstvo i Fakultetu za menadžment Univerziteta Singidunum.

Osnovni cilj predmeta je izlaganje materije koja omogućava razumevanje i praktično korišćenje metoda, tehnika i savremenih programskih alata u podrški donošenju različitih odluka, kao što su grupne odluke, odluke u uslovima neizvesnosti i odluke koje se donose u nestrukturiranim, odnosno nedovoljno definisanim situacijama.

Teorijski aspekti nauke o menadžmentu, statistike, matematičke logike i računarskih nauka daju se samo u neophodnom obimu, koji omogućava razumevanje postojećih i samostalno kreiranje pojedinih klasa sistema za podršku odlučivanju korišćenjem savremenih, široko dostupnih programskih alata.

U knjizi se daje sažet pregled softverskih tehnologija, sistema i alata koji se koriste u realizaciji modernih sistema za podršku odlučivanju: baza i skladišta podataka, sistema i alata za razvoj prostornih sistema za podršku odlučivanju, sistema poslovne inteligencije i samostalnih sistema i alata za istraživanje podataka, automatizovano (mašinsko) učenje i razvoj sistema zasnovanih na znanju, odnosno inteligentnih sistema.

Pošto savremene organizacije često raspolažu velikim obimom podataka, koji ne mogu klasičnim metodima obraditi i upotrebiti u procesu donošenja odluka, u knjizi je posvećena pažnja osnovama istraživanja podataka (*Data Mining*) i mašinskog učenja (*Machine Learning*), kao i naprednim metodima analize podataka u okviru sistema poslovne inteligencije (*Business Intelligence*).

Upotreba klasično prikupljenog i mašinski generisanog znanja u sistemima zasnovanim na znanju obrađena je kroz primere realizacije jednostavnih ekspertnih sistema zasnovanih na pravilima.

Web okruženje predstavlja platformu savremenih poslovnih komunikacija, informacionih sistema i sistema za podršku odlučivanju. U knjizi su kratko prikazani inteligentni sistemi i podrška odlučivanju u elektronskoj trgovini, kao jednoj od najvažnijih poslovnih primena globalne svetske mreže i pratećih Web tehnologija.

Način izlaganja većine tema zasniva se na kratkom teorijskom uvodu i primerima, nakon čega se daju detaljnija uputstva za praktičnu realizaciju i samostalne eksperimente, koji treba da pojasne i demistifikuju materiju koja danas više nije akademska, već se rutinski primenjuje u realizaciji savremenih softverskih sistema, posebno u Web okruženju.

Beograd

26.08.2013. godine

1 Uvod u sisteme za podršku odlučivanju

1. Osnovni pojmovi
2. Proces donošenja poslovnih odluka
3. Modeliranje procesa donošenja odluka
4. Automatizacija podrške odlučivanju
5. Nastanak i razvoj sistema za podršku odlučivanju

U ovom poglavlju se uvode važniji pojmovi neophodni za razumevanje procesa odlučivanja i sistema za podršku procesa donošenja odluka. Objašnjava se proces donošenja poslovnih odluka, potreba njegovog modeliranja i osnovne prednosti automatizacije procesa poslovnog odlučivanja.

1.1 Osnovni pojmovi

Sistemi za podršku odlučivanju obuhvataju sisteme za podršku donošnju odluka zasnovane na kvantitativnim modelima i inteligentne sisteme, kao što su npr. ekspertni sistemi, sistemi za preporučivanje i inteligentni asistenti.

Donošenje odluka se vezuje za aktivni odnos živog sveta prema okruženju. Odluke o izboru jedne od mogućnosti ili puteva rešavanja nekog problema donose i najjednostavniji oblici života, kao što su virusi i jednoćelijske biljke i životinje, što im omogućava da prežive kao jedinke i kao vrste u promenljivim uslovima životne sredine. Od posebnog interesa za ljudsko društvo su odluke koje donosi čovek, koji pri tome koristi racionalno mišljenje i logičko zaključivanje. Pošto nisu uvek poznate sve činjenice i posledice mnogih odluka, za njihovo donošenje se čovek oslanja na intuiciju i iskustvo, tako da odluke ne zavise samo od okolnosti, već i od toga ko donosi odluke [1], [2].

Posledice pojedinih odluka mogu biti dalekosežne. U istoriji ljudskog društva zabeležene su mnoge značajne odluke poznatih ličnosti, kao i okolnosti u kojima su donesene [61]:

- *Aleksandar Veliki* (356-323. pne) - antičkom grčkom vladaru i vojskovođi Aleksandru pripisuje se sposobnost za donošenje odluka u teškim situacijama. Prema legendi, kada je od njega zatraženo da razmrsi čvor koji do tada niko nije uspeo, izvukao je mač i jednostavno presekao čvor. Danas je izraz "Gordijev čvor" metafora za veoma težak, praktično nerešiv problem.
- *Qin Shi Huangdi*, prvi car ujedinjene Kine (259-210. pne) - doneo je odluku da započne izgradnju velikog zida, koji je njegovu državu trebalo da brani od napada nomadskih plemena. Budućnost je potvrdila ispravnost odluke, pošto je zid bio u funkciji i dorađivan je sve do 17. veka.

- *Gaj Julije Cezar* (100-44. pne) - na povratku iz Galije, pre prelaska reke Rubikon, odlučio je da se ogluši o naredbu senata da raspusti vojsku i sam se vrati u Rim, gde bi bio bespomoćan pred uticajnim Pompejem. Navodno je pre prelaska reke Rubikon izgovorio čuvenu rečenicu "Kocka je bačena" (latinski: Alea iacta est!).

Primer: Strategijska igra Hamurabi



Kao ilustracija mogućih posledica intuitivnog donošenja odluka može se uzeti klasična strategijska igra Hamurabi. Igrač je u ulozi Hamurabija[1], vladara Vavilona, koji donosi poslovne odluke o feudalnoj proizvodnji: kupoprodaji zemlje i proizvoda, setvi i raspodeli letine. Odluka se sastoji u definisanju tri veličine: (1) površini zemljišta koje će se kupiti (zamenom za seme), (2) površini zemljišta koja će se zasejati i (3) količini semena koja se ostavlja za ishranu stanovništva. Za uspešno vladanje, neophodno je održavati balans potreba stanovništva i raspoloživih količina hrane u deset ciklusa proizvodnje (godina). Otežavajuće okolnosti su iznenadna pojava bolesti, ratovi i štetočine (glodari), koji uništavaju letinu.

Igra je nastala 1968. godine, ali je postala popularna verzija za mikroračunare u jeziku BASIC iz 1973. godine. Na na slici je prikaz jednog ciklusa iste igre, koja je prevedna u jezik C#:

```
Command Prompt
POKUŠAJTE USPEŠNO DA VLADATE ANTIČKIM SUMEROM
S DESETOGODIŠNJIM MANDATOM.



HAMURABI: MOLIM ZA DOZVOLU DA VAS OBAVESTIM,
U GODINI 1, 0 LJUDI JE UMRLO, 5 DOŠLO U GRAD.
BROJ STANOVNIKA JE 100
GRAD SADA POSEDUJE 1000 JUTARA.
POŽNJEVENO JE 3 BUŠELA SEMENA PO JUTRU.
GLODARI SU POJELI 200 BUŠELA SEMENA.
SADA IMATE 2800 BUŠELA SEMENA U ZALIHAMA.

ZEMLJIŠTE SE PRODAJE PO 25 BUŠELA SEMENA PO JUTRU.
KOLIKO JUTARA ŽELITE DA KUPITE? 100

KOLIKO BUŠELA SEMENA ŽELITE ZA ISHRANU SVOG NARODA? 100

KOLIKO JUTARA ŽELITE DA ZASEJETE? 100

OD GLADI VAM JE UMRLO 95 LJUDI U JEDNOJ GODINI!!!
ZBOG IZUZETNO LOŠEG UPRAVLJANJA NE SAMO DA STE
SVRGNUTI S VLASTI I ZBAČENI S TRONA NEGO
STE TAKOĐE PROGLAŠENI NACIONALNOM SRAMOTOM!!

SREĆNO ZA SADA.
```

Postoje brojne Web verzije igre, npr. www.hamurabigame.com, kao i verzije za mobilne platforme, obogaćene grafikom i zvučnim efektima.

[1] Hamurabi (1810-1750 pne) je osnivač prvog Vavilonskog carstva. Poznati Hamurabijev zakonik, jedna od najstarijih poznatih zbirki zakona, bio je uklesan na bazaltni stub i postavljen na javno mesto. Imao je 282 člana i predviđao isti tretman za sve slojeve stanovništva.

Jedna od poznatih metafora, koja razmatra kreativnost donosioca odluke u rešavanju teških problema, je tzv. "Kolumbovo jaje" (*egg of Columbus*). Kristifor Kolumbo (1451-1506) je, nakon povratka sa svog putovanja, naišao na brojne kritike i omalovažavanja svog poduhvata. Jednom takvom prilikom javno je izazvao svoje kritičare da reše težak problem: da postave jaje tako da stoji uspravno. Kad su odustali, udarcem je izravnao jedan njegov kraj i postavio ga uspravno. Time je javno demonstrirao da teški problemi više ne izgledaju tako kad se jednom pronađe rešenje, nakon čega svako može da zna kako se problem rešava.

Savremeni način rešavanja Kolumbovog problema je tzv. "Teslino jaje". Na svetskoj izložbi u Čikagu 1893. godine, Nikola Tesla (1856-1943) je privukao veliku pažnju kad je uspravio metalno jaje pomoću svog rotacionog magnetnog polja.

Odlučivanje u organizacionim sistemima

Ponekad se pojam donošenja odluka izjednačava s rešavanjem problema (*problem solving*), kada se ne podrazumeva i primena same odluke. Menadžment se može posmatrati kao menadžersko donošenje odluka, jer se u svakoj menadžerskoj funkciji donose odluke.

Donošenje odluka (*decision making*) je proces izbora između dva ili više alternativnih puteva delovanja radi postizanja cilja ili ciljeva [1].

Menadžment je proces u kome se, korišćenjem resursa, postižu organizacioni ciljevi [1]. Resursi su ulazi procesa, ciljevi su izlazi, a njihov odnos je stepen uspešnosti ili produktivnost organizacije (*productivity*). Produktivnost ili uspešnost sistema zavisi od menadžerskih funkcija, kao što su planiranje, organizovanje, upravljanje i kontrola, koji podrazumevaju neprekidno donošenje odluka prilikom izbora najboljeg od mogućih rešenja.

Neke od najsloženijih i najvažnijih dokumentovanih odluka u istoriji ljudskog društva odnose se na vojni menadžment ili ratovodstvo. Do danas su sačuvana značajna dela različitih kultura, koja objedinjavaju tadašnje znanje i iskustvo o organizaciji i vođenju složenih ratnih operacija, npr.

- *Sedam vojnih klasika* (500 g. pne), Kina - klasični tekstovi iz vojnog menadžmenta, koji su nastali u veoma dugom periodu istorije Kine, a objedinjeni su u XI veku. Sadrže različite teme, od jednostavnih taktičkih principa do složenih metoda organizovanja vojske i države. Najpoznatije delo iz ove zbirke je *Umetnost ratovanja* Sun Cu-a.
- *Peloponeski rat*, Tukidid, Grčka (400 g. pne) - grčki istoričar analitički je opisao dugotrajni i iscrpljujući niz ratova za prevlast između dve grupacije grčkih gradova (431-404. g. pne) pod vođstvom Sparte na jednoj i Atine na drugoj strani.

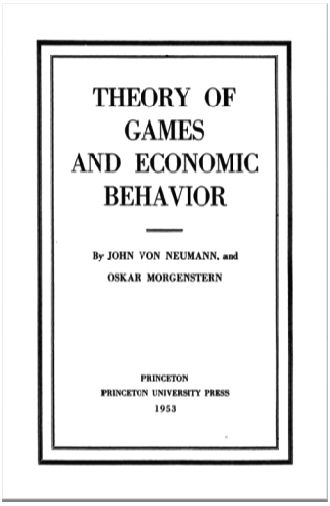
- *Vojni komentari* Gaja Julija Cezara, Rim (50 g. pne), koji se sastoje od Cezarovih zapisa o galskim ratovima (lat. *Commentarii de bello Gallico*, u 7 knjiga), odnosno opisima pohoda u Galiji, Germaniji i Britaniji i zapisa o građanskom ratu (lat. *Commentarii de bello civīli*, u 3 knjige).
- *Strategikon*, Vizantija, VI vek, delo koje se pripisuje caru Mauricijusu, predstavlja priručnik za strategе vizantijske vojske, koji u 12 knjiga obrađuje različite aspekte tadašnjeg vojnog organizovanja, te principe i metode ratovanja.

Naučne osnove odlučivanja

Naučne osnove donošenja odluka predstavljaju naučne discipline teorija odlučivanja (*decison theory*) i nauka o menadžmentu (*management science*). Savremena teorija odlučivanja nastala je u dvadesetom veku, proučavanjem različitih aspekata pojedinačnog i kolektivnog odlučivanja u okviru više drugih naučnih disciplina: ekonomije, statistike, psihologije, političkih i društvenih nauka i filozofije. Nauka o menadžmentu ili *operaciona istraživanja* je disciplina koja se bavi primenom kvantitativnih tehnika na donošenje odluka [1], [61].

Teorija odlučivanja

Teorija odlučivanja proučava različite aspekte pojedinačnog i kolektivnog odlučivanja: racionalnost odluke, odlučivanje u uslovima neizvesnosti, donošenje optimalne odluke, grupno odlučivanje. Statistička teorija odlučivanja koristi teoriju verovatnoće, tako da je u primeni ograničena relativno strogim pretpostavkama koje moraju biti zadovoljene da bi teorijski rezultati bili primenljivi u praksi. Postoje i drugi formalni pristupi odlučivanju, koji nastoje da prevaziđu ograničenja koja postavlja teorije verovatnoće, npr. fazi logika, teorija Dempstera-Šafera i druge generalizacije statističke teorije odlučivanja [3], [4].


THEORY OF
GAMES
AND ECONOMIC
BEHAVIOR

By JOHN VON NEUMANN, and
OSKAR MORGENSTERN

PRINCETON
PRINCETON UNIVERSITY PRESS
1953

Teorija igara

Teorija igara (*game theory*) je područje vrlo blisko teoriji odlučivanja, koje se bavi matematičkim modeliranjem procesa odlučivanja u uslovima konkurencije donosilaca odluka, koji se nazivaju igračima.

Teorija igara postala je posebno područje nakon rada Džona fon Nojmana[2] *On the Theory of Games of Strategy* (1928) [5]. Zajedno sa Oskarom Morgenšternom napisao je knjigu *Theory of Games and Economic Behavior* (1944), u kojoj se razmatraju kooperativne igre više igrača. U drugom izdanju knjige izložena je aksiomatska teorija korisnosti (*expected utility*), koja omogućava matematičko tretiranje odlučivanja u uslovima neizvesnosti.

Nauka o menadžmentu

Nauka o menadžmentu, koja je još poznata pod nazivom operaciona istraživanja (*operations research*), bavi se primenom kvantitativnih metoda na donošenje odluka [6], [7]. Osnovni zadatak u razvoju i primeni ovih metoda je optimizacija organizacionih sistema. Optimizacija je postupak pronalaženja optimalnog rešenja, koje obezbeđuje maksimumalnu ili minimalnu vrednost određene veličine koja se naziva *cilj*, a koja zavisi od konačnog broja *ulaznih promenljivih*, međusobno nezavisnih ili povezanih preko jednog ili više *ograničenja*. Nauka o menadžmentu koristi brojne matematičke metode i tehnike optimizacije, kao što su linearno i nelinearno programiranje, teorija igara, teorija repova ili redova čekanja (*queueing theory*), teorija grafova, višekriterijumska optimizacija, simulacije, itd. [6], [7], [8], [9].

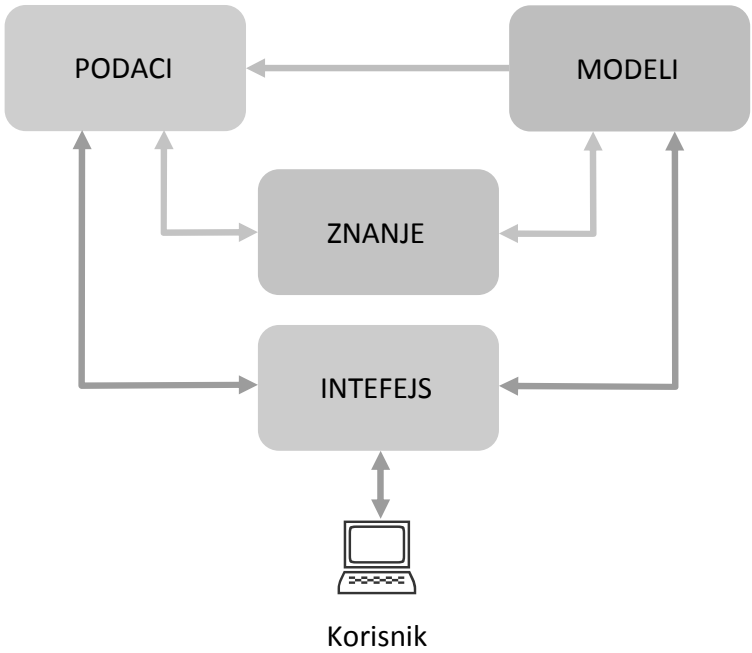
Sistemi za podršku odlučivanju (Decision Support Systems, DSS)

Ne postoji opšteprihvaćena precizna definicija pojma sistema za podršku odlučivanju [1], [2]. Termin se koristi u više sasvim različitih značenja, npr.

1. Interaktivni (računarski zasnovan) sistem ili podsistem za pomoć donosiocima odluka u
   - korišćenju komunikacionih tehnologija, podataka, dokumenata, znanja i/ili modela
   - identifikaciji, rešavanju problema i izvršavanju zadataka u procesu odlučivanja i
   - donošenju odluka;
2. Opšti izraz za računarsku aplikaciju, koja poboljšava ličnu ili grupnu sposobnost donošenja odluka;
3. Naziv akademskog područja, koje istražuje i proučava sisteme za podršku odlučivanju u kontekstu njihove upotrebe.

[2] John von Neumann (1903-1957), američki matematičar i naučnik mađarskog porekla, jedan od najvećih naučnika dvadesetog veka, koji je dao doprinos kvantnoj fizici, funkcionalnoj analizi, teoriji skupova, topologiji, informatici, ekonomiji, numeričkoj analizi, hidrodinamici, statici i drugim matematičkim poljima. Stvorio je i teoriju igara, a u toku rada na projektu atomske bombe, rešavajući probleme konstrukcije i rada računskih mašina, postavio je temelj savremene računarske arhitekture i simulacija.

U ovom materijalu će se pod sistemom za podršku odlučivanju podrazumevati interaktivni računarski sistem ili podsistem koji pomaže u donošenju odluka. Osnovne komponete ovakvog sistema su podaci, modeli, znanje i korisnički interfejs, Sl. 1.



Sl. 1. Opšta struktura sistema za podršku odlučivanju

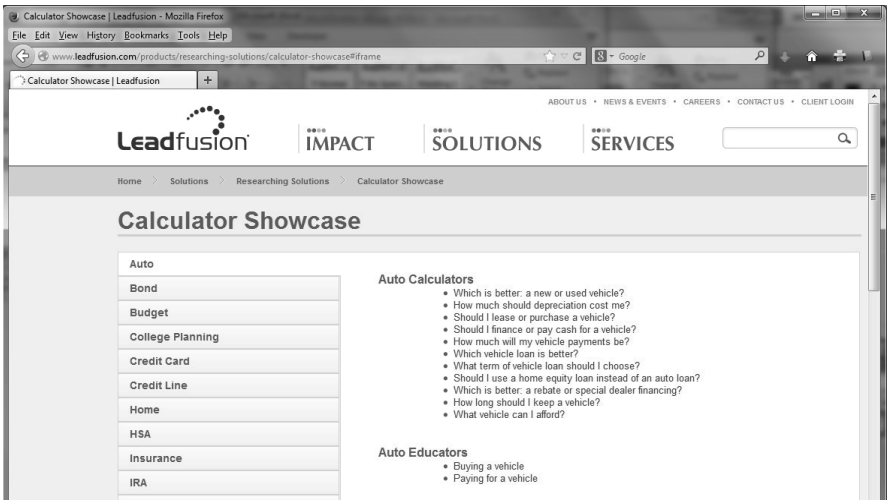
Važna razlika ovih sistema u odnosu na druge informacione sisteme je da se sistemi za podršku odlučivanju zasnivaju na *modelima*. Znanje je opciona komponenta koju imaju sistemi za podršku odlučivanju zasnovani na znanju [1], [10].

Treba naglasiti da uloga sistema za podršku odlučivanju nije da zamene čoveka u procesu odlučivanja, već da olakšaju donošenje odluka, tako što pripremaju elemente za bolje razumevanje samog problema, mogućih alternativa i njihovih posledica.

Ilustracija: Finansijski Web kalkulatori za podršku odlučivanju

Relativno jednostavan Web sistem, koji podržava donošenje nekoliko vrsta odluka u oblasti finansija može se naći na sajtu kompanije Leadfusion (www.leadfusion.com/products/researching-solutions/calculator-showcase).

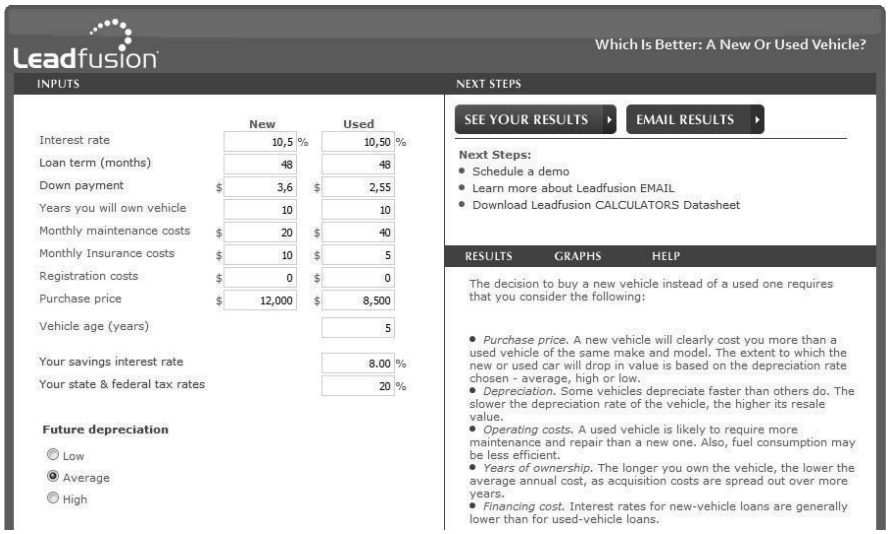
Sistem se sastoji od većeg broja aplikacija tipa kalkulatora, Sl. 2.



Sl. 2. Primeri finansijskih kalkulatora

Prikazaće se samo jedna od korisnih kalkulacija, koja treba da olakša donošenje odluke o kupovini automobila na kredit. Kalkulator računa efekte kupovine novog (*New*) ili polovnog (*Used*) automobila, odnosno odgovara na pitanje: *da li je bolje kupiti novi ili polovni automobil?*

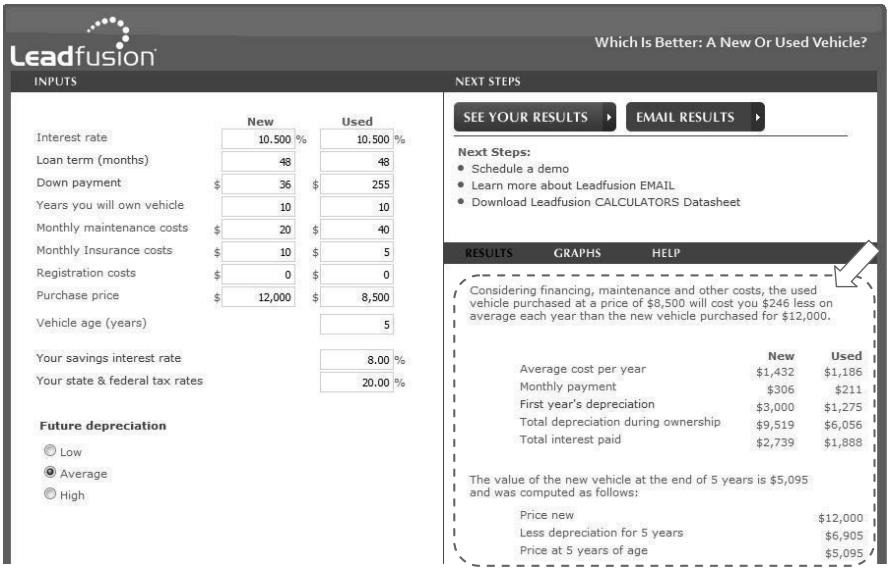
Primer upotrebe sistema na konkretnom problemu je na Sl. 3, gde su uneseni podaci neophodni za analizu: o uslovima kreditiranja (1), automobilu (2) i cenama (3).



Sl. 3. Podaci za odluku o načinu kupovine automobila (novi ili polovni)

Na Sl. 3. uneseni su okvirni podaci o ceni i uslovima kreditiranja za jedan model porodičnog automobila, s procenjenim vrednostima cene održavanja i prosečnom stopom amortizacije.

Na osnovu konkretnih podataka unesenih kao na Sl. 3, sistem predlaže rešenje i obrazlaže svoj predlog odluke kao na Sl 4.



Sl. 4. Predlog odluke o načinu kupovine automobila (polovni)

U konkretnom slučaju, na osnovu nabavne cene, iznosa kredita i kamate, planiranog perioda korišćenja i mesečnih troškova održavanja i osiguranja, sistem je predložio kupovinu polovnog automobila starog 5 godina, s obrazloženjem da su ukupni godišnji troškovi vlasništva manji za 246 EUR.

Sistemi za podršku odlučivanju zasnovani na elektronskim tabelama

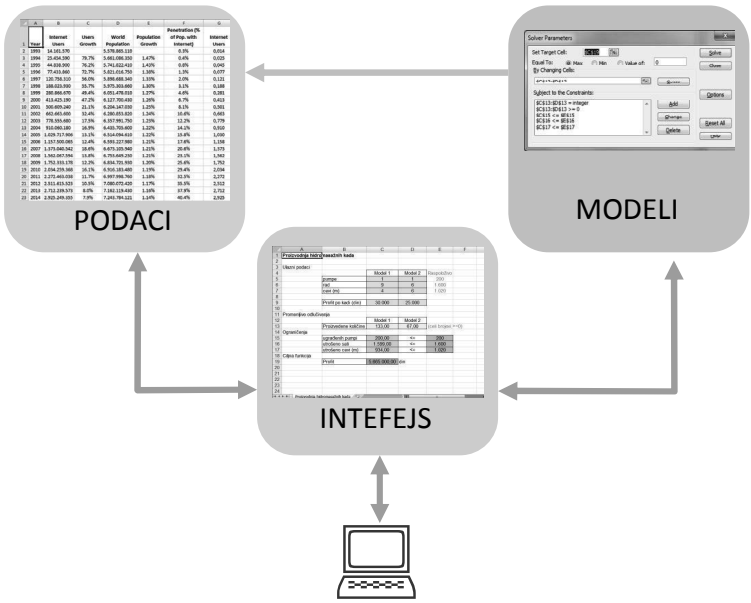
Sistemi za podršku odlučivanju načelno se sastoje od modela, podataka i korisničkog interfejsa, čija programska realizacija zahteva velike razvojne napore ako se razvijaju od početka.

Jedan od pristupa, koji može da pojednostavi i olakša razvoj manje zahtevnih sistema za podršku odlučivanju, jeste korišćenje mogućnosti softverskih alata zasnovanih na elektronskim tabelama (*spreadsheet*) [1], [8], [11], kao što je npr. sistem Microsoft *Excel*. Standardne komponente sistema za podršku odlučivanju, podaci, modeli i korisnički interfejs, lakše i brže se mogu realizovati u ovoj tehnologiji, Sl. 5:

- *Model* - modeli sistema se mogu realizovati korišćenjem formula i drugih računskih i programskih mogućnosti elektronskih tabela, a moguće ih je

proširiti pomoću programskih dodataka, kao što je dodatak (Add-In) *Solver*, koji je namenjen rešavanju modela linearnog programiranja;

- *Podaci* - baza podataka se može realizovati u obliku elektronskih tabela, a moguće je pristupiti i eksternim bazama podataka preko SQL upita;
- *Korisnički interfejs* - može se realizovati korišćenjem tabela kao formi za unos podataka i brojnih ugrađenih grafičkih mogućnosti za prikaz izlaznih rezultata.



Sl. 5. Komponente sistema za podršku zasnovanog na elektronskim tabelama

Dodatne funkcije i specifični zahtevi se u sistemima s elektronskim tabelama mogu realizovati pomoću ugrađenog programskog jezika, kao što je *Visual Basic for Applications* (VBA) u sistemu Microsoft *Excel*.

Ilustracija: Sistem realizovan pomoću Microsoft Excel tabela i dodataka

Na Sl. 7 (a) prikazan je sistem za podršku odlučivanju (DSS) realizovan kao elektronska tabela u sistemu Microsoft *Excel* uz korišćenje njegovog dodatka za rešavanje linearnih (i nelinearnih) sistema jednačina *Solver*.

Podaci su uneseni u polja *Excel* tabele, koja predstavlja osnovnu bazu podataka.

Model problema se sastoji od (1) formula unesenih u polja tabele i (2) modela optimizacije koji koristi dodatak *Solver*.

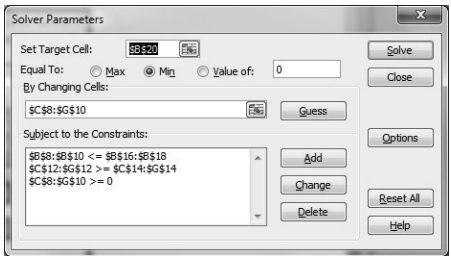
Interfejs sistema su (1) forma elektronske tabele, gde se bojama i grafikom ističu pojedini elementi i (2) interfejs dodatka *Solver*.

Na Sl. 7 (b) prikazan je dijalog za unos elemenata *Solver* modela, koji se kreira na osnovu unesenih podataka i formula u *Excel* tabeli. Dijalog se koristi i za pokretanje izvršavanja modela.

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Example 2: Transportation Problem.												
2	Minimize the costs of shipping goods from production plants to warehouses near metropolitan demand												
3	centers, while not exceeding the supply available from each plant and meeting the demand from each												
4	metropolitan area.												
6			*Number to ship from plant x to warehouse y (at intersection):*										
7	*Plants:*	*Total*	*San Fran*	*Denver*	*Chicago*	*Dallas*	*New York*						
8	S. Carolina	5	1	1	1	1	1						
9	Tennessee	5	1	1	1	1	1						
10	Arizona	5	1	1	1	1	1						
11			—	—	—	—	—						
12	Totals:		3	3	3	3	3						
13													
14	*Demands by Whse →*		180	80	200	160	220						
15	*Plants:*	*Supply*	*Shipping costs from plant x to warehouse y (at intersection):*										
16	S. Carolina	310	10	8	6	5	4						
17	Tennessee	260	6	5	4	3	6						
18	Arizona	280	3	4	5	5	9						
19													
20	*Shipping:*	$83	$19	$17	$15	$13	$19						

Color Coding
- Target cell
- Changing cells
- Constraints

(a) elektronska tabela (transportni problem)



(b) interfejs za kreiranje i ažuriranje modela

Sl. 6. Primer sistema za podršku odlučivanju (DSS) zasnovanog na Excel tabelama

1.2 Proces donošenja poslovnih odluka

Proces odlučivanja u smislu ovog materijala je *proces izbora između dva ili više alternativnih puteva delovanja radi postizanja cilja ili ciljeva* (radna definicija, [1]). Kao ilustracija veoma složenog problema odlučivanja može se uzeti odluka centralne nacionalne banke o promeni referentne kamatne stope, koja predstavlja osnovu za određivanje kamatnih stopa celog bankarskog sektora [1].

Ilustracija: Određivanje referentne kamatne stope centralne banke

Donošenje odluke o povećanju ili smanjenju referentne kamatne stope centralne banke je veoma složen proces, pošto odluka zavisi od velikog broja faktora i ima ozbiljne posledice na celokupnu privrevredu jedne zemlje. Odluku načelno donose pojedinci, guverneri centralne banke, i za

nju odgovaraju. Kamatna stopa se može povećati ili smanjiti za različite iznose, tako da treba razmotriti veoma veliki broj mogućih alternativa, koje se odnose na hiljade mogućih vrednosti kamatne stope.

Za pripremu odluke su potrebni brojni podaci, koje obezbeđuju banke, privreda i statistika. Takođe je potrebno znanje za njihovo razumevanje i pravilnu upotrebu. Pri tome je potrebno prikupiti veliki broj podataka i izvršiti različite analize, što se ne može sprovesti bez adekvatne računarske podrške. U analizama se koriste različiti modeli, što omogućava analizu samo bitnih aspekata nekog problema. Za povećanje pouzdanosti analiza koristi se više izvora informacija, tako da se kompletna analiza dobija objedinjavanjem rada brojnih specijalizovanih timova.

Opisani postupak se može rezimirati u nekoliko zapažanja:

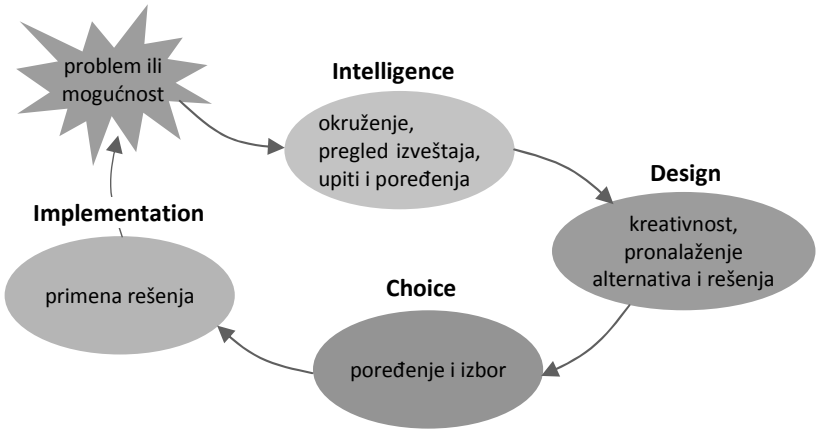
- odluke načelno donose pojedinci (guverner) i *odgovaraju* za njih;
- postoji veoma *veliki broj* alternativa koje treba razmotriti (hiljade);
- potrebni su brojni *podaci* (banke, privreda, statistika) i detaljna analiza i znanje za njihovo *razumevanje*. Složene odluke zahtevaju *računarsku podršku* za pristup podacima i analize;
- razmatranje problema zahteva *modeliranje* i proizvodi *modele*. Za donošenje odluke se tipično koristi više izvora informacija i rezltati rada brojnih stručnih timova;
- prethodno iskustvo i podaci mogu biti *nedovoljni* za predviđanje budućih rezultata;
- u procesu donošenja odluke postoji više *ciljeva*, koji mogu biti *suprotstavljeni*;
- mnoge odluke nose *rizik*, koji različito utiče na pojedince koji odlučuju;
- *grupno odlučivanje* može dati bolje odluke, npr. promene kamatne stope predlaže odbor guvernera.

Menadžer je donosilac odluka (*decision maker*). Donošenje odluke je složen *proces*, koji se obično razmatra kroz više faza (različiti autori navode 3-6). Prema [1], proces donošenja odluka se može opisati kao:

1. Definisanje problema (problemske situacije)
2. Konstrukcija *modela* koji opisuje realni problem
3. Identifikacija i ocena mogućih rešenja modeliranog problema
4. Poređenje, *izbor* i preporuka mogućih rešenja problema

Jednu podelu procesa odlučivanja na faze predložio je 1960. godine poznati američki nobelovac Herbert A. Sajmon[3] [1]. Sajmon je proces donošenja odluke prikazao prvo u tri faze: prikupljanje informacija (*intelligence*), kreiranje rešenja (*design*) i izbor rešenja (*choice*), kojima je zatim dodao i četvrtu fazu, primenu rešenja (*implementation*).

Faze procesa odlučivanja prikazane su na Sl. 7 [1]:
1. Prikupljanje informacija (*intelligence*)
2. Kreiranje rešenja (*design*) je faza u kojoj se kreira *model* problema
3. Izbor rešenja (*choice*)
4. Primena rešenja (*implementation*)



Sl. 7. Faze procesa odlučivanja

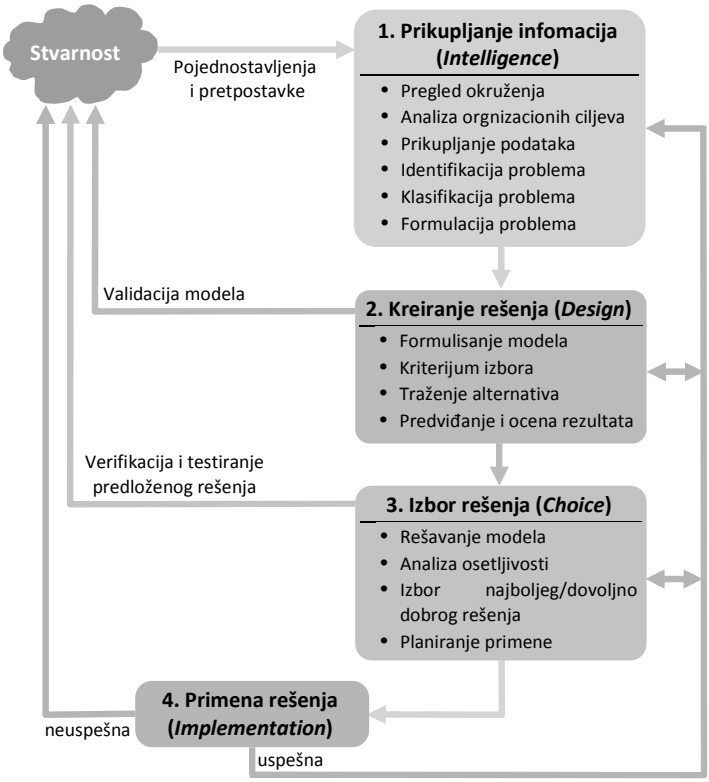
Prve tri faze često se nazivaju rešavanje problema (*problem solving*), dok je proces primene rešenja uveden naknadno, kao veoma važan apekt procesa odlučivanja.

Proces odlučivanja se načelno ne odvija sekvencijalno [1], kao što je prikazano na Sl. 8. Svaka faza predviđa mogućnost povratka na neku od prethodnih faza. Pri tome je *modeliranje* suštinski element procesa odlučivanja.

Sadržaj pojedinih faza procesa odlučivanja se može ukratko sumirati [1]:

1. Faza prikupljanja informacija (*intelligence*) obuhvata pregled okruženja, analizu organizacionih ciljeva, prikupljanje podataka, identifikaciju problema, klasifikaciju problema u poznate kategorije (npr. programirani ili neprogramirani), dekompoziciju u manje celine i obezbeđenje nadležnosti, odnosno prava i odgovornosti za rešavanje problema.

[3] Herbert Alexander Simon (1916-2001), američki naučnik, dobitnik Nobelove nagrade za ekonomiju 1978. godine, jedan od najuticajnijih naučnika 20. veka, koji se primarno bavio političkim naukama, ali i kognitivnom psihologijom, računarstvom, ekonomijom, menadžmentom, filozofijom nauke i sociologijom.



Sl. 8. Faze procesa donošenja odluke (Sajmon, 1960)

2. Faza kreiranja rešenja (*design*) podrazumeva razradu alternativnih puteva rešavanja, analizu potencijalnih rešenja, *kreiranje modela* (koji se može primeniti na dati problem), testiranje izvodljivosti (*feasibility*) i ocenu rezultata, kao i izbor postupka za izbor rešenja (*principle of choice*), koji podrazumeva utvrđivanje ciljeva, uključivanje u model procene rizika, ocene prihvatljivosti rešenja i drugih kriterijuma i ograničenja.

   Normativni modeli, npr. *optimizacija*, koja daje efekte svake alternative, koriste se za izbor najbolje alternative. Moguća je *suboptimizacija*, kada se odluke donose u različitim delovima organizacije bez uvažavanja celine.

   Deskriptivni modeli, kao što su simulacioni modeli i modeli redova čekanja (*queueing*, *wating lines*), služe za opis efekata primene alternativa u različitim konkretnim uslovima. Koriste se i drugi deskriptivni modeli, npr. kongnitivne mape i narativni modeli.

3. Faza izbora rešenja (*choice*) predstavlja akt izbora alternative na osnovu modela, koji sadrži opis prihvatljivosti izabranog pristupa rešavanju problema. Pre izbora rešenja često se ponavlja prethodna faza, radi generisanja novih ili modifikacije postojećih alternativa.

Izbor najboljeg rešenja prema modelu odlučivanja zahteva pronalaženje najboljeg puta rešavanja, koji podrazumeva primenu različitih analitičkih tehnika, algoritama i heuristika pretraživanja. Ocenjuje se svaka alternativa u odnosu na ostale, kao i osetljivost svakog rešenja na male promene ulaznih parametara (*sensitivity analysis*), efekti promene ulaznih parametara (*what-if analysis*), kao i vrednosti promenljivih za koje se određeni ciljevi postižu (*goal seeking*).

4. Faza primene rešenja (*implementation*) se odnosi na primenu izabranog rešenja, gde se pojavljuju tipični problemi npr. zbog neodređenih ograničenja, otpora promenama, nedovoljne obuke korisnika i izostanka podrške viših nivoa menadžmenta.

Različite softverske tehnologije predstavljaju osnov računarske podrške pojedinim fazama donošenja odluka [1], npr. veštačke neuronske mreže (ANN), menadžment informacioni sistemi (MIS), istraživanje podataka (Data Mining), interektivne analitičke obrade (OLAP), ekspertni sistemi (ES) i integisani kompanijski informacioni sistemi (ERP), sistemi za odnose s klijentima (CRM) i sistemi zasnovani na znanju (KMS).

1.3 Modeliranje procesa donošenja odluka

Model je pojednostavljena predstava ili apstrakcija stvarnosti, u kojoj se izdvajaju najvažnija svojstva neke pojave i smanjuje kompleksnost sistema koji se razmatra, npr. procesa odlučivanja.

Osnovne kategorije modela su, prema rastućem stepenu apstrakcije u odnosu na stvarnost [1], [9]:

- *Skalirani* modeli - umanjene kopije, kao što su makete i fotografije, npr. model Sunčevog sistema, globus i anatomski model čoveka.
- *Analogni* modeli - modeliraju samo *ponašanje* sistema i ne moraju da podsećaju na stvarni sistem. Različiti planovi, dijagrami i mape, kao što su UML dijagrami i *Word Cloud* modeli, spadaju u ovu kategoriju.
- *Konceptualni* ili *mentalni* modeli - kvalitativni opisi, koji pomažu da se istaknu važne veze u realnim sistemima i procesima. Mentalne modele različitih aspekata sveta u kome žive imaju svi ljudi. Predstavljaju prvi korak u razvoju složenijih modela.
- *Matematički* (*kvantitativni*) modeli - opisuju sistem pomoću relevantnih jednačina sistema ili statističkih parametara, kao što su sredina, režim rada, varijanse ili koeficijenati regresije. Matematički modeli uključuju analitičke i numeričke modele, a statistički modeli pomažu u otkrivanju obrazaca i osnovnih relacija između skupova podataka.

U nauci o menadžmentu se izučavaju različiti, pre svega matematički (kvantitativni) modeli i njihova primena u podršci odlučivanju.

1.4 Automatizacija podrške odlučivanju

Prema [1], početak računarske podrške odlučivanju početkom sedamdesetih godina prošlog veka je razrada Sajmonove ideje da proces odlučivanja zavisi od problema koji se rešava, odnosno od stepena strukturiranosti problema, koji može biti u rasponu od strukturiranih (programiranih) do nestrukturiranih (neprogramiranih) problema:

1. *Strukturirani* problemi su rutinski, repetitivni problemi, za koje postoje standardni metodi rešavanja. Strukturirani problemi imaju strukturirane sve faze rešavanja, procedure nalaženja najboljeg rešenja su poznate, a ciljevi jasno definisani. Kao računarska podrška se mogu koristiti sistemi za podršku menadžmentu, npr. za izradu budžeta, strategiju investiranja i sl.

- *Nestrukturirani* problemi su složeni, nedovoljno jasno definisani problemi, za koje ne postoje unapred poznati metodi njihovog rešavanja. Nemaju strukturirane faze, a za rešavanje ovakvih problema koriste se iskustvo i intuicija donosioca odluke.
- *Delimično strukturirani problemi* imaju strukturirane neke faze ili njihove delove. Za rešavanje strukturiranih delova problema odlučivanja koriste se standardne procedure rešavanja, dok se za nestrukturirane koriste iskustvo, ljudsko rasuđivanje i intuicija.

Nazivi *programirani* i *neprogrermirani* u nauci o mendžmentu (ili operacionim istraživanjima) nemaju značenje koje su kasnije dobili u računarskim disciplinama, već označavaju mogućnost ili nemogućnost tretiranja nekog problema matematičkim metodima.

Savremeno kompetitivno poslovno okruženje suviše je složeno i promenljivo, tako da klasičan pristup donošenju odluka nije dovoljan. Da bi se zadovoljili kriterijumi kvaliteta i brzine donošenja odluka, neophodna je računarska podrška. Neke prednosti računarske podrške odlučivanju su [1]:

- *Brzina* - veliki broj računanja može se realizovati brzo i po niskoj ceni, što omogućava analizu veoma velikog broja alternativa;
- *Prevazilaženje ograničenja* u obradi i memorisanju podataka, koja mogu smanjiti sposobnost rešavanja problema (npr. loše pamćenje);
- *Poboljšana komunikacija i saradnja* u grupnom radu, posebno kada su članovi tima fizički dislocirani;
- *Bolje upravljanje podacima* koje se ostvaruje kroz informacione sisteme, Web resurse i upotrebu namenskih skladišta i metoda istraživanja podataka.

- *Tehnička podrška* u memorisanju, pretraživanju i prenosu podataka omogućava bržu i jeftiniju realizaciju;
- *Kvalitet podrške*, koji je vidljiv kroz bolje odluke, obradu više alternativa, analizu rizika i ekspertsku podršku;
- *Smanjenje cene* koštanja podrške;
- *Agilnost* podrške omogućava brzo donošenje dobrih odluka, čak i kada se ne raspolaže svim neophodnim znanjem;
- *Dostupnost podrške* putem globalne računarske mreže, posebno korišćenjem mobilnih uređaja i bežičnih mreža.

Treba napomenti da podrška donošenju odluka ne mora biti samo računarska i da su tehnike podrške odlučivanju razvijene pre masovne upotrebe računara.

1.4.1 Računarska podrška strukturiranom odlučivanju

Prilikom rešavanja strukturiranih problema koriste se matematički modeli odlučivanja i modifikovani proces odlučivanja:
1. Definisanje problema (problemske situacije),
2. *Razvrstavanje problema* u neku od poznatih kategorija,
3. Konstrukcija *modela* koji opisuje realni problem,
4. Identifikacija i ocena mogućih rešenja modeliranog problema,
5. Poređenje, izbor i preporuka mogućih rešenja problema.

Kao primer strukturnog problema će se uzeti poznati problem organizacije transporta robe, tzv. transportni problem. Računarska podrška se sastoji u automatizaciji izrade plana koji obezbeđuje minimalne troškove prevoženja.

Primer: Transportni problem

Jedan od tipičnih problema optimizacije je transportni problem, odnosno problem optimizacije transporta robe [7], [12], [13]. U ovom primeru izložiće se proces rešavanja ovog problema prema modifikovanom procesu odlučivanja namenjenom rešavanju strukturiranih problema:

1. *Definisanje problema*

   Treba organizovati transport proizvoda od više teritorijalno raspršenih proizvođača do distributivnih centara, koji se nalaze u blizini nekoliko velikih gradova, tako da se minimizuje ukupna cena transporta.

   Pri tome su poznate maksimalne mogućnosti proizvodnje svih proizvođača i potrebe distributivnih centara svakog od velikih gradova.

2. *Razvrstavanje problema u neku od poznatih katergorija*

Jedan od matematičkih modela koji se može primeniti na ovaj problem je model sistema linearnih jednačina, pošto je zavisnost između nepoznatih veličina (količina proizvoda koje treba transportovati od pojedinih proizvođača do distributivnih centara) linearna.

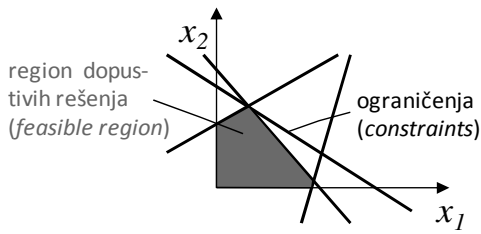
Standardni oblik problema linernog programiranja sastoji se od linearne ciljne funkcije $F(x_1,...,x_n) = c_1x_1+c_2x_2+ ... + c_nx_n$, za koju se traži maksimum, uz ograničenja postavljena na moguće vrednosti promenljivih $x_1,...,x_n$, koja se prikazuju nizom od m nejednakosti:

$a_{11}x_1+a_{12}x_2+ ... + a_{1n}x_n \leq 0$

$a_{21}x_1+a_{22}x_2+ ... + a_{2n}x_n \leq 0$

...

$a_{m1}x_1+a_{m2}x_2+ ... + a_{mn}x_n \leq 0$



Sl. 9. Problem linearnog programiranja sa dve promenljive

Traži se takvo rešenje, ako postoji, koje maksimizuje ciljnu funkciju F i zadovoljava sve postavljene uslove (ograničenja).

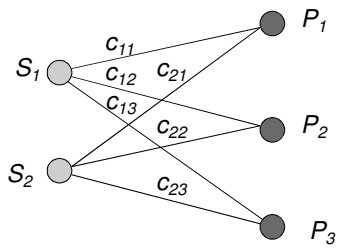
Ograničenja definišu n-dimenzionalni podprostor u kome se nalaze sva moguća rešenja problema (*feasible solutions*), tzv. region dopustivih rešenja (*feasible region*). Na Sl. 9. prikazan je slučaj za n=2, kod koga je region dopustivih rešenja dvodimenzionalni poligon.

Optimalno rešenje je tačka poligona u kojoj ciljna funkcija F ima maksimalnu vrednost, uključujući njegove ivice i temena.

Primer problema optimalne organizacije transporta robe do potrošača prikazan je na Sl. 10.

Poznata su stovarišta S_i i potrošači P_j, raspoložive i potrebne količine robe, broj i kapacitet transportnih sredstava i troškovi prevoženja.

Treba planirati niz prevoženja x_{ij} (od skladišta i do potrošača j) tako da se minimizuju troškovi prevoženja, koji se mogu izraziti sumom troškova svakog od ukupno šest pojedinačnih prevoženja
$F = c_{11}x_{11} + ... + c_{23}x_{23}$.



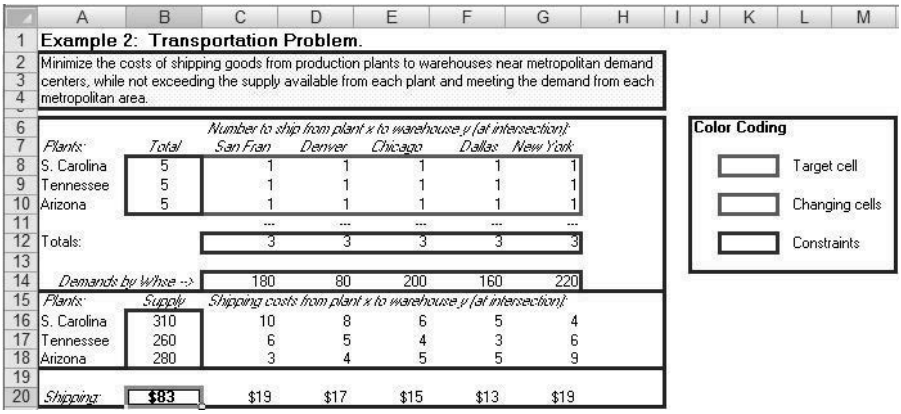
$x_{11}=?$ $x_{12}=?$ $x_{13}=?$
$x_{21}=?$ $x_{22}=?$ $x_{23}=?$

nepoznate veličine x_{ij}: količine robe za svako prevoženje

Sl. 10. Primer transportnog problema

3. *Konstrukcija modela koji opisuje realni problem*

Problem optimalne organizacije transporta robe do potrošača jedan je od primera za metod linearnog programiranja sistema Microsoft *Excel*, koji se nalazi na sistemskom fajlu SOLVSAMP.XLS, u listu (*tab*) *Shipping Routes*.



Example 2: Transportation Problem.

Minimize the costs of shipping goods from production plants to warehouses near metropolitan demand centers, while not exceeding the supply available from each plant and meeting the demand from each metropolitan area.

Number to ship from plant x to warehouse y (at intersection):

Plants:	*Total*	*San Fran*	*Denver*	*Chicago*	*Dallas*	*New York*
S. Carolina	5	1	1	1	1	1
Tennessee	5	1	1	1	1	1
Arizona	5	1	1	1	1	1
		---	---	---	---	---
Totals:		3	3	3	3	3
Demands by Whse -->		180	80	200	160	220

Plants:	*Supply*	*Shipping costs from plant x to warehouse y (at intersection):*				
S. Carolina	310	10	8	6	5	4
Tennessee	260	6	5	4	3	6
Arizona	280	3	4	5	5	9
Shipping:	$83	$19	$17	$15	$13	$19

Color Coding

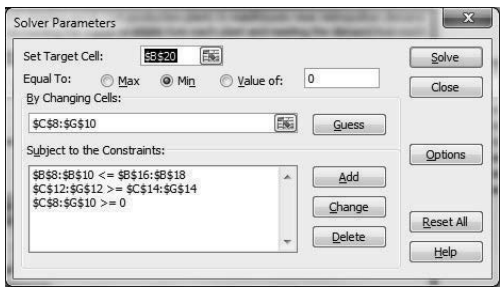
Target cell

Changing cells

Constraints

Microsoft *Excel* tabela sadrži *n* ćelija izdvojenih za promenljive i vizuelno označenih zelenom bojom. Formule koje sumiraju podatke po redovima i kolonama koriste se za kraći zapis ciljne funkcije i ograničenja u *modelu* problema.

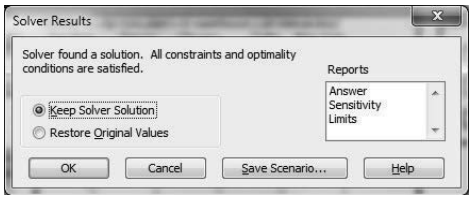
Model problema se unosi preko posebnog interfejsa MS Excel dodatka (*Ad-In*) *Solver*, koji se pokreće sa *Tools/Solver*.



4. *Identifikacija i ocena mogućih rešenja modeliranog problema*

Na ekranu interfejsa dodatka za rešavanje sistema linearnih jednačina *Solver* aktivira se taster *Solve*. Rešenje se pojavljuje u zeleno označenim ćelijama C8:G10 *Excel* tabele, a vrednost kriterijumske funkcije u njenom donjem levom delu, u ćeliji B20.

5. *Poređenje, izbor i preporuka mogućih rešenja problema*



Ponuđeno rešenje je optimalno, odnosno vrednost ciljne funkcije je maksimalna. Dobijene optimalne vrednosti mogu se zadržati (OK), ali se može vratiti i originalno stanje tabele.

Example 2: Transportation Problem.

Minimize the costs of shipping goods from production plants to warehouses near metropolitan demand centers, while not exceeding the supply available from each plant and meeting the demand from each metropolitan area.

Plants:	Total	Number to ship from plant x to warehouse y (at intersection): San Fran	Denver	Chicago	Dallas	New York
S. Carolina	300	0	0	0	80	220
Tennessee	260	0	0	180	80	0
Arizona	280	180	80	20	0	0
		---	---	---	---	---
Totals:		180	80	200	160	220
Demands by Whse -->		180	80	200	160	220

Plants:	Supply	Shipping costs from plant x to warehouse y (at intersection):				
S. Carolina	310	10	8	6	5	4
Tennessee	260	6	5	4	3	6
Arizona	280	3	4	5	5	9
Shipping:	$3.200	$540	$320	$820	$640	$880

Color Coding
- Target cell
- Changing cells
- Constraints

Pošto je, prema postavljenim kriterijumima, pronađeno optimalno rešenje, ono se može prihvatiti kao konačno ili se, u skladu s izmenjenim ili nekim dodatnim kriterijumima, neki elementi modela mogu promeniti i tražiti drugo rešenje.

1.4.2 Računarska podrška nestrukturiranom odlučivanju

Podrška standardnih kvantitativnih metoda nestrukturiranom odlučivanju je veoma mala. Za rešavanje ovakvih problema koriste se, pre svega, ljudska intuicija i rasuđivanje, kao i neka specijalizovana rešenja, na kojima se zasniva kategorija *inteligentnih* programskih sistema.

Medicinska i tehnička dijagnostika primeri su rešavanja nestrukturiranih problema. Dijagnoza se postavlja na osnovu raspoloživih simptoma oboljenja, odnosno manifestacija neželjenog ponašanja sistema. Postavljanje dijagnoze omogućava pronalaženje i eventualno otklanjanje uzroka oboljenja, odnosno kvara tehničkog sistema.

Problem dijagnostike je nestrukturiran jer:

- jednoj dijagnozi odgovara veliki broj različitih simptoma, a određeni simptom odgovara različitim dijagnozama;
- simptomi (činjenice) nisu uvek poznati niti pouzdani;
- nije poznat način na koji eksperti daju dijagnoze;
- same dijagnoze nisu potpuno pouzdane.

Kao opšti pristup rešavanju ovakvih problema razvijena je klasa metoda, odnosno sistema za podršku odlučivanju pod nazivom *ekspertni sistemi* (*Expert Systems*). Ekspertni sistemi koriste opšte *metode zaključivanja* na osnovu nepotpunih i nepouzdanih podataka i vrlo *specifično znanje* o načinima rešavanja pojedinih tipova problema.

U računarsku podršku procesa odlučivanja spadaju i komunikacione i kolaboracione tehnologije (kolaborativni i sistemi grupnog odlučivanja), kao i sistemi za upravljanje znanjem (*knowledge management*).

Primer: Dijagnostički ekspertni sistemi

Najstariji i najpoznatiji primeri ekspertnih sistema su primeri prvih sistema iz šezdesetih i sedamdesetih godina prošlog veka: DENDRAL je sistem za identifikaciju molekula u organskoj hemiji na osnovu masenih spektrograma, dok je MYCIN sistem za dijagnostiku bakterijskih infekcija i propisivanje antibiotskih terapija [1], [4].

Na Sl. 11. prikazan je deo dijaloga lekara i ekspertnog sistema MYCIN u dijagnostici i propisivanju terapija bakterijskih infekcija [14]. Nakon nekoliko desetina pitanja sistema (numerisanih rednim brojevima) i odgovora lekara (označenih dvema zvezdicama), ekspertni sistem daje svoj predlog: preciznu dijagnozu i odgovarajuću antibiotsku terapiju. Adekvatna terapija bila je od suštinske važnosti, jer su pacijenti bili u teškom stanju i bez infekcije koju je trebalo lečiti.

Sistem MYCIN je eksperimentalni sistem, koji je demonstrirao mogućnost rešavanja nestrukturiranih problema relativno jednostavnim postupkom zaključivanja pomoću velikog broja pravila različite pouzdanosti specifičnih za dijagnostiku i lečenje bakterijskih infekcija.

Rezultati eksperimenta pokazali su da je osnovni problem izgradnje ovakvih sistema zasnovanih na znanju pronalaženje, odnosno prikupljanje specijalističkog znanja, kao i da je primena ovakvih sistema u lečenju problematična zbog nedefinisane odgovornosti softverskog sistema prema pacijentima.

Razvoj novih ekspertnih sistema naglo se razvio pojavom personalnih računara osamdesetih godina, prvo uglavnom u oblasti medicine, a manje hemiji, elektronici, geologiji, menadžmentu, upravljanju procesima i vojsci,

dok su u narednom periodu preovladale primene ekspertnih sistema u proizvodnji i poslovanju [18].

```
1) Patient's name: (tirs-fast)
** FRED BRAUN
2) Sex:
** M
3) Age:
** 55
4) Are there any cultures for Fred Braun which may be related to the present
   illness, and from which organisms have been grown successluly in the
   microbiology laboratory?
** Y

--------CULTURE-1--------
5) From what site was the specimen for CULTURE-1 taken?
** BLOOD
=BLOOD
6) Please give the date and time when CULTURE-1 was obtained.
   (mo/da/yr time)
** JUN 20, 1977
   The first organism Isolated from the blood culture of 20-JUN-77
   (CULTIURE-1) will be referred to as:

--------ORGANISM-1--------
7) Enter the laboratory-reported identity of ORGANISM-i:
** UNKNOWN
6) The stain (gram or Ziehl-Neelsen acid last) of ORGANISM-i:
** NEG
9) Is ORGANISM-i rod or coccus (etc.):
** ROD
 . . .

[REC-1] My preferred therapy reccommendation is as follows:
  In order to cover for items <1 2 3 4>:
  Give: GENTAMICIN
  Dose 119 mg (6.0 ml) q8h IV fOr lO days [calculated on basis of
       1.7 mg/kg]
  Comments: Modify dose in renal failure.
```

Sl. 11. Deo tekstualnog dijaloga lekara i ekspertnog sistema MYCIN

1.4.3 Računarska podrška delimično strukturiranom odlučivanju

Za rešavanje delimično strukturiranih problema kombinuju se standardni metodi rešavanja, koji su pogodni za strukturirane delove problema i za koje postoje matematički metodi, s ljudskim rasuđivanjem i specijalizovanim metodima rešavanja, za ostale delove kompletnog problema.

Sistem za podršku odlučivanju može uticati na povećanje kvaliteta informacija potrebnih za odlučivanje, jer omogućava razmatranje većeg broja alternativa, kao i analizu i opis mogućih posledica pojedinih odluka.

1.4.4 Softverski alati za razvoj sistema za podršku odlučivanju

U softverske alate za razvoj sistema za podršku odlučivanju spadaju:

- opšti alati za razvoj informacionih sistema i integraciju aplikacija, kao što su razvojna okruženja i sistemi za upravljanje bazama podataka (DBMS);
- posebni alati za razvoj sistema za podršku odlučivanju, kao što su skupovi alata poslovne inteligencije (*Business Intelligence Suits*);
- namenski alati za istraživanje podataka (*Data Mining*) i razvoj sistema zasnovanih na znanju (*Expert Systems*).

Alati za razvoj sistema za podršku odlučivanju i istraživanje podataka (ili otkrivanje znanja) u današnjim softverskim sistemima isporučuju se kao komponente razvojnih sistema za upravljanje bazama podataka, kao što su Oracle 11g, IBM DB2 i Microsoft SQL Server 2008/2012/2014/2016.

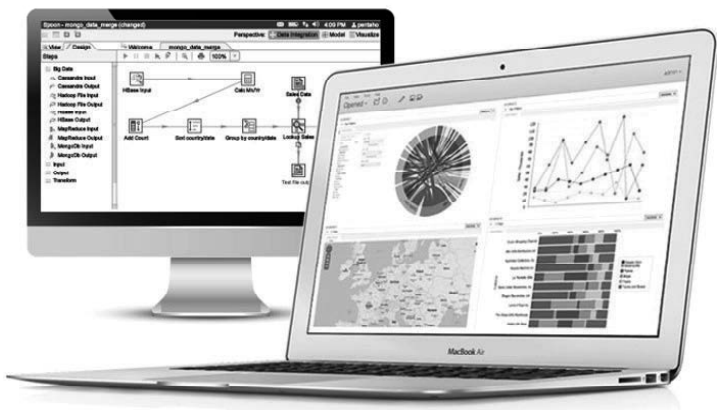
Za njihovu uspešnu upotrebu u razvoju sistema za podršku odlučivanju potrebno je poznavanje metoda koju su uključeni u ove pakete, kao načina njihove pravilne upotrebe.

Ilustracija: Primeri alata za razvoj sistema za podršku odlučivanju

1. Sistem poslovne inteligencije *Pentaho*

Sistem *Pentaho* je softverski alat otvorenog koda, koji se zasniva na Java platformi i sistemu otvorenog koda za istraživanje podataka (*data mining*) Weka [15]. Osnovne funkcije kompletnog sistema *Pentaho* su, Sl. 12:

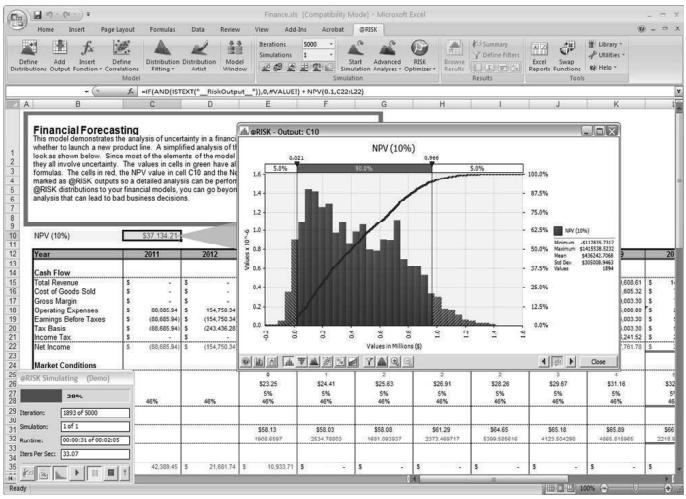
- integracija podataka iz više izvora (podsistem *Kettle*),
- izveštavanje i vizualizacija,
- izgradnja upravljačkih tabli (podsistem *Dashboard Designer*),
- istraživanje/otkrivanje znanja (podsistem *Weka*).



Sl. 12. Sistem poslovne inteligencije Pentaho

2. Sistem Palisade *DecisionTools Suite*

Sistem Palisade *DecisionTools Suite* je integrisani skup programa za analizu rizika i podršku donošenju odluka u uslovima neizvesnosti, Sl.15. [75]. Koristi se kao nadgradnja sistema Microsoft *Excel*.



Sl. 13. Radni list sistema Excel i funkcije dodatka Palisade@RISK

Osnovne komponente ili podsistemi sistema *DecisionTools Suite* su:

- *@RISK* za Monte Karlo simulacije,
- *PrecisionTree* za učenje stabala odlučivanja,
- *TopRank* za "what if" analizu osetljivosti,
- *StatTools* za statističke analize i predviđanja,
- *NeuralTools* za predikciju korišćenjem modela veštačkih neuronskih mreža (ANN),
- *Evolver* i *RISKOptimizer* za različite optimizacije.

3. Sistemi Microsoft SQL Server *Business Intelligence* i Microsoft *SharePoint*

   Kompanija Microsoft svoje komercijalne alate poslovne inteligencije i poslovne analitike isporučuje uz sistem za upravljanje bazama podataka, a njihovo korišćenje velikom broju korisnika omogućava u paketima kao što su Microsoft *Office* i *SharePoint*:

   - SQL Server 2008 *Business Intelligence Development Studio* (BIDS)
   - SQL Server 2012/2014/2016 *Data Managemet Tool* (DT)
   - *SharePoint 2010/2013/2015*

1.5 Nastanak i razvoj sistema za podršku odlučivanju

Istraživanja i razvoj računarski zasnovanih sistema za podršku odlučivanju traju gotovo 50 godina. Početkom sedamdesetih godina dvadesetog veka pojavio se koncept *sistema za podršku odlučivanju* (*Decision Support Systems*) kao softverskih alata koji pomažu menadžerima u donošenju odluka u problemima različite složenosti, od tzv. *strukturiranih* problema, za koje postoje razrađeni razrađeni matematički modeli i metodi rešavanja do *nestrukturiranih*, za čije rešavanje menadžeri uglavnom koriste iskustvo i intuiciju. Metodi operacionih istraživanja razvijeni su za strukturirane probleme, dok je za nestrukturirane razvijena klasa tzv. *inteligentnih* sistema, kao što su npr. *ekspertni sistemi* i drugi sistemi zasnovani na znanju, automatizovanom rezonovanju i metodima mašinskog učenja.

Od prvih sistema zasnovanih na modelima šezdesetih godina, preko razvoja teorija i sistema finansijskog planiranja sedamdesetih i implementacije sistema za *what-if* analize zasnovanih na elektronskim tabelama (*spreadsheets*) i grupnih sistema za podršku odlučivanju početkom osamdesetih godina prošlog veka koncept podrške odlučivanju evoluirao je do savremenih sistema zanovanih na metodima i tehnologiji tzv. *inteligentnih sistema.*

Ranih devedestih razvila se tehnologija skladišta podataka (*Data Warehouses*), menadžerski informacioni sistemi (*Executive Information Systems*), sistemi za interaktivne analitičke obrade (OLAP) i sistemi poslovne inteligencije (*Business Intelligence Systems*). Sredinom devedesetih pojavili su se sistemi za podršku zasnovani na znanju (*knowledge-driven DSS*) i Web-zasnovani sistemi.

Sredinom devedesetih pojavio se termin *poslovna inteligencija* (*Business Intelligence*). Danas se pod tim nazivom podrazumeva skup alata i tehnika koji korisnicima omogućava (1) izradu izveštaja i postavljanje upita, otkrivanje informacija i višedimenzionalni uvid u podatke s mogućnošću interaktivne promene nivoa detaljnosti, (2) napredne analitičke metode, korišćenjem statističkih, matematičkih i drugih modela i (3) istraživanje podataka, gde se traže nove, prethodno nepoznate i potencijalno korisne relacije u podacima, pre svega u velikim bazama i skladištima podataka.

Naziv *sistemi za podršku odlčivanju* danas se koristi i kao opšti naziv za *skup metoda* i kao naziv za posebne *softverske sisteme i aplikacije*, koji pružaju podršku donosiocima odluka [1].

Sistemi za podršku odlučivanju imaju određene zajedničke elemente, od kojih je najvažniji postojanje *modela*, koji omogućava analizu i interpretaciju podataka u konkretnim problemskim situacijama. Inteligentni sistemi, osim podataka i modela, imaju komponentu za upravljanje znanjem. Podrška odlučivanju je

vezana i za dobru komunikaciju s donosiocima odluka, pa je veoma važna komponeta sistema *korisnički interfejs*, koji omogućava konciznu vizualizaciju, npr. pomoću upravljačkih tabli (*dashboards*) i govornu komunikaciju i komunikaciju u prirodnom jeziku, za šta se koriste metodi *veštačke inteligencije*.

Razvoj tehnologije za podršku odlučivanju poslednjih decenija doveo je do pojave kolaborativno-komunikacionih tehnologija i sistema za podršku grupnom radu na donošenju odluka, složenih prostornih sistema za podršku odlučivanju i mobilnih aplikacija za korišćenje računarske podrške odlučivanju u svim situacijama.

Ponekad se kao zajednički naziv za sisteme i aplikacije za podršku odlučivanju i sisteme poslovne inteligencije, koji mogu biti deo različitih drugih sistema, koristi termin *sistemi za podršku menadžmentu* (*Management Support Systems*, MSS) [1].

Nove tehnologije razvoja sistema za podršku odlučivanju oslanjaju se na analizu informacija dostupnih na Internetu metodima istraživanja podataka i mašinskog učenja, kao i novim tehnologijama predstavljanja i korišćenja znanja. Razvoj ovih tehnologija omogućio je uspešnu primenu sistema za podršku odlučivanju u različitim oblastima, kao što su menadžment, obrazovanje, medicina i poljoprivreda.

Savremeni sistemi za podršku odlučivanju (ili menadžmentu) obuhvataju široku klasu softverskih rešenja koji se zasnivaju na modelima, od jednostavnih kalkulatora, preko softvera za optimizacije i simulacije, prostornih informacionih sistema i sistema za poslovne analize, do inteligentnih sistema, pre svega inteligentnih agenata i sistema zasnovanih na znanju ili ekspertnih sistema.

Softver za izgradnju ovih sistema obuhvata savremene programske jezike, komunikacione i Web tehnologije, alate poslovne inteligencije, pre svega za istraživanje podataka metodima mašinskog učenja i alate za izgradnju sistema zasnovanih na znanju.

Pitanja za ponavljanje

1. Definišite pojam donošenja odluka.
2. Koje discipline daju naučne osnove odlučivanju?
3. Čime se bavi nauka o menadžmentu?
4. Definišite sistem za podršku odlučivanju.
5. Navedite osnovne komponete sistema za podršku odlučivanju.
6. Koji su osnovni koraci procesa donošenja odluka?
7. Navedite bar tri prednosti računarske podrške procesu donošenja odluka

8. Koje vrste problema odlučivanja postoje prema strukturiranosti faza rešavanja?
9. Navedite osnovna svojstva računarske podrške rešavanju strukturiranih problema.
10. Koji matematički model se koristi za rešavanje transportnog problema?
11. Navedite osnovna svojstva računarske podrške rešavanju nestrukturiranih problema.
12. Navedite osnovna svojstva računarske podrške rešavanju delimično strukturiranih problema.
13. Navedite bar tri softverske tehnologije za podršku pojedinim fazama procesa odlučivanja.

2 Osnove teorije odlučivanja

1. Osnovni pojmovi
2. Modeli odlučivanja
3. Proces donošenja odluka
4. Statistička teorija odlučivanja
5. Alternativne teorije odlučivanja

U ovom poglavlju se ukratko izlažu osnove teorije odlučivanja, uvode osnovni pojmovi iz statističke teorije odlučivanja i daju primeri alternativnih teorija, koje predstavljaju osnov za realizaciju ekspertnih sistema, važne kategorije sistema za podršku odlučivanju.

2.1 Osnovni pojmovi

Teorija odlučivanja je skup znanja i odgovarajućih analitičkih tehnika različitog stepena formalnosti dizajniranih da pomognu donosiocu odluka u izboru alternative na osnovu njenih mogućih posledica [1], [3]. U odnosu na broj i vrstu mogućih alternativa, razlikujemo odlučivanje u uslovima *sigurnosti*, *rizika* i odlučivanje u uslovima *neizvesnosti*.

Odlučivanje u uslovima sigurnosti znači da svaka alternativa ima jednu i samo jednu posledicu, a izbor između alternativa je ekvivalentan izboru između njihovih posledica. U odlučivanju u uslovima rizika, svaka alternativa ima jednu od nekoliko mogućih posledica, a verovatnoća pojave svake posledice je poznata. Kada su distribucije verovatnoća nepoznate, radi se o odlučivanju u uslovima neizvesnosti.

Teorija odlučivanja (*decision theory*) bavi se pitanjima značajnim za ocenu neke date odluke i donošenje optimalne odluke. Predstavlja interdisciplinarno područje, deo je matematike i statistike, veoma je bliska matematičkoj teoriji igara (*game theory*).

Teorija igara je grana primenjene matematike koja se koristi u društvenim naukama (ekonomiji, filozofiji, političkim naukama), inženjerstvu i prirodnim naukama [12]. Koristi matematičke modele za analizu procesa odlučivanja (manjeg broja) učesnika. Igre se mogu klasifikovati na različite načine, npr.

- *kooperativne igre*, kada igrači sarađuju u zajedničkom interesu i *nekooperativne* (oponentske) igre;
- *igre sa fiksnom sumom*, kada se dobitak deli među igračima i *promenljivom sumom*, koja zavisi od izabrane strategije;
- *statičke igre*, kada se sve odluke donose istovremeno i *dinamičke (sekvencijalne) igre*, kada se odlučuje u više faza ili rundi.

Primer: Igra zatvorenikova dilema

Odvojeno se ispituju dvojica osumnjičenih za koje policija zna da su počinili zločin, ali nema dovoljno dokaza. Policija dvojici osumnjičenih nudi dogovor:

- ako jedan od njih prizna, biće oslobođen, a drugi osumnjičeni će biti osuđen na maksimalnu kaznu od 20 godina zatvora;
- ako priznaju obojica, dobijaju manju kaznu, po 5 godina zatvora;
- ako nijedan ne prizna, obojica će biti osuđeni na po 6 meseci zatvora zbog posedovanja oružja.

Zatvorenici ne mogu međusobno da komuniciraju.

Postavlja se pitanje, koja strategija je optimalna za svakog od zatvorenika?

Ako se odgovor daje samo jednom (*statička* igra, jedna runda) i svakom zatvoreniku odgovara niža kazna, više se isplati da priznaju (*nekooperativna* igra). Bez obzira šta drugi zatvorenik uradi, osumnjičenom sledi manja kazna. Ako se pitanje postavlje više puta zaredom, igrači imaju mogućnost da u narednoj rundi kazne jedan drugog za izdaju u prethodnoj rundi.

Neizvesnost vezana za buduća stanja stvarnosti tipična je za probleme odlučivanja koje razmatra teorija igara. Razmatraju se sve moguće alternative za svakog od igrača i procenjuje njihova isplativost. Procene posledica svake od mogućih alternativa sumiraju se u *matrici odlučivanja*:

	B ćuti	**B priznaje**
A ćuti	obojica po 6 meseci	A - 20 godina B - *slobodan*
A priznaje	A - *slobodan* B - 20 godina	obojica po 5 godina

Slična situacija odlučivanja u uslovima potpune neizvesnosti je tzv. trka u naoružanju. Eksperimentalna provera ponašanja pojedinaca u sličnoj situaciji pokazala je da 40% ljudi bira ćutanje.

Normativne i deskriptivne teorije odlučivanja

Normativne teorije odlučivanja bave se načinom na koji *treba* donositi odluke. Uvek se traži najbolja odluka, pri čemu se podrazumeva idealan donosilac odluke, koji je potpuno informisan i racionalan [1], [16].

Deskriptivne teorije opisuju način na koji se odluke zaista donose, odnosno razmatraju praktičnu primenu normativne teorije (*decision analysis*). Usmerene su na pronalaženje alata, metoda i softvera za pomoć u donošenju boljih odluka.

Osnovni softverski alati za podršku odlučivanju su sistemi za podršku odlučivanju (*decision support systems*).

Najvažnija svojstva odluka su [16]:

1. *Izbor u uslovima neizvesnosti*

   Predstavlja suštinski problem teorije odlučivanja.

2. *Intertemporalni izbor*

   Razmatra se problem donošenja optimalne odluke, kada se ishodi realizuju u različitim vremenskim trenucima, npr. da li ići na odmor sada ili uložiti sredstva u penzijski fond za budućnost.

3. *Postoji kompeticija donosilaca odluka*

   Uračunava se odnos ljudi prema odluci, čime se bavi teorija igara.

4. *Složene odluke*

   Poseban problem je donošenje odluka u složenim situacijama ili složenoj organizaciji odlučivanja, gde je problem prvo definisati šta je optimalno rešenje.

Paradoks izbora

Paradoks izbora (*paradox of choice*) pojava je kad veća mogućnost izbora može dovesti do lošije odluke ili čak do nemogućnosti donošenja bilo kakve odluke [1]. Suviše veliki izbor otežava odlučivanje, pošto je potrebno razmotriti i proceniti neprihvatljivo veliki broj alternativa, Sl. 14, što je isto kao da i nema izbora.



(a)



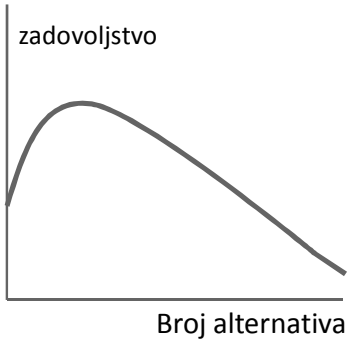
(b)

Sl. 14. Ilustracija suviše malog izbora (a) i suviše velikog izbora (b)

Problem izbora javlja se u procesu kupovine, gde kupci donose odluke u procesu izbora proizvoda. U procesu kupovine učestvuju dve vrste kupaca [1], [61]:

- oni koji uvek traže najbolje (perfekcionisti, *maximizers*), koji moraju uvek da istraže sve alternative da bi bili sigurni u svoj izbor;
- oni koji traže dovoljno dobro, prema svojim standardima (*satisficers*) i ne brinu zbog postojanja mogućniosti boljeg izbora.

Sloboda i autonomija izbora treba da povećaju osećaj zadovoljstva kupaca. No, u praksi, velika mogućnost izbora u razvijenom svetu *nije* dovela do takve pojave, kao što se vidi na Sl. 15 [17].



Sl. 15. Mogućnost izbora i zadovoljstvo potrošača

U praksi se pokazalo da nije dobro kad nema izbora, ali ni kada je izbor prevelik, odnosno da je najbolje kada postoji ograničen izbor.

2.2 Modeli odlučivanja

Donošenje odluka je složen proces, koji podrazumeva određeno vremensko trajanje i može se podeliti u više faza, obično tri do šest.

Prvu opštu teoriju procesa donošenja odluke dao je francuski filozof Condorcet[4], koji je proces odlučivanja podelio u tri faze. U prvoj se razmatraju opšti principi i posledice pojedinačnog odlučivanja, u drugoj se pojedinačna mišljenja porede i kreira manji broj opštih mišljenja, dok se u trećoj bira jedna od tih alternativa [16].

Ipak, moderni pristup analizi procesa odlučivanja je u dvadesetom veku ustanovio Džon Djui[5], koji je proces rešavanja problema podelio u pet sukcesivnih faza: (1) ustanovljavanje problema, (2) definisanje karaktera problema, (3) predlog mogućih rešenja, (4) evaluacija predloga i (5) dalje posmatranje i eksperiment, koji omogućavaju prihvatanje ili odbijanje predloga.

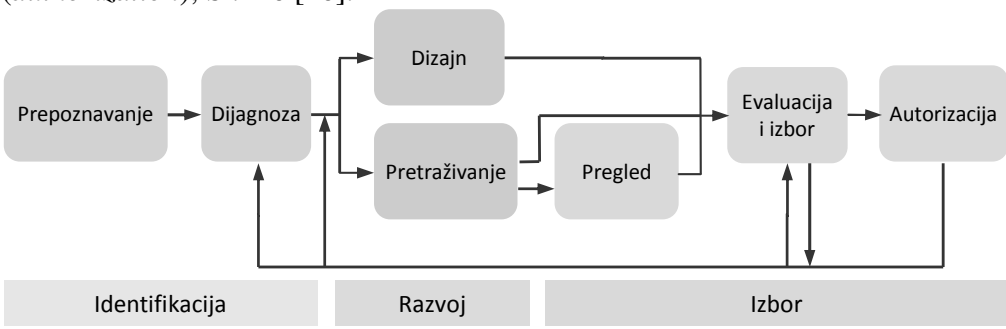
Herbert Sajmon je prilagodio ovaj proces donošenju odluka u organizacionim sistemima i ustanovio tri glavne faze: (1) prikupljanje informacija (*intelligence*), (2) kreiranje mogućih rešenja ili pravaca delovanja (*design*) i (3) izbor rešenja (*choice*). Kompletan proces rešavanja problema dobija se kada se procesu donošenja odluke pridruži i (4) proces njene primene (*implementation*).

Kada se prve dve faze ovog procesa podele (1.1 identifikacija problema, 1.2 prikupljanje informacija, 2.1 kreiranje mogućih rešenje i 2.2 evaluacija mogućih rešenja), dobija se proces koji donošenje odluka posmatra u pet, odnosno šest faza, zajedno s implementacijom odluke.

[4] Condorcet - Marie Jean Antoine Nicolas de Caritat, marquis de Condorcet (1743-1794)
[5] John Dewey (1859-1952), američki filozof, socijalni kritičar i teoretičar obrazovanja

Proces donošenja odluka nije sekvencijalan [1], [16]. Pojedine faze su vremenski različito raspoređene, zavisno od odluke koja se donosi.

Osnovne faze odlučivanja se, po Sajmonu, dele na manje celine ili rutine. Prva faza prikupljanja informacija deli se na dve rutine, prepoznavanje odluke (*decision recognition*) i dijagnozu (*diagnosis*). Faza kreiranja rešenja se takođe deli na dve rutine, traženje (*search*) i kreiranje (*design*) mogućih rešenja. Faza izbora se deli na tri rutine - preispitivanje alternativa (*screen*), ocena i izbor najbolje alternative (*evaluation-choice*) i eventualno odobrenje odluke (*authorization*), Sl. 16 [16].



Sl. 16. Relacije između faza i rutina procesa odlučivanja

Poređenje različitih razmotrenih pristupa dekompoziciji procesa odlučivanja je na Sl. 17 [16].

Condorcet	Prvo razmatranje			Drugo razmatranje		Rešenje
Simon	Prikupljanje informacija		Dizajn	Izbor		
Mintzberg	Prepozna-vanje	Dijagnoza	Pretraživanje /Dizajn	Prikaz	Evaluacija i izbor	Autorizacija
Brim	Identifikacija	Prikupljanje informacija	Kreiranje rešenja	Evaluacija	Izbor	

Sl. 17. Poređenje dekompozicije procesa odlučivanja različitih autora

Istraživanja su pokazala da se menadžeri najveći deo vremena bave kreiranjem rešenja, dobar deo vremena pronalaženjem informacija, dok izboru rešenja posvećuju mali deo vremena. S druge strane, teorija odlučivanja se upravo najviše bavi samim procesom izbora najboljeg rešenja, odnosno rutinom ocene i izbora najbolje alternative [16].

Teorija društvenog odlučivanja

Pravila odlučivanja razvijena za donošenje odluka pojedinaca mogu se primeniti i u grupnom odlučivanju, kad tretiraju organizaciju kao jedinstvenu celinu.

Teorija društvenog ili kolektivnog odlučivanja (*social decision theory*) modelira situacije u kojima odluke donose dve ili više osoba, koje mogu da imaju suprotstavljene ciljeve ili poglede na način postizanja ciljeva, koje realizuju otvorenim sukobom ili kompromisom. Najvažniji aspekt teorije su metodi kombinovanja individualnih preferencija i izbora u kolektivne odluke. Najčešće su u pitanju glasanje, pregovaranje i sl. Osnovni problem je pronalaženje racionalnog načina kombinovanja pojedinačnih preferencija u skup društvenih preferencija, odnosno grupnu odluku [16].

Osnove teorije društvenog odlučivanja dali su Borda i Condorcet, koji su primetili da jednostavno većinsko pravilo odlučivanja daje nestabilnu većinu, pošto se može formirati manjinska koalicija [16].

Polazna tačka moderne teorije društvenog odlučivanja je teorema o nemogućnosti, koju je formulisao Kenet Arou[6] (1951). On je pokušao da pronađe pravilo društvenog odlučivanja koje bi bilo drugačije od većinskog odlučivanja, kako bi se izbegao problem cikličnih društvenih preferencija. Negativan rezultat je čuvena teorema Aroua, po kojoj svako pravilo društvenog odlučivanja, ako zadovoljava četiri naizgled razumna kriterijuma racionalnosti, ne može izbeći cikličnost, odnosno da je nemoguće postojanje funkcije društvenog izbora. To znači da nije moguće naći mehanizam odlučivanja koji bi doneo kolektivan izbor na osnovu individualnih preferencija, odnosno koji ne bi bio diktatorski [16], [61].

Na ovom području postoji više rezultata slične prirode. Kad se niz alternativa proširi sa jednostavnih lista preferencije na skup tačaka u Euklidskom prostoru, nemogućnost je još izrazitija nego prema teoremu Aroua. Skoro svi važniji rezultati teorija društvenog odlučivanja su negativne prirode, što pokazuje da neki zahtevi racionalnosti društvenog odlučivanja nisu kompatibilni [16].

2.3 Proces donošenja odluka

Donošenje odluka podrazumeva izbor one alternative koja daje najbolji ishod na osnovu nekog kriterijuma. Teorija odlučivanja pretpostavlja da postoji takav kriterijum, koji se može jasno i precizno izraziti.

Najčešći načini ocenjivanja alternativa su relacioni izrazi (kriterijumi preferencije) i numeričke vrednosti. Kriterijumi preferencije omogućavaju međusobno poređenje alternativa, ali se javljaju problemi ako nisu kompletni

[6] Kenneth Arrow (1921-), američki ekonomista, dobitnik Nobelove nagrade, jedan od vodećih ekonomskih teoretičara 20. veka, formulisao teoremu o nemogućnosti.

(ne omogućavaju međusobno poređenje svih alternativa) i tranzitivni, jer dolazi do kontradikcija i nemogućnosti izbora najbolje alternative. Na primer [16], kako izabrati CD Mocarta, Hajdna ili Betovena, pri čemu imamo samo tri kriterijuma preferencije: "Mocart > Hajdn", "Hajdn > Betoven" i "Betoven > Mocart"?

Numerički izraz vrednosti alternative omogućava apsolutno uređenje prostora svih mogućih alternativa, njihovo rangiranje i izbor najbolje alternative. Osnovni problem je u smislu i načinu određivanja tih vrednosti. U ekonomiji one obično izražavaju vrednost novca, a u većini ostalih oblasti apstraktnu veličinu, koja se naziva korist (*utility*).

U teoriji odlučivanja uvek se bira alternativa koja daje najveću korist ili dobitak, a ako ima više alternativa s istom vrednošću, slučajno se bira bilo koja od njih. To je pravilo maksimizacije koristi (*rule of maximization*).

Matrice odlučivanja

Standardni prikaz problema odlučivanja je pomoću matrice odlučivanja (*decision matrix*). Redovi matrice su moguće alternative, kolone matrice su moguća stanja okruženja, a elementi odgovaraju ishodima. Obično se podrazumeva da je skup alternativa zatvoren, odnosno da obuhvata sve mogućnosti i da se one međusobno isključuju. Ishodi mogu biti različiti, verbalni ili neki kriterijumi preferencije, ali se najčešće koriste numeričke vrednosti, koje definišu korist ishoda (matrice koristi), Sl. 19 [16].

	pada kiša	suvo vreme
poneti košobran	normalna odeća, veliki kofer	normalna odeća, veliki kofer
ne nositi kišobran	kišni mantil, mali kofer	normalna odeća, mali kofer

(a)

	pada kiša	suvo vreme
poneti košobran	15	15
ne nositi kišobran	0	18

(b)

Sl. 18. Primer (a) opšte matrice odlučivanja i (b) matrice koristi

Očekivana korist

Očekivana korist (*expected utility*) svakog od n ishoda dobija se na osnovu vrednosti koristi u dodeljene ishodu i verovatnoće odgovarajućeg stanja okruženja p kao težinskog faktora, $EU = \sum_1^n p_i \times u_i$. Prema teoriji očekivane koristi, nastaloj kad i teorija verovatnoće, bira se alternativa koja daje maksimalnu očekivanu korist (*maximal expected utility*, MEU).

Ovakva definicija koristi ima kao posledicu paradoks poznat iz torije verovatnoće, nazvan po Sankt Peterburgu [3], [16]. Posmatra se igra na sreću, koja se sastoji u bacanju novčića, sve dok se prvi put ne pojavi glava. Ako padne pismo u prvom bacanju, dobija se jedan zlatni novčić, nakon dva bacanja

dva novčića, a nakon n uzastopnih pisama 2^n zlatnih novčića. Verovatnoća pojavljivanja glave nakon n bacanja je $1/2^n$. Maksimalna očekivana korist ili dobitak igre je $\sum_1^n 1/2^n \times 2^{n-1} = \sum_1^n (1/2)$, za $n \to \infty$, odnosno beskonačan. Rešenje problema predložio je još D. Bernuli, tako što se maksimizuje očekivana *subjektivna* korist, odnosno u ovom slučaju, korist ne treba da raste linearno, već logaritamski kao $log(w)$, gde je w bogatstvo igrača. Tako dolazi do zasićenja, odnosno kad igrač postane veoma bogat, ne želi dalje da se kocka i rizikuje gubitak svega. Ovakav pristup je npr. uobičajen u ekonomskim problemima, dok se u analizi rizika pretežno upotrebljava očekivana objektivna korist.

Računanje očekivane koristi povezano je s estimacijom objektivnih verovatnoća, što se često vrši na osnovu empirijskih frekvencija događaja. Razlika između ovih vrednosti i subjektivnih verovatnoća, koje su nepouzdane, zavisi od kompetentnosti samih ocenjivača (problem kalibracije) [16].

Izbor u uslovima neizvesnosti

Specijalni slučajevi teorije odlučivanja u statistici su testiranje hipoteza i statistička estimacija. Primer *izbora u uslovima neizvesnosti* je tzv. Paskalova opklada (*Pascal's wager*).

Primer: Paskalova opklada

Po Paskalu, neizvesno je da li bog postoji, ali treba doneti odluku da li verovati u boga ili ne. Paskal je proces odlučivanja u uslovima neizvesnosti u ovom primeru razmatrao kao neizvesnost bacanja novčića ili opklade, a ishode je analizirao i procenio kao u matrici odlučivanja:

	Bog postoji	Bog ne postoji
Verovati	+∞ (raj)	konačno
Ne verovati	-∞ (pakao)	konačno

U slučaju da bog postoji, dobitak verovanja u boga je beskonačan (raj), a u slučaju da bog ne postoji gubitak je konačan (nešto manja sloboda uživanja i ponašanja), odnosno mali u poređenju s dobitkom.

Gubitak u slučaju neverovanja je beskonačan (pakao), dok su uživanja zemaljskog života opet konačan dobitak, takođe mali u poređenju s mogućim gubitkom.

Zaključak ovakve analize je da, koliko god bila mala verovatnoća postojanja boga, očekivani dobitak verovanja prevazilazi dobitak neverovanja, tako da je bolje verovati u boga. To nije bila rasprava o postojanju boga, jer, kako je gotovo vek kasnije prokomentarisao filozof

> Volter, interes koji imamo da verujemo u nešto nije dovoljan dokaz da to i postoji.

U teoriji odlučivanja razmatra se neizvesnost, koja se razlikuje od pojma verovatnoće, pa se u okviru teorije javljaju specifični paradoksi [9], [16].

Teorija odlučivanja koristi različite kriterijume za donošenje odluka u uslovima neizvesnosti, kao što su npr. maksimin, maksimaks, minimaks i metode s težinskim faktorima. Najbolji metod odlučivanja u uslovima neizvesnosti za konkretni problem se ustanovljava eksperimentalno, poređenjem rezultata više različitih metoda [16].

Kriterijumi odlučivanja u uslovima potpune neizvesnosti

Razmatraju se situacije u kojima se ne znaju verovatnoće poznatih stanja stvarnosti [16]. Klasični primer potpune neizvesnosti je varijanta prikazanog primera predviđanja vremenskih prilika. Učestvuje se u takmičenju, čija je prva nagrada tajno putovanje avionom na jednonedeljni odmor na nepoznatoj lokaciji. Ne zna se gde je to mesto, a verovatnoća da će padati kiša je bilo koja vrednost 0..1. Matrica odlučivanja je kao na Sl. 19 (a):

	pada kiša	**suvo vreme**
poneti košobran	normalna odeća, veliki kofer	normalna odeća, veliki kofer
ne nositi kišobran	kišni mantil, mali kofer	normalna odeća, mali kofer

Bez informacija o koristi, preferencije ishoda mogu biti:

"normalna odeća, mali kofer"
je bolje od
"normalna odeća, veliki kofer"
je bolje od
"kišni mantil, mali kofer"

Pravilo *maksimina* predviđa da se maksimizuje minimalna korist. U uslovima neznanja, za svaku alternativu se definiše nivo sigurnosti kao najgori slučaj te alternative. Pravilo maksimina maksimizuje nivo sigurnosti, odnosno daje najbolji od najgorih mogućih ishoda. U primeru prognoze vremena, pravilo maksimina traži da se izabere alternativa u kojoj se ponese kišobran.

Ovo pravilo ne može da razlikuje ishode s istim nivoom sigurnosti, već se koristi pravilo leksikografskog maksimina ili *leximin*, po kome se u tom slučaju alternative razlikuju po sledećem najgorem ishodu itd.

Obe varijante maksimina su pesimističke. Optimistička varianta je *maximax*, odnosno izbor najbolje alternative od onih s najboljim ishodom, u primeru

alternative gde se ne nosi kišobran. Iako ovaj pristup ne deluje racionalno, ljudi se u stvarnim situacijama ponekad ponašaju na ovaj način.

Da bi se izbegao krajnji optimizam, kao i krajnji pesimizam, potrebna je informacija o koristi, kao u primeru vremenske prognoze, Sl. 19 (b):

	pada kiša	suvo vreme
poneti košobran	15	15
ne nositi kišobran	0	18

Jedan način izbegavanja krajnosti je uvođenje indeksa optimizma ili pesimizma $\alpha \in [0..1]$. Za svaku alternativu A nivo sigurnosti je $min(A)$, a nivo očekivanja $max(A)$. Tada se α-indeks alternative računa kao $\alpha \times min(\alpha) \times min(A) + (1 - \alpha) \times max\ (A)$. Za α=0, dobija se pesimistički kriterijum maksimina, a za α=1 optimistički maksimaks [16].

Još jedan način odlučivanja je kriterijum minimaksnog rizika ili gubitka (*minimax regret/risk/loss*), odnosno samo *minimaks* (*minimax*). Matrica rizika se dobija iz matrice koristi, tako što se za određeni ishod računa razlika između vrednosti koristi ishoda i najveće vrednosti koristi u određenoj koloni matrice:

	pada kiša	suvo vreme
poneti košobran	0	3
ne nositi kišobran	15	0

Kriterijum minimaksa bira opciju koja minimizuje najveći rizik svake alternative (u primeru 3 i 15), pa bira manju vrednost 3, odnosno u razmatranom primeru alternativu "poneti kišobran".

Još jedan pristup je smanjenje neizvesnosti o riziku, odnosno pristup dovoljnog razloga, po kome, ako nema razloga da se veruje da će se neki događaj dogoditi verovatnije nego neki drugi, događajima se dodeljuje jednaka verovatnoća. U primeru prognoze, verovatnoća da pada kiša je 0,5. Problem kod ovog pristupa je osetljivost na particiju alternativa.

2.4 Statistička teorija odlučivanja

Obuhvata statističke alate i metode za organizaciju činjenica, evaluaciju rizika i podršku u donošenju odluka [9]. Donošenje odluke se posmatra kao rezultat statističkog eksperimenta, koji se analizira metodima testiranja hipoteza i ocenjivanjem parametara. Testiranje hipoteza je zaključivanje o populaciji na osnovu nekog uzorka, pri čemu se razmatra neka alternativna hipoteza, koja se može opovrgnuti ili potvrditi.

Bez obzira na rezultat, odluka može biti pogrešna, pri čemu se javljaju dve vrste grešaka:

- *greška tipa I*, kada se istina razvrsta kao laž (*False Positive*);
- *greška tipa II*, kada se laž razvrsta kao istina (*False Negative*).

Rizici greške tipa I i tipa II mogu se kvantifikovati i tako poboljšati donošenje racionalne odluke.

Primer koji ilustruje dve vrste greški odlučivanja je donošenja odluke o krivici u krivičnom postupku, čija struktura je data u matrici konfuzije (*confusion matrix*) odluke porote:

		stvarno stanje	
		kriv je	nije kriv
odluka porote	kriv je	TP	FP - *greška tipa I*
	nije kriv	FN - *greška tipa II*	TN

2.5 Teorija verovatnoće i matematička logika

Matematička logika se koristi za predstavljanje znanja i zaključivanje u uslovima izvesnosti. Formule matematičke logike koje obuhvataju *činjenice* kao što je *majka(Marija,Petar)* i pravila (tvrdnje, predikate) kao što je *majka(x,y)* $\Rightarrow$ *dete(y,x)* čine bazu znanja sistema zasnovanih na znanju. Rezonovanje se zasniva na primeni pravila zaključivanja predikatnog računa prvog reda, kao što je *Modus Ponens*:

$$\frac{p, p \Rightarrow q}{q}$$

što se interpretira kao "ako važe tvrdnje p i $p \Rightarrow q$, onda se može zaključiti (dokazati) istinitost q".

Klasična matematička logika je dovoljna za realizaciju zaključivanja, ako su zadovoljeni određeni strogi preduslovi:

- naše znanje o "stvarnosti" je potpuno;
- nema kontradiktornih činjenica ili pravila;
- sve tvrdnje su isključivo ili istinite ili lažne.

Ilustracija: Jezik Prolog i klasična logika

Interpreter jezika Prolog (od *Programmming in Logic*) zasniva se na zaključivanju unazad (*backward chaining*) nad Hornovim klauzulama, uz neke dodatne mogućnosti. Nastao je u Evropi, a koristi se za izgradnju inteligentnih sistema, uglavnom u Evropi i Japanu, gde je krajem prošlog veka usvojen kao zvanični jezik tzv. Pete generacije računara.

Program u jeziku Prolog je skup klauzula oblika

```
head :- literal1, ... , literaln.
```

Npr. logička tvrdnja "Građanin koji prodaje oružje neprijateljskim državama je kriminalac" može se formalno zapisati kao iskaz matematičke logike:

$$\text{Gradjanin}(x) \wedge \text{Oruzje}(y) \wedge \text{Prodaje}(x,y,z) \wedge \text{Neprijatelj}(z) \Rightarrow \text{Kriminalac}(x)$$

odnosno kao iskaz (klauzula) u programskom jeziku Prolog:

```
kriminalac(X):-gradjanin(X),oruzje(Y),prodaje(X,Y,Z),neprijatelj(Z).
```

Jezik Prolog ima ugrađene predikate za aritmetičke izraze i druga računanja, npr. "X **is** Y*Z+3".

Stvarnost je znatno složenija od pretpostavki koje postavlja klasična matematička logika, pa su neegzaktno znanje i rezonovanje deo svakodnevice i većine ljudskih aktivnosti i odluka.

Primer: Neegzaktno znanje i rezonovanje kao deo svakodnevice

Osoba A hoda ulicom prema autobuskoj stanici. Nekoliko stotina metara dalje A vidi nekoga i gotovo je siguran je da je to njegov prvi komšija B koji najčešće ide kolima na posao.

U veoma kratkom tekstu mogu se uočiti različite vrste neegzaktnog znanja i zaključivanja:

- nejasno znanje (*fuzzy*) - *nekoliko stotina metara*

  Definiše preslikavanje 'broj stotina' u 'nekoliko', 'mnogo', itd. Nije neizvesno ili nepotpuno, ali je procenjeno, nejasno.

- probabilističko znanje - *komšija najčešće ide kolima*

  Verovatnoća zasnovana na meri koliko često koristi automobil; računa se uvek $P(F) = 1 - P(\bar{F})$;

- uverenje (*belief*) - *to je njegov prvi komšija B*

  Predstavlja "obrazloženu pretpostavku", podrazumeva se da je istinito.

- implicitno, podrazumevajuće znanje (*default*) - *A želi da ide autobusom*

  Pretpostavka zasnovana na zdravorazumskom znanju i zaključivanju.

Postoje proširenja klasične matematičke logike, koja omogućavaju logičko zaključivanje u uslovima neizvesnosti: modalna logika, intuicionistička logika, probabilistička, nemonotona logika, itd.

Modalna logika je proširenje klasične logike, koristi modalne operatore □(*necessity*) i ◊(*possibility*), sa značenjem "neophodno istinito" i "moguće istinito". Povezani su izrazima $\Diamond P \leftrightarrow \neg\Box\neg P$ ("moguće" je ekvivalentno "nije neophodno da nije istinito") i $\Box P \leftrightarrow \neg\Diamond\neg P$ ("neophodno" je ekvivalentno "nije moguće da nije istinito").

Višeznačna logika (*multi-valued logic*) je pristup u kome se koristi više od dve diskretne vrednosti istinitosti, npr. *true*, *false* i *unknown*. Specijalni slučaj višeznačne logike je fazi logika, koja koristi beskonačan broj vrednosti istinitosti. Neprekidna/kontinualna logika (*continuous-valued logic*) koristi kao vrednost istinitosti neke tvrdnje realni broj (Lukasiewicz-Tarski, 1930).

Razvijene su i druge alternativne matematičke teorije, posebno za potrebe zaključivanja u ekspertnim sistemima.

Probabilističko zaključivanje

Probabilističko zaključivanje je izraz koji se odnosi na kombinovanje u rezonovanju (zaključivanju) mogućnosti teorije verovatnoće da tretira neizvesnost i deduktivne logike da koristi logičku strukturu. Probabilistička logika izvodi rezultate na osnovu probabilističkih izraza umesto proširenih matrica istinitosti. Dobijeni formalizam može se primeniti na široku klasu pojava.

Problem primene probabilističke logike su računska kompleksnost i neočekivani rezultati, npr. u slučaju teorije Dempstera i Šafera, gde se kao rezultat tvrdnji dvaju jednako pozdanih lekara, koji daju različite preve dijagnoze s verovatnoćom 0,99 i jednu istu drugu dijagnozu s verovatnoćom 0.01, po teoriji kao konačna odluka dobija druga dijagnoza s verovatnoćom 1,0 [95].

Metodi zaključivanja u probabilističkoj logici su različita proširenja klasične matematičke logike, kao što su višeznačna logika (*multi-valued logic*) i neprekidna ili kontinualna logika (*continous logic*). Postoje i druge alternativne teorije odlučivanja, kao što su npr. teorija faktora pouzdanosti (*certainty factors*) i teorija fazi skupova (*fuzzy sets theory*).

Teorija verovatnoće i Bajesovo pravilo

Teorija verovatnoće omogućava da se verovatnoća neke tvrdnje (hipoteze) izračuna pomoću Bajesove[7] formule (pravila), koja se izvodi iz pravila proizvoda verovatnoća [14], [16], [18]:

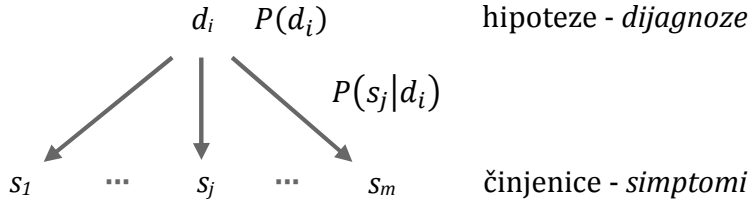
$$P(B|A) = \frac{P(A|B) \cdot P(B)}{P(A)}$$

[7] Thomas Bayes (1701-1761) engleski matematičar i sveštenik, koji je formulisao čuvenu teoremu iz teorije verovatnoće, nazvanu po njemu, koju za života nije objavio.

Bajesov pristup često se koristi za donošenje odluka u dijagnostici, gde se razmatraju neke posledice A, koje zavise od nekih (skrivenih) uzroka B. Izraz $P(A|B)$ je model koji povezuje uzroke B i posledice A.

Abduktivno zaključivanje o uzrocima B na osnovu efekata A vrši se na osnovu verovatnoće pojave uzroka B. Verovatnoća se računa na osnovu apriorne verovatnoće posledica $P(B)$ i uslovne verovatnoće pojave posledica za razmatrane uzroke $P(A|B)$, koje se dobijaju iz iskustva, na osnovu praćenja frekvencije njihovog pojavljivanja.

Dijagnostičko zaključivanje (*evidential reasoning*) razmatra hipoteze ili *dijagnoze*, koje predstavljaju *uzrok* pojave različitih *simptoma* (činjenica, posledica), kao na slici:



U praksi mogu se pronaći podaci o apriornim verovatnoćama dijagnoza $P(d_i)$ i uslovnim verovatnoćama simptoma za razmatrane dijagnoze $P(s_j|d_i)$, npr. iz medicinske statistike i podataka tehničke dijagnostike. Pomoću Bajesove teoreme mogu se izračunati aposteriorne verovatnoće nepoznatih dijagnoza na osnovu poznatih simptoma:

$$P(d_i|s_j) = P(d_i)\frac{P(s_j|d_i)}{P(s_j)}, \qquad i = 1, \dots, n, \qquad j = 1, \dots, m$$

Pri tome se pretpostavlja da se dijagnoze međusobno isključuju (*non-overlapping*) i da su potpune, odnosno da obuhvataju sve moguće slučajeve.

Poznate činjenice o konkretnom slučaju ili problemu odlučivanja su ustanovljeni simptomi $s_1, \dots, s_m$. Potrebno je pronaći takvu hipotezu d_i koja ima najveću aposteriornu verovatnoću $max_i\, P(d_i|\, s_1, \dots, s_m)$, prema izrazu:

$$P(d_i|s_1, \dots, s_m) = P(d_i)\frac{P(s_1, \dots, s_m|d_i)}{P(s_1, \dots, s_m)}, \qquad j = 1, \dots, m$$

Ako se pretpostavi uzajamna nezavisnost simptoma s_i, za zadanu hipotezu d_i, brojilac se može izračunati kao:

$$P(s_1, \dots, s_m|d_i) = \prod_{j=1}^{m} P(s_j|d_i)$$

Hipoteza s najvećom aposteriornom verovatnoćom d_i može se pronaći pomoću izraza:

$$P(d_i|s_1,\dots,s_m) = P(d_i)\cdot\frac{\prod_{j=1}^{m} P(s_j|d_i)}{P(s_1,\dots,s_m)}$$

Najverovatnija hipoteza je potencijalno najbolja dijagnoza, odnosno odluka.

2.6 Alternativne teorije odlučivanja

Primena verovatnoća u statističkoj teoriji odlučivanja oslanja se na različita pojednostavljenja i pretpostavke uz koje važe izvedeni teorijski rezultati. Praktični problemi odlučivanja ne zadovoljavaju ove stroge uslove, npr. uslov uzajamne nezavisnosti pojedinih ishoda odluke.

Zbog toga su razvijene alternativne teorije, koje nemaju tako stroge pretpostavke primenjivosti, npr.

- teorija Dempstera i Šafera (Dempster A.,1967; Schaffer G.,1976);
- teorija faktora pouzdanosti (*Certainity Factors*);
- fazi logika (Zadeh L.,1965) ;

Alternativne teorije uspešno se primenjuju u praksi, npr. u ekspertnim sistemima. Faktori pouzdanosti (*Certainity Factors*, CF) su uvedeni kod jednog od prvih ekspertnih sistema MYCIN, koji je razvijen na univerzitetu Stanford.

Ekspertni sistemi zasnovani na pravilima računaju pouzdanost zaključka nekog pravila $CF \in [-1..1]$ na osnovu poznatih faktora pouzdanosti svakog izraza iz preduslova CF_i koristeći odgovarajuću matematičku operaciju, npr. $min\{CF_1,\dots CF_n\}\cdot CF\_pravila$, ako je preduslov u obliku konjukcije.

Ipak, dokazano je da za svako pravilo odlučivanja, koje se generiše nekim od ovih metoda, postoji ili se može izvesti ekvivalentno bajesovsko pravilo, koje nikad nije manje uspešno (pouzdano), a može biti i uspešnije.

2.6.1 Faktori pouzdanosti

Faktori pouzdanosti (*Certainty Factors*) su metod zaključivanja razvijen na univerzitetu Stanford za potrebe razvoja ekspertnog sistem MYCIN [18]. Primena bajesovskog zaključivanja u medicini bila je problematična, ne samo zbog nerealnih pretpostavki o nezavisnosti simptoma, već i zbog toga što brojne uslovne verovatnoće za opštu populaciju nisu poznate [18]. U praksi se takve činjenice prikupljaju postepeno, sporo i uz znatne troškove.

Bajesova formula takođe podrazumeva inkrementalno ažuriranje podataka:

$$P(D_i|E) = \frac{P(E|D_i)P(D_i)}{P(E)} = \frac{P(E|D_i)P(D_i)}{\sum_j P(E|D_j)P(D_j)}$$

Verovatnoća koja se dobija nakon dodavanja poznatoj verovatnoći događaja E_1 verovatnoće događaja E_2 je:

$$P(D_i|E) = \frac{P(E_2|D_i \cap E_1)P(D_i \cap E_1)}{\sum_j P(E_2|D_j \cap E_1)P(D_j \cap E_1)}$$

Povećenjem broja činjenica E_i, raste i broj verovatnoća koje treba ustanoviti. Osim toga, u praksi se pokazalo da medicinski eksperti nerado razmatraju negaciju hipoteze, koja za njih predstavlja subjektivno uverenje, tako da verovatnoća hipoteze $\bar{H}$ za njih nije 1-$P(H)$.

Tako je razvijena je teorija faktora pouzdanosti, koja se zasniva na subjektivnoj pozitivnoj i negativnoj meri poverenja u neku tvrdnju, pri čemu mere ne predstavljaju verovatnoće. Mera pouzdanosti neke tvrdnje dobija se kao razlika subjektivne mere poverenja (*Measure of Belief*, MB) i nepoverenja (*Measure of Disbelief*, MD) u tu tvrdnju. Mera MB predstavlja stepen poverenja u istinitost tvrdnje H, a mera MD stepen nepoverenja u istinitost H.

Mera pouzdanosti hipoteze ili faktor pouzdanosti (*Certainty Factor*, CF) je razlika mere poverenja MB i mere nepoverenja MD u hipotezu H, uz poznate činjenice E:

$$CF(H|E) = MB(H|E) - MD(H|E), \quad -1 \leq CF(H) \leq 1$$

Vrednosti MB i MD za posmatranu hipotezu H su *subjektivne*, određuju ih eksperti na osnovu skupa činjenica E. Njihovo kombinovanje u jedinstvenu meru omogućava rangiranje hipoteza. Mere poverenja i nepoverenja mogu se definisati na osnovu apriornih i uslovnih verovatnoća $P(H)$ i $P(H|E)$ [18]:

$$MB(H,E) = \begin{cases} 1 & ,ako\ je\ P(H) = 1 \\ \dfrac{max(P(H|E),P(H)) - P(H)}{max[1,0] - P(H)} & ,inače \end{cases}$$

$$MD(H,E) = \begin{cases} 1 & ,ako\ je\ P(H) = 0 \\ \dfrac{\min(P(H|E),P(H)) - P(H)}{min[1,0] - P(H)} & ,inače \end{cases}$$

Njihova kombinacija daje faktore pouzdanosti, koji nisu verovatnoće, odnosno ne važi $CF(H,E) + CF(\bar{H},E) = 1$, već je $CF(H,E) + CF(\bar{H},E) = 0$. Npr. vrednost CF=0,7 samo znači da je stepen poverenja u tvrdnju 70% veći od stepena nepoverenja, a vrednost -0,7 da je isto toliko veće nepoverenje u tvrdnju ili hipotezu.

Pravila iz baze znanja su u prvoj verziji primenjivana samo ako su uslovi pravila bili dovoljno pouzdani, iznad nekog praga (koji je empirijski postavljen na vrednost 0,2), kako bi se sačuvala efikasnost sistema. Definicija *CF* za tvrdnje u uslovima pravila je kasnije izmenjena, radi ublažavanja negativnog uticaja jedne odrečne na veliki broj potvrdnih činjenica u uslovima pravila koja imaju mali stepen poverenja [18]:

$$CF(H,E) = \frac{MB(H,E) - MD(H,E)}{1 - min\big(MB(H,E), MD(H,E)\big)}$$

Ovakva izmena omogućila je npr. da pravilo sa *MB*=0.999 i *MD*=0,799 nema stepen poverenja 0,999-0,799=0,2, već 0,2/0,201=0,995. Vrednost faktora poudanosti u uslovima pravila računa se kao:

$$\begin{array}{ll} E_1 \ and \ E_2 & min[CF(H,E_1), CF(H,E_2)] \\ E_1 \ or \ E_2 & max[CF(H,E_1), CF(H,E_2)] \\ not \ E & -CF(H,E) \end{array}$$

Vrednost faktora pouzdanosti složenog uslova *H* nekog pravila se tako računa na osnovu faktora pouzdanosti premisa P_i u pravilu kao:

$$\begin{array}{ll} CF(H) = CF(P_1 \ and \ P_2) = min[CF(P_1), CF(P_2)] & za \ konjukciju \\ CF(H) = CF(P_1 \ or \ P_2) = max[CF(P_1), CF(P_2)] & za \ disjunkciju \end{array}$$

Faktor poverenja pravila oblika IF *E* THEN *H* računa se kao [18]:

$$CF(H,e) = CF(E,e) \cdot CF(H,E)$$

gde su:

$CF(E,e)$ - faktor pouzdanosti činjenice *E,* koja čini uslov pravila, zasnovan na neizvesnoj činjenici *e*;

$CF(H,E)$ - faktor pouzdanosti hipotreze *H,* u kojoj se pretpostavlja da je činjenica potpuno izvesna kada je $CF(E,e)$=1;

$CF(H,e)$ - faktor pouzdanosti hipotreze *H,* koja se zasniva na neizvesnoj činjenici *e*.

Kada je uslov pravila potpuno izvestan, odnosno $CF(E,e) = 1$, pouzdanost hipoteze *H* je $CF(H,e) = CF(H,E)$.

Faktor pouzdanosti integriše različita znanja eksperata i predstavlja osnovu za kombinovanje prihvatanja ili odbijanja *H* u okviru jednog ili više pravila. Pouzdanost kombinacije dva pravila se računa, zavisno od predznaka njihovih faktora pouzdanosti CF_1 i CF_2, kao [18]:

$$CF_{komb}(CF_1, CF_2) = \begin{cases} CF_1 + CF_2(1 - CF_1) & , ako\ su\ CF_i > 0 \\ \dfrac{CF_1 + CF_2}{1 - min(|CF_1|, |CF_2|)} & , ako\ je\ jedan\ CF_i < 0 \\ CF_1 + CF_2(1 + CF_1) & , ako\ su\ CF_i < 0 \end{cases}$$

Primer: Kombinovanje pravila iz baze znanja

Kombinuju se dva pravila iz baze znanja, Rule 1 i Rule 9 [20]:

Rule 1:
IF postoje kardiološki simptomi AND
tip bola u grudima je srčana bol
THEN potvrda postojanja oboljenja srca (CF=0,75)

Rule 9:
IF postoji osećaj lupanja srca
THEN potvrda postojanja oboljenja srca (CF=0,4)

Pretpostavka je da u radnoj memoriji postoje podaci: CF(postoje kardiološki simptomi) = 0,6, CF(tip bola u grudima je srčana bol) = 0,8 CF(postoji osećaj lupanja srca) = 1,0. Uslovi oba pravila imaju CF>0,2, pa će se oba primeniti i potvrditi postojanje srčanog oboljenja kod pacijenta.

Neka se prvo prvo primeni Rule 1, a zatim Rule 9. Pouzdanost elemenata uslova pravila Rule 1 su 0,6 i 0,8, tako da je pouzdanost uslova njihov minimum ili 0,6. Pouzdanost pravila Rule 1 je 0,75, tako da je pouzdanost zaključka 0,6·0,75= 0,45. Na isti način je pouzdanost zaključka pravila Rule 9 u odnosu na poznate činjenice *min*(1,0)·0,4 = 0,40.

Pouzdanost hipoteze "potvrda postojanja oboljenja srca" je na početku 0,0. Nakon primene pravila pravila Rule 1, prema izrazu (1) pouzdanost je 0,0+0,45·(1-0,0)=0,45, a nakon pravila Rule 9 je 0,45+0,40·(1-0,45)=0,67.

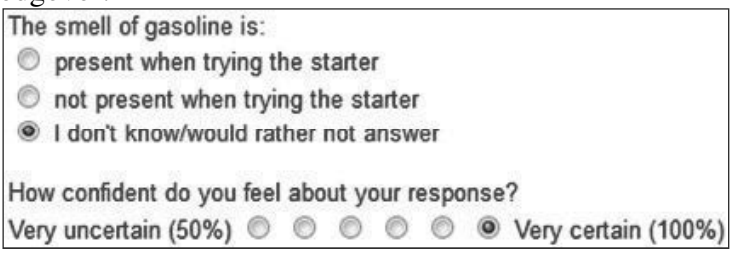
Iako je teorija faktora pouzdanosti zasnovana na nekim elementima teorije verovatnoće, faktori pouzdanosti su delom proizvoljne konstrukcije, koje mogu da poprime neke vrednosti koje protivreče uslovnim verovatnoćama. Pokazano je da je teorija faktora pouzdanosti aproksimativna u odnosu na klasičnu teoriju verovatnoće [19].

Praktična primena ove teorije dala je dobre rezultate, posebno kod ekspertnih sistema zasnovanih na pravilima. Mnogi softverski alati, kao što su ljuske ekspertnih sistema, imaju ugrađenu podršku za ovaj način zaključivanja u uslovima neizvesnosti.

Primer: Mehanizam zaključivanja ljuske ekspertnog sistema Expertise2Go

Ljuska ekspertnog sistema *Expertise2Go* je komercijalni alat za razvoj ekspertnih sistema, čiji se mehanizam zaključivanja isporučuje u obliku Java arhive [65]. Sistem ima i besplatnu verziju, koja se koristi u ovom materijalu.

Mehanizam zaključivanja ljuske ekspertnog sistema koristi produkciona pravila i mehanizam faktora pouzdanosti. Primer pravila ekspertnog sistema, koje koristi vrednost atributa [*smell of gasoline*] s faktorom pouzdanosti koji određuje korisnik, kao poverenje (50..100%) u sopstveni odgovor:



Predviđeni metodi kombinovanja faktora pouzdanosti premisa pravila iz baze znanja (npr. CF1 i CF2) u ljusci *Expertise2Go*, za neke oblike pravila, prikazani su u tabeli:

Metod	Definicija	Primena
Minimum	*min*(CF1, CF2)	IF a (CF1) AND b (CF2)
Maksimum	*max* (CF1, CF2)	IF a (CF1) OR b (CF2)
Prosek	(CF1+CF2)/2	IF a AND b (CF1); IF a OR b (CF2)
Zbir verovatnoća	CF1+CF2/100 x (100-CF1)	CFa=CF1; CFa=CF2
Proizvod	CF1 x CF2	IF a AND b (CF1) Then c (CF2)

2.6.2 Teorija fazi skupova

Prilikom rešavanja složenih problema, eksperti se uglavnom oslanjaju na iskustvo i zdrav razum [18]. Pri tome koriste nejasne i dvosmislene izraze, npr. "Transformator je *malo* preopterećen, ali se može dozvoliti takvo opterećenje *neko vreme*". Ovakve izjave razumeju eksperti iz iste oblasti, pošto imaju odgovarajuće predznanje, koje im omogućava ispravno razumevanje nejasne formulacije problema. Problem se javlja kada je ekspertsko znanje, koje koristi nejasne i dvosmislene izraze, potrebno predstaviti u računaru. Jedan od načina obezbeđuje teorija fazi skupova (*fuzzy sets*) i odgovarajuća fazi ili rasplinuta logika, koja se koristi za izgradnju fazi ekspertnih sistema.

Osnovna ideja je da tvrdnje o pripadnosti nekom skupu nisu isključivo istinite ili lažne (oznake 0/1 ili *da*/*ne*), već se uvodi stepen pripadnosti elementa nekom skupu, koji se izražava kao realni broj 0..1. To znači da isti objekt može istovremeno pripadati različitim skupovima, samo u različitom stepenu, tako da je pripadnost skupu nejasna, zamagljena (*fuzzy*), odnosno raspodeljena na više skupova istovremeno.

Fazi logika nije nejasna, već logika koja se koristi za opis nejasnog. Njena osnova je teorija fazi skupova, kojima se meri neodređenost. Fazi logika se zasniva na ideji da svi objekti imaju neka svojstva u određenom stepenu, npr. temperatura, visina, brzina, rastojanje i lepota su svojstva, koja se mogu izraziti brojem, prema nekoj skali, kao u iskazima "Petar je *veoma visok* čovek", "električni automobili *nisu veoma brzi*" i "Beograd je *divan* grad". Potreba za postojenjem takve skale često otežava određivanje pripadnosti nekoj klasi objekata, npr. određivanje uslova da je neko uzvišenje "brdo" ili "planina" [18].

Klasična ili Bulova logika koristi jasne razlike između pripadnika jedne klase i ostalih objekata, tako što za numerička svojstva koristi empirijske, diskutabilne granične vrednosti. Računajući da je radijus do 300km mali, a preko toga velik, može se reći "radijus kretanja električnih vozila je *mali*". Prema tom kriterijumu, vozilo koje pređe 301km (ili 300km i 1m) je vozilo s velikim radijusom kretanja. Ako se kao granica visokih ljudi uzme 180cm, čovek je visok ako ima 181cm, a nizak sa 179cm, što nije logično. Fazi logika je pokušaj da se u logiku ugradi ljudski način razmišljanja, tako što se modelira smisao reči, odlučivanje i zdrav razum (*common sense*) i omogući kreiranje inteligentnih sistema koji bolje modeliraju ljudski način razmišljanja [18].

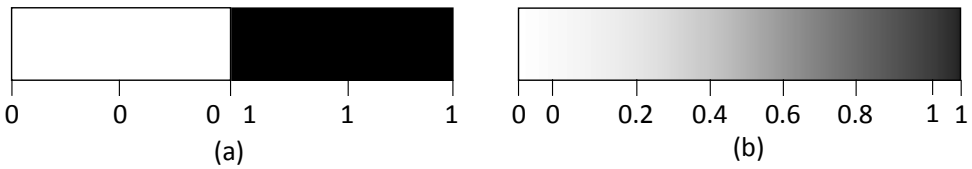
Prethodno je još 1930. godine poljski logičar i filozof Jan Lukašijevič[8] uveo višeznačnu logiku za matematičko predstavljanje neodređenosti (*fuzziness*) pomoću termina kao što su *visok*, *star* i *vruć*. Umesto dve logičke vrednosti istinitosti, 1 (*true*) i 0 (*false*), uveo je logiku u kojoj se vrednost istinitosti označava realnim brojem 0..1, koji označava *mogućnost* da je iskaz istinit ili lažan. Npr. mogućnost da je čovek visine 181cm zaista visok bi mogla biti 0.86, tako da je čovek te visine *verovatno* visok. Svoju teoriju je nazvao teorija mogućnosti (*possibility theory*).

Jednostavne fazi skupove definisao je 1937. godine filozof Maks Blek i opisao osnovne ideje operacija nad njima.

Lotfi Zade, profesor i dekan departmana za elektrotehniku na kalifornijskom univerzitetu Berkli je 1965. godine (ponovo) otkrio, istražio, proširio i dalje promovisao ovakav pristup tretmanu neodređenosti. Teoriju mogućnosti je proširio u formalni sistem matematičke logike i uveo novi koncept upotrebe izraza iz prirodnog jezika.

[8] Jan Lukasiewicz (1878-1956), poljski logičar i matematičar

Fazi logika je višeznačna (*multi-valued*) i koristi pojam stepeni pripadnosti skupu (*degrees of membership*) i stepen istinitosti iskaza (*degrees of truth*), čija vrednost je kontinualna u intervalu od 0 (poptuno lažno) do 1 (potpuno istinito), kao na Sl. 20. Na taj način, klasična Bulova logika je specijalni slučaj višeznačne fazi logike.



Sl. 19. Vrednosti istinitosti u (a) Bulovoj logici i (b) višeznačnoj logici

Fazi logika je skup matematičkih principa predstavljanja znanja na osnovu stepena pripadnosti skupu vrednosti, umesto na osnovu isključive pripadnosti, kao u klasičnoj binarnoj logici [18].

Fazi logika je mali deo teorije fazi skupova. Skup je osnovni pojam u matematici, povezan s ljudskim načinom izražavanja. Kad se kaže *automobil*, obično se misli na skup automobila, dok se pojedinačni automobil obično dodatno označava (*ovaj automobil*).

Ako je X klasičan skup i x njegov element, u klasičnoj logici element x ili pripada skupu X ($x \in X$) ili mu ne pripada ($x \notin X$). Svaki objekt koji pripada skupu označava se vrednošću 1, a svi ostali vrednošću 0. Uz ovaj princip su od antičkih vremena vezani logički paradoksi, jer se u okviru klasične logike mogu formirati protivrečni iskazi, o čijoj istinitosti se ne može odlučiti.

Poznat je sofizam iz vremena Pitagorine filozofske škole (400. pne), koji postavlja pitanje da li filozof Epimenid s Krita govori istinu kad kaže da *svi Krićani uvek lažu*. Klasična logika ne može da odgovori na ovo pitanje, jer ta tvrdnja sadrži kontradikciju. Fazi logika daje odgovor: filozof govori istinu i ne govori istinu. Iz novijeg Raselovog paradoksa, koji demonstrira probleme naivne teorije skupova, izveden je paradoks berberina [61], gde se opisuje situacija kada u selu postoji samo jedan berberin, koji brije one i samo one koji se ne briju sami. Postavlje se pitanje: ko brije berberina? Klasična logika opet ne može da odgovori na pitanje, jer tvrdnja sadrži kontradikciju. Fazi logika daje odgovor da berber brije i ne brije samog sebe. Prema fazi logici, oba moguća odgovora su delimično istinita [18].

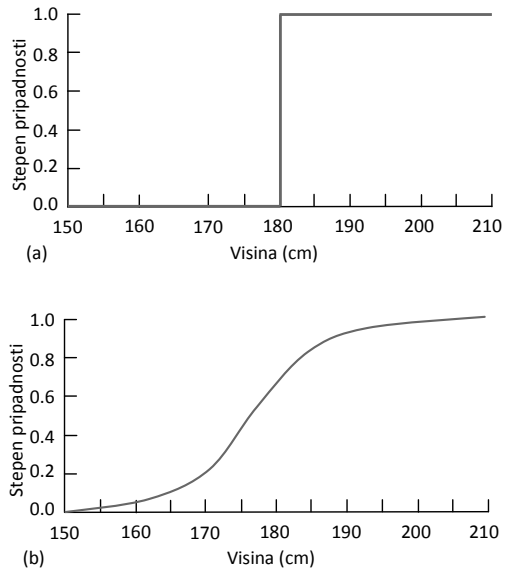
Pojam *visok čovek* je u fazi logici fazi skup, koji obuhvata sve ljude s različitim stepenom pripadnosti, koji zavisi od njihove visine, Tabela 1 [18]. Vidi se da je binarni skup, koji odgovara na pitanje "da li je čovek visok", formiran oko granice od 180cm. Fazi skup odgovara na pitanje "koliko je čovek visok" i odgovor daje u obliku stepena pripadnosti skupu visokih iz Tabele 1. Stepen

pripadnosti skupu visokih brzo opada za osobe F-J. Za određivanje stepena pripadnosti fazi skupovi koriste funkciju pripadnosti skupu, kao na Sl. 20.

Tabela 1. Stepen pripadnosti skupu visok čovek

Ime	Visina (cm)	Stepen pripadnosti	
		Binarno	Fazi
A	208	1	1.00
B	205	1	1.00
C	198	1	0.98
D	181	1	0.82
E	179	0	0.78
F	172	0	0.24
G	167	0	0.15
H	158	0	0.06
I	155	0	0.01
J	152	0	0.00

Na Sl. 20. prikazan je kompletan pogled na razmatrani problem i skup mogućih vrednosti određene promenljive, u ovom slučaju visine čoveka [18]. Skup visokih ljudi sadrži sve ljude, ali može biti i deo šireg konteksta visine sisara ili životinja uopšte. Na sličan način se zadaju fazi skupovi "veoma nizak čovek", "prosečan čovek" i "veoma visok čovek".



Sl. 20. Funkcija pripadnosti skupu visok čovek za (a) klasični i (b) fazi skup

U razmatranom univerzumu može se definisati više povezanih fazi skupova. Npr. visina čoveka se može posmatrati preko tri fazi skupa: *nizak*, *prosečan* i *visok čovek*. Čovek visine 184cm će tako pripadati različitim skupovima, skupu prosečnih sa stepenom pripadnosti 0,1, a skupu visokih sa 0,4.

Fazi skup se može definisati pomoću nekog skupa i fazi granica skupa. Ako je X razmatrani univerzum, čiji su elementi označeni sa x, u klasičnoj teoriji skupova se neki skup A određuje pomoću *karakteristične funkcije* $f_A(x)$:

$$f_A(x) : X \rightarrow 0, 1$$

definisane kao:

$$f_A(x) = \begin{cases} 1, za\ x \epsilon A \\ 0, za\ x \notin A \end{cases}$$

Ovaj skup preslikava univerzum X u skup od dva elementa. Fazi skup A univerzuma X se određuje pomoću *funkcije pripadnosti* $\mu_A(x)$ skupu A:

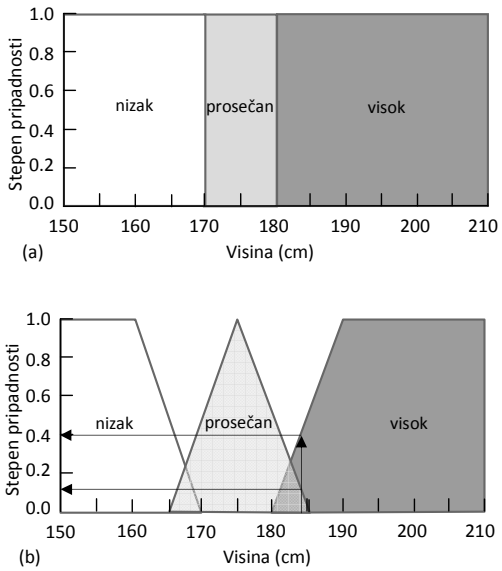
$$\mu_A(x) : X \rightarrow [0, 1]$$

definisane kao:

$\mu_A(x) = 1$ ako je x potpuno u skupu A,
$\mu_A(x) = 0$ ako x nije element A,
$0 < \mu_A(x) < 1$ ako je x delimično u skupu A.

Prvi korak definisanja fazi skupa je određivanje funkcije pripadnosti μ, obično na osnovu ekspertskog znanja jednog ili više eksperata ili metodima mašinskog učenja na osnovu raspoloživih podataka.

Tipične funkcije koje se koriste kao funkcije pripadnosti su sigmoid, Gausova kriva i pi-funkcija, ali se u praksi češće koristi kombinacija linearnih funkcija, koja se efikasnije izračunava, Sl. 21 [18].



Sl. 21. Skupovi nizak, prosečan i visok čovek za (a) klasične i (b) fazi skupove

Fazi promenljive, koje se koriste u fazi pravilima ekspertnog sistema, nazivaju se *lingvističke promenljive*. Npr. promenljiva *Speed* može da ima vrednosti 0..220 km/h i fazi podskupove *very slow*, *slow*, *medium*, *fast* i *very fast*, koji predstavljaju njene lingvističke vrednosti.

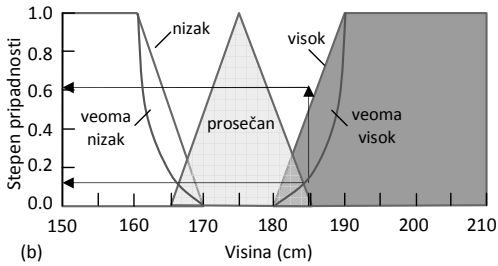
Fazi pravilo može da ima oblik:

IF *Speed* is *slow*
THEN *Stopping_distance* is *short*

Promenljive mogu da imaju fazi kvalifikatore (*hedges*), koji lingvistički predstavljaju priloge i modifikuju oblik fazi skupova. Fazi kvalifikatori mogu biti:

- opšte namene, kao što su *very*, *quite* i *extremely*;
- vrednosti istinitosti, kao što su *quite true* i *mostly false*;
- verovatnoće, kao što su *likely* i *not very likely*;
- kvantifikatori, kao što su *most*, *several* i *few*;
- opis mogućnosti, kao što su *almost impossible* i *quite possible*.

Kvantifikatori istovremeno predstavljaju i operacije, jer definišu nove fazi skupove, uže (*very tall men*) ili šire od početnih (*more or less tall men*), kao i fazi intervale, Sl. 22 [18].



Sl. 22. Fazi skupovi s fazi kvalifikatorom veoma

Matematički efekti primene operacija definisanih nekim od fazi kvalifikatora su, npr.

- *very* $\mu_A^{very}(x) = [\mu_A(x)]^2$
- *extremely* $\mu_A^{extremely}(x) = [\mu_A(x)]^3$
- *more or less* $\mu_A^{more\ or\ less}(x) = \sqrt{\mu_A(x)}$
- *indeed* $\mu_A^{indeed}(x) = \begin{cases} 2[\mu_A(x)]^2 & za\ 0 \le \mu A(x) \le 0{,}5 \\ 1 - 2[\mu_A(x)]^2 & \text{za } 0{,}5 < \mu A(x) \le 1 \end{cases}$

Osnovne operacije nad fazi skupovima su komplement skupa, podskup, presek i unija dva skupa, defisane kao:

- komplement $\mu_{\neg A}(x) = 1 - \mu_A(x)$
- podskup $\mu_{A \subseteq B}(x) = \mu_A(x) \leq \mu_B(x) = \mu_A(x) \subseteq \mu_B(x), x \in X$
- presek $\mu_{A \cap B}(x) = min[\mu_A(x), \mu_B(x)] = \mu_A(x) \cap \mu_B(x), x \in X$
- unija $\mu_{A \cup B}(x) = max[\mu_A(x), \mu_B(x)] = \mu_A(x) \cup \mu_B(x), x \in X$

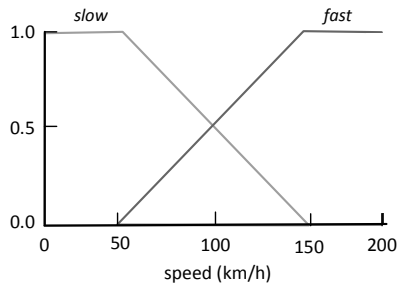
Za ove operacije važe pravila komutacije, asocijacije, distribucije, idempotentnost, tranzitivnost i De Morganovi zakoni, kao i za operacije nad običnim skupovima. Korišćenjem ovih osobina, lako se formiraju fazi skupovi koji predstavljaju lingvistički opis nepreciznih iskaza prirodnog jezika.

Na Sl 23. je primer kako se umesto klasičnih produkcionih pravila i relacionih izraza (a) koriste fazi pravila s lingvističkim promenljivima (b), koja su bliža ljudskom načinu mišljenja i izražavanja.

Rule: 1	Rule: 1
IF Speed > 100	IF Speed is *fast*
THEN Stopping_distance is long	THEN Stopping_distance is *long*
Rule: 2	Rule: 2
IF Speed < 40	IF Speed is *slow*
THEN Stopping_distance is short	THEN Stopping_distance is *short*
(a) klasična pravila	(b) fazi pravila

Sl. 23. Primer klasičnih produkcionih pravila i ekvivalentnih fazi pravila

Promenljiva *Speed* u fazi pravilu je tzv. lingvistička promenljiva, čija je vrednost npr. iz intervala 0..250 km/h. Uključuje fazi skup {*slow*, *fast*}, u kome je npr. brzina od 100 km/h brzina koja je 0.5 *slow* i 0.5 *fast*, Sl. 24.



Sl. 24. Fazi skup speed

Fazi zaključivanje sastoji se od evaluacije preduslova pravila (IF deo) i implikacije, odnosno primene rezultata zaključka pravila (THEN deo). U klasičnom pravilu, ako je preduslov istinit, istinit je i zaključak. U fazi sistemima, gde je preduslov pravila fazi iskaz, sva pravila se primenjuju u

izvesnom stepenu, odnosno primenjuju se delimično. Ako je preduslov istinit u određenom stepenu i zaključak ima taj stepen istinitosti.

Fazi pravila mogu imati više preduslova i više zaključaka. Konačni zaključak se dobija agregacijom svih izlaznih fazi skupova u jedan skup, koji se zatim transformiše u jedan broj (*defuzzification*).

Postoje dve vrste fazi zaključivanja koji se mogu realizovati u fazi logici, Mamdani i Sugeno metod, koji se donekle razlikuju u načinu na koji se određuju izlazi. Mamdani metod se široko koristi za opis ekspertskog znanja, omogućava da se opiše ekspertiza na čoveku razumljiv način, ali je računski zahtevniji. Sugeno metod je računski manje zahtevan i koristi se u optimizaciji, posebno za probleme upravljanja dinamičkim i nelinearnim sistemima.

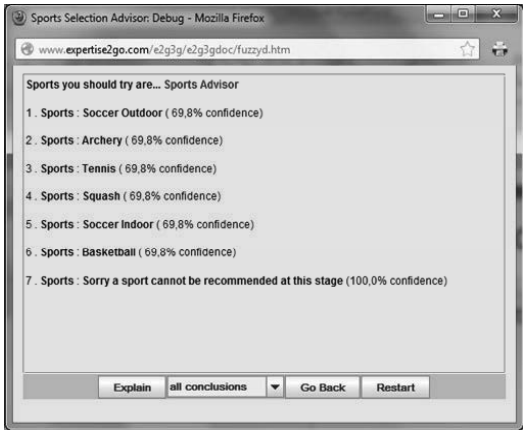
Izgradnja fazi ekspertnih sistema se sastoji od:

1. Specifikacije problema i definisanja linvističkih promenljivih;
2. Određivanja fazi skupova;
3. Otkrivanja i izgradnje fazi pravila;
4. Kodiranja fazi skupova, fazi pravila i procedura fazi zaključivanja u ekspertnom sistemu;
5. Evaluacije i podešavanja sistema.

Ilustracija: Izbor sportske discipline na osnovu fuzzy logike

Primer izbora najpogodnije sportske discipline na osnovu fazi logike. Sistem *Sports Advisor* [65] postavlja niz pitanja o mogućnostima kandidata, a zatim daje višeznačnu, fazi odluku, u kojoj se svakom elementu skupa mogućih odluka dodeljuje stepen pripadnosti.

Sistem je samo demonstracioni primer, koji je implementiran korišćenjem ljuske ekspertnog sistema Expertise2Go [65].



Na slici je prikazan višeznačni rezultat, koji se dobije kad se na sva postavljena pitanja sistema odgovori *ne znam* (*unknown*), tako da nema dovoljno elemenata da se prednost da bilo kojoj konkretnoj sportskoj disciplini.

Pitanja za ponavljanje

1. Čime se bavi teorija odlučivanja?
2. Koja je razlika između normativnih i deskriptivnih teorija odlučivanja?
3. Koja su osnovna svojstva odluka koje razmatra teorija odlučivanja?
4. Koja je specifičnost teorije igara u odnosu na teorije odlučivanja?
5. Navedite svojstva odluka.
6. Šta je paradoks izbora?
7. Šta demonstrira tzv. Paskalova opklada?
8. Šta je greška tipa I u statističkoj teoriji odlučivanja?
9. Šta je greška tipa II u statističkoj teoriji odlučivanja?
10. Koje su prednosti alternativnih teorija odlučivanja?
11. Na osnovu čega se rangiraju hipoteze metodom faktora pouzdanosti (*Certainity Factor*, CF)?
12. Kako se izračunava mera pouzdanosti (*Certainity Factor*, CF)?
13. Kako se računa faktor pouzdanosti (*Certainity Factor*, CF) konjukcije dveju tvrdnji?
14. Kako se računa faktor pouzdanosti (*Certainity Factor*, CF) disjunkcije dveju tvrdnji?
15. Navedite neka proširenja klasične matematičke logike.

3 Modeliranje i analiza procesa donošenja odluka

1. Modeli i modeliranje
2. Matematički modeli za podršku odlučivanju
3. Softverska podrška modeliranju
4. Primeri rešavanja strukturiranih problema

U ovom poglavlju se ukratko izlažu osnovni metodi modeliranja različitih aspekata poslovnog odlučivanja. Objašnjavaju se osnovni matematički modeli i drugi metodi i alati modeliranja, kao i odgovarajuća softverska podrška i primeri sistema i njihova upotreba.

3.1 Modeli i modeliranje

Model je pojednostavljena predstava ili apstrakcija stvarnosti, čija je osnovna namena smanjenje kompleksnosti sistema koji se razmatra, čime se omogućava usmeravanje pažnje samo na bitne elemente sistema. Modeliranje procesa donošenja odluke olakšava podršku odlučivanju i pojednostavljuje sam proces odlučivanja.

Primer: Modeliranje u sportu

Modeliranje odlučivanja u sportu koriste treneri, menadžeri i posrednici. Npr. prediktivni analitički modeli se koriste u planiranju karijere sportista, transfera igrača i sastavljanju timova [1].

U analizi se koristi veliki broj faktora (promenljivih) koji utiču na rezultat, posebno statistički podaci ili istorijat rezultata za pojedince i timove.

Prvi sistemi za podršku odlučivanju u profesionalnom sportu razvijeni su za bejzbol i košarku [1]:

- www.sabr.org - razvijene složene metrike uspešnosti sportskih bejzbol timova;
- www.hoopmasters.com - sistem za podršku odlučivanju košarkaških trenera, koji sadrži podatke o više od 500.000 igrača amaterskih selekcija, njihove karakteristike, video snimke i predloge za dalje školovanje.

Primena kvantitativnih modela na analizu i predviđanje počela je u okviru bejzbol udruženja SABR (*Society for American Baseball Reasearch*). Korišćeni su statistički modeli za analizu uticaja brojnih empirijskih parametara (popularno *sabermetrics* [89]).

Primer je statistika *Runs Created* (Bill James, 1970) za predviđanje uspešnosti timova na osnovu ranijih rezultata [89]:

G	AB	H	2B	3B	HR	BB	K	AVG
161	5.517	1.451	234	22	214	604	908	0,263

Predviđao se broj napada po utakmici u toku godine na osnovu statistike bacanja (*batting line*) po formuli (modelu)

$Runs\ Created = TB \cdot (H+BB)/(AB+BB) = 826$

gde su:

TB - *Total Bases* $= 1 \cdot B + 2 \cdot 2B + 3 \cdot 3B + 4 \cdot HR = H + 2B + 2 \cdot 3B + 3 \cdot HR$
H - *Hits*
BB - *Bases on Balls* (*Walks*)
AB - *At-Bats* (*Appearances*)

Rezultati modeliranja služili su kao podrška objektivnijem odlučivanju o sastavu timova, strategiji igre, transferu igrača itd.

Proces modeliranja i analize može se opisati kao niz aktivnosti, koje obuhvataju identifikaciju problema, identifikaciju promenljivih, prediktivnu analitiku/prognostiku, istovremenu upotrebu više modela, izbor kategorije modela, upravljanje modelima i modeliranje zasnovano na znanju [1].

U Tabeli 2 [1] prikazane su kategorije modela, koje se najviše koriste u sistemima za podršku odlučivanju:

Tabela 2 Kategorije modela u sistemima za podršku odlučivanju

Vrsta		Cilj	Tipične tehnike
Optimizacija	mali broj varijanti	najbolja alternativa od malog broja razmatranih	tebele odlučivanja, stabla odlučivanja
	veliki broj varijanti	najbolja alternativa od velikog broja razmatranih	*linearno programiranje*, *mrežni modeli*
	analitičke formule	najbolja alternativa prema formuli	upravljanje zalihama
Simulacije		eksperimentalno ispitivanje i izbor dovoljno dobrog rešenja	više vrsta *simulacije*
Heuristike		izbor dovoljno dobrog rešenja na osnovu pravila	*ekspertni sistemi*
Prediktivni modeli		predviđanje budućnosti prema scenariju	prognostički modeli, Markovljevi modeli
Ostali modeli		*what-if* analiza (formule)	finansijski modeli

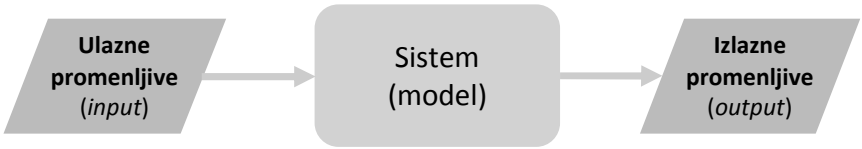
Prema stepenu apstrakcije, modeli mogu biti skalirani, analogni, mentalni i matematički. Matematički ili kvantitativni modeli su od posebnog značaja, jer omogućavaju rangiranje alternativa i izbor optimalnog rešenja.

3.2 Matematički modeli za podršku odlučivanju

Matematički modeli za podršku odlučivanju mogu biti *statički* i *dinamički*. Statički modeli su fiksni i odnose se na jedan presek situacije, dok se dinamički modeli menjaju tokom vremena.

Modeli omogućavaju modeliranje neizvesnosti i rizika. Znanje o nekom problemu umanjuje neizvesnost, dok postojanje neizvesnosti stvara rizik. Probabilističko odlučivanje je odlučivanje uz postojanje rizika, u kome su mogući svi ishodi. Analiza rizika (*risk analysis*) izračunava vrednost alternativa i bira onu alternativu koja ima najveću očekivanu vrednost.

Opšta struktura matematičkih ili kvantitativnih modela je prikazana na Sl. 26 [1].



Sl. 25. Opšta struktura kvantitativnih modela i rešavanje problema

Poznati matematički modeli su tabele i stabla odlučivanja, sistemi jednačina, modeli teorije igara, klasična matematička logika i njena proširenja. U ovoj tački ukratko će se objasniti matematički modeli, koji su posebno značajni za podršku odlučivanju:

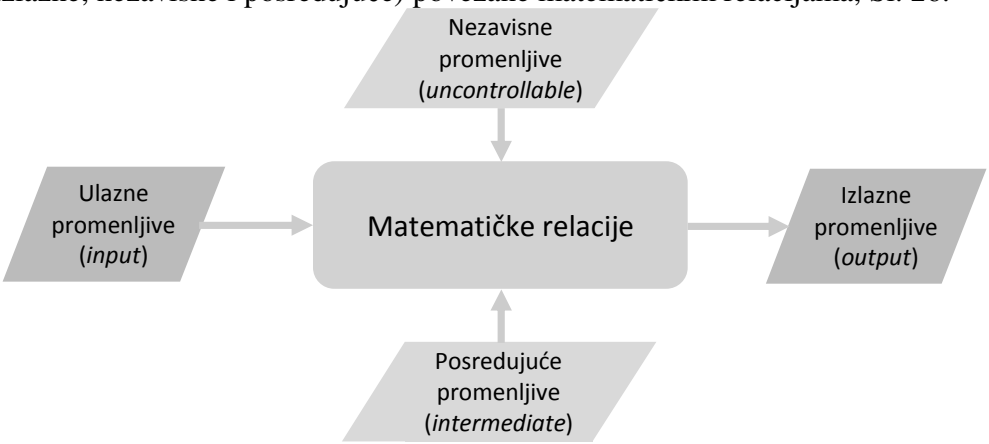
1. Optimizacija i linearno programiranje
2. Tabele i stabla odlučivanja
3. Modeli redova čekanja
4. Opšti metodi rešavanja problema
5. Simulacije

3.2.1 Optimizacija i linearno programiranje

Operaciona istraživanja su naučna disciplina koja istražuje najbolje puteve rešavanja problema menadžmenta složenih organizacija. Sam naziv povezan je s rešavanjem problema planiranja proizvodnje početkom drugog svetskog rata, kada je veliki broj naučnika angažovan u rešavanju različitih problema menadžmenta na svim nivoima, naročito problema *vojnih operacija*.

Opšti problem u operacionim istraživanjima je distribucija ograničenih resursa između konkuretnih aktivnosti na najbolji mogući (optimalan) način. Koriste se

kvantitativni modeli, koji obuhvataju četiri kategorije promenljivih (ulazne, izlazne, nezavisne i posredujuće) povezane matematičkim relacijama, Sl. 26.



Sl. 26. Struktura kvantitativnih modela i rešavanje problema

Opšta procedura rešavanja problema u operacionim istraživanjima obuhvata:

1. Definisanje problema i prikupljanje podataka,
2. Formulisanje matematičkog modela problema,
3. Razvoj računarski zasnovanog rešenja problema na osnovu modela,
4. Testiranje i eventualnu korekciju modela,
5. Izradu aplikacije,
6. Implementaciju.

Optimizacija je proces pronalaženja najboljeg (optimalnog) rešenja nekog problema. Ako postoji matematički izraz kriterijuma izbora najboljeg rešenja, optimizacija je pronalaženje *maksimuma* ili *minimuma* funkcije kriterijuma.

Linearno programiranje je vrsta optimizacije, gde su kriterijumska funkcija i ograničenja linearne funkcije.

Analiza osetljivosti (*sensitivity analysis*) je ispitivanje uticaja malih promena ulaznih vrednosti na izlaz sistema.

Analiza varijanti (*What-if analysis*) ispituje efekte promene vrednosti ulaznih promenljivih na izlaz, odnosno ciljnu funkciju.

Analiza cilja (*Goal-seek analysis*) pronalazi moguće ulazne vrednosti za izabrane vrednosti izlaza, odnosno ciljne funkcije.

Linearno programiranje

Linearno programiranje za opis ovog problema koristi matematički model, koji se sastoji samo od linearnih funkcija, odnosno polinoma prvog reda, čija geometrijska interpretacija su prave. Naziv *programiranje* nastao je pre pojave računara kao sinonim za pažljivo *planiranje* aktivnosti, koje će dovesti do

postizanja postavljenog cilja na najbolji od mogućih načina definisanih modelom.

Osim alokacije resursa i mnogi drugi problemi optimizacije mogu se rešiti pomoću modela koji se zasniva na linearnim jednačinama, čije optimalno rešenje se pronalazi računski efikasnim metodima kao što je *simpleks* metod (*simplex method*).

Počeci lineranog programiranja mogu se pronaći početkom 19. veka u radovima Furijea, koji je izučavao linearne nejednakosti. Više matematičara dokazalo je specijalne slučajeve teoreme o dualnosti (*duality theorem*) [21]. Ruski matematičar Kantorovič je 1939. godine ukazao na praktični značaj nekih klasa problema linearnog programiranja, kao i algoritme za njihovo rešavanje, ali njegov rad nije zapažen i ostao je nepoznat na Zapadu.

Američki matematičar Dancig[9] je 1947. godine razvio *simpleks metod* za rešavanje problema linearnog programiranja i primenio ga za planiranje u američkom vazduhoplovstvu. U isto vreme, 1963. godine, ekonomista Kupermans pokazao je da je linearno programiranje adekvatan model za analizu klasičnih ekonomskih teorija.

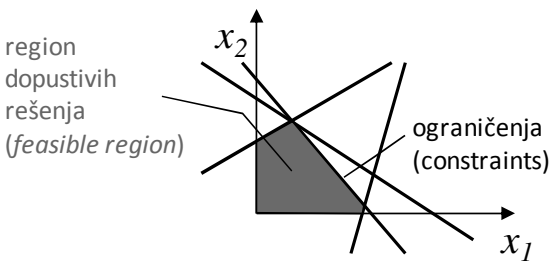
Godine 1975. švedska akademija nauka je dodelila Nobelovu nagradu za ekonomiju Kantoroviču i Kupermansu, za njihov doprinos teoriji optimalne alokacije resursa. Dancigov matematički doprinos ostao je bez Nobelove nagrade, jer ona za matematiku nije ustanovljena [21].

Sistemi linearnih jednačina

Matematički model problema linearnog programiranja su linearne jednačine, odnosno sistemi linearnih jednačina.

Standardni matematički oblik traži maksimum ciljne funkcije $F(x_1,\ldots,x_n)= c_1x_1+c_2x_2+ \ldots + c_nx_n$, uz zadana ograničenja:

$$a_{11}x_1+a_{12}x_2+ \ldots + a_{1n}x_n \leq 0$$
$$a_{21}x_1+a_{22}x_2+ \ldots + a_{2n}x_n \leq 0$$
$$\ldots$$
$$a_{m1}x_1+a_{m2}x_2+ \ldots + a_{mn}x_n \leq 0$$



Traži se optimalno rešenje, ako postoji, koje zadovoljava ograničenja ili uslove. Ograničenja određuju region dopustivih rešenja (*feasible region*), koji je u opštem slučaju *n*-dimenzionalni *politop* (dopustivi politop), geometrijsko telo sa ravnim stranicama u

[9] George B. Dantzig (1914-2005), američki matematičar, značajan po velikom doprinosu operacionim istraživanjima (simplex algoritam), računarskim naukama, ekonomiji i statistici

proizvoljnom broju dimenzija. Dvodimenzionalni politop je poligon, trodimenzionalni poliedar, itd.

Primer: Elementarni primer planiranja proizvodnje (manuelno rešavanje)

Proizvođač stakla proizvodi kvalitetno staklo, vrata i prozore u tri različta pogona: *pogon 1* za proizvodnju aluminijumskih okvira, *pogon 2* za proizvodnju drvenih okvira i *pogon 3* za proizvodnju stakla i sklapanje vrata i prozora.

Zbog pada profita, odlučeno je da se proizvodnja reorganizuje, tako da se neprofitabilni proizvodi zamene sa dva nova proizvoda:
1. staklenim vratima sa aluminijumskim ramom i
2. dvokrilnim prozorom sa drvenim ramom.

Proizvodnja vrata zahteva kapacitete pogona 1 i 3, a proizvodnja prozora pogone 2 i 3. Razmatra se nedeljna proizvodnja serija od 20 komada svakog od prozvoda.

Definicija problema optimizacije je: *koje količine svakog od proizvoda treba proizvesti da se maksimizuje ukupni profit, uzimajući u obzir raspoložive kapacitete tri proizvodna pogona*?

Osnovni koraci manuelnog rešavanja su:

1. *Korak 1*: prikupljanje podataka (kapacitet pogona u satima/nedeljno, vreme izrade i profit po seriji)

Pogon	**Vreme izrade** (sati/seriji)		**Ukupni kapacitet** (sati/nedeljno)
	proizvod		
	1	**2**	
1	1	0	4
2	0	2	12
3	3	2	18
Profit po seriji	3.000	5.000	

2. *Korak 2*: formulisanje matematičkog modela

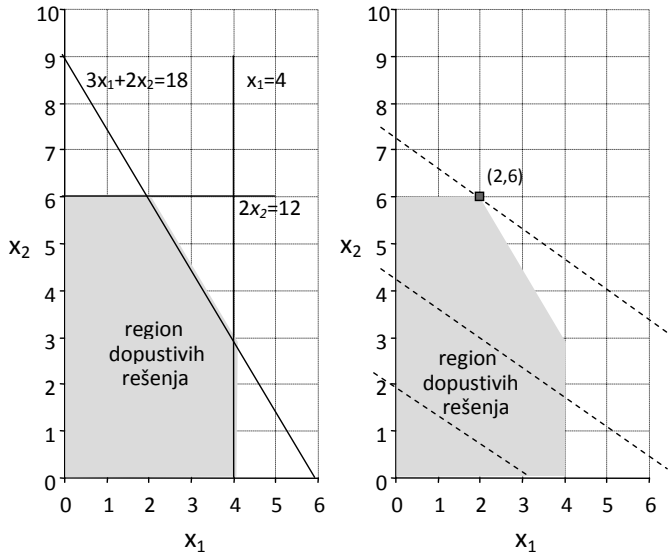
   Maksimizovati funkciju profita $F=3.000x_1+5.000x_2$

   tako da važe ograničenja
   $x_1 \leq 4$
   $2x_2 \leq 12$
   $3x_1+2x_2 \leq 18$, kao i $x_1 \geq 0$ i $x_2 \geq 0$

3. *Korak 3*: grafičko i analitičko rešenje

   Poslednji korak je izbor tačke iz regiona dopustivih rešenja (*feasible region*) koja maksimizuje vrednost funkcije cilja $F=3000x_1+5000x_2$

Eksperimentalni, geometrijski pristup je iscrtavanje familije paralelnih linija, koje sadrže bar jednu tačku iz regiona dopustivih rešenja, te usvajanje one koja daje najveću vrednost *F*, Sl. 27.



Sl. 27. Grafičko rešavanje problema linearnog programiranja

Linija koja sadrži samo tačku regiona (2,6) daje najveću vrednost i predstavlja *optimalno rešenje* problema.

Region dopustivih rešenja u ovom primeru je poligon, a optimalno rešenje je teme pologona.

Računarsko rešenje problema linearnog programiranja

Fundamentalna teorema linearnog programiranja definiše skup u kome se nalazi optimalno rešenje problema linearnog programiranja [7], [8].

Teorema: Ako problem linearnog programiranja ima rešenje, ono se nalazi u temenima dopustivog politopa.

Efikasan metod rešavanja sistema linearnih jednačina nazvan *simpleks metod* (*symplex method*) pronašao je 1949, a publikovao 1963. godine matematičar Dancig. Rešavanje problema ovim metodom uz pomoć računara ilustrovaće se na sledećem primeru [8].

Problem: Proizvodnja hidromasažnih kada

Proizvođač kada proizvodi dva modela hidromasažnih kada i treba da odluči koliko komada svakog modela treba proizvesti u narednom periodu.

Hidromasažna kada se dobija kada se na školjku od stakloplastike dodaju pumpa i odgovarajuće cevi. Školjka se nabavlja kod lokalnog dobavljača, koji može da isporuči njihov dovoljan broj u narednom periodu, ali u narednom periodu proizvođač može da obezbedi samo 200 pumpi.

Za kompletiranje prvog modela kade proizvođaču je potrebno 9 sati rada i 4 metra cevi, a za drugi model 6 sati rada i 6 metara cevi. Proizvođač ukupno raspolaže sa 1.600 radnih sati i 1.020m cevi za ugradnju.

Profit po prodatoj kadi je za prvi model 30.000 din, a za drugi 25.000 din. Uz pretpostavku da će se sve proizvedene kade prodati, koliko treba u narednom periodu proizvesti svakog od dva modela kada da bi se ostvario *najveći mogući* profit?

Matematičak formulacija modela linearnog programiranja

Opšti pristup formulisanju ispravnog modela problema linearnog programiranja obuhvata sledeće standardne korake:

1. *Razumevanje problema*

   Detaljno sagledavanje i razumevanje svih aspekata konkretnog problema je preduslov za dobru formulaciju problema i ispravan model linearnog programiranja.

   U konkretnom primeru problem je da se odredi koliko komada hidromasažnih kada svakog od modela treba proizvesti u narednom periodu, tako da se maksimizuje profit i ne utroši više od 200 pumpi, 1.600 radnih sati i 1.020 metara cevi.

2. *Identifikacija promenljivih odlučivanja (decision variables)*

   Promenljive odlučivanja su veličine o čijoj vrednosti treba doneti odluku, kako bi se rešio postavljeni problem. Obično se kraće označavaju matematičkim oznakama $x_1, x_2, \dots, x_n$.

   U konkretnom problemu to su količine hidromasažnih kada svakog od dva modela koje treba proizvesti u narednom periodu i koje je pogodno označiti sa x_1 i x_2.

3. *Iskazivanje ciljne funkcije kao linearne kombinacije promenljivih odlučivanja*

   Nakon identifikacije promenljivih odlučivanja, sledeći korak je kreiranje ciljne funkcije modela. Ciljna funkcija matematički izražava vezu promenljivih odlučivanja u modelu, a rešenje problema su vrednosti promenljivih odlučivanja koje daju njenu maksimalnu ili minimalnu vrednost.

U konkretnom primeru, ciljna funkcija izražava ukupan profit, koji se dobija sabiranjem profita od prodatih x_1 kada jednog i x_2 kada drugog modela $F = 30.000x_1 + 25.000x_2$.

Pošto nas zanima najveći mogući profit, izabraćemo takve vrednosti za x_1 i x_2 da vrednost ciljne funkcije F bude *maksimalna*.

4. *Iskazivanje svakog ograničenja kao kao linearne kombinacije promenljivih odlučivanja*

Vrednosti promenljivih nisu proizvoljne, već zavise od uslova konkretnog problema i iskazuju se kao ograničenja.

U primeru su to tri ograničenja: broj pumpi, broj radnih sati i ukupna dužina cevi za ugradnju. Ukupni broj pumpi određuje ukupni broj kada oba modela koji se može proizvesti, pošto se ugrađuje po jedna pumpa u oba modela kade:

$$1x_1 + 1x_2 \leq 200$$

To znači da veći broj proizvedenih kada jednog modela znači manji broj kada drugog modela. Ukupni broj od 1.600 radnih sati se takođe deli, po 9 sati za kadu prvog i 6 sati za proizvodnju kade drugog modela:

$$9x_1 + 6x_2 \leq 1.600$$

Treće ograničenje je dužina cevi za ugradnju, po 4 metra za prvi i 6 metara za drugi model kade, tako da se ukupno ne utroši više od raspoloživih 1.020 metara:

$$4x_1 + 6x_2 \leq 1.020$$

5. *Određivanje globalne gornje i donje granice vrednosti promenljivih odlučivanja*

Matematički se definiše skup mogućih vrednosti koje mogu da poprime promenljive odlučivanja. To je istovremeno i dodatno ograničenje u problemu.

U problemu proizvodnje kada jasno je da promenljive x_1 i x_2 mogu imati samo pozitivne vrednosti, jer se ne može proizvesti negativni broj kada, pa se uvode ograničenja $x_1 \geq 0$ i $x_2 \geq 0$. Uslovi nenegativnosti su česti u problemima linearnog programiranja.

Kompletna specifikacija modela odlučivanja je:

$$
\begin{aligned}
max \quad F = 30.000x_1 + 25.000x_2 \quad & profit \\
1x_1 + 1x_2 \leq 200 \quad & ograničen\ broj\ pumpi \\
9x_1 + 6x_2 \leq 1.600 \quad & ograničenje\ radnih\ sati \\
4x_1 + 6x_2 \leq 1.020 \quad & ograničenje\ dužine\ cevi\ (m) \\
x_1, x_2 \geq 0 \quad & uslov\ nenegativnosti
\end{aligned}
$$

Promenljive x_1 and x_2 čiju vrednost treba odrediti predstavljaju broj proizvedenih kada dvaju modela. Vrednosti za koje će se proizvođač odlučiti moraju da maksimizuju vrednost funkcije F i istovremeno zadovolje sve postavljene uslove.

Računarski model problema linearnog programiranja

Pre upotrebe dodatka Mucrosoft Excel *Solver* za rešavanje problema optimizacije metodom linearnog programiranja, treba pažljivo pročitati i razumeti problem koji se rešava i pregledno organizovati *Excel* tabelu.

Matematički model problema prevodi tekstualno zadani problem u skup jednačina i nejednačina, kojima se eksplicitno iskazuju tražene vrednosti i zadana ograničenja. Model se sastoji iz tri osnovna dela, koja treba uočiti i posebno prikazati:

1. promenljive (*decision variables*),
2. ciljna funkcija (*objective function*),
3. ograničenja (*constraints*).

Promenljive odlučivanja su one promenljive kojima se dodeljuju nepoznate vrednosti, koje predstavljaju rešenje problema. Načelno mogu biti negativne ili nenegativne. Promenljive se koriste za predstavljanje svih ostalih relacija u modelu, uključujući ciljnu funkciju i ograničenja.

Ciljna funkcija je jednačina koja iskazuje cilj modela. U zavisnosti od prirode problema, može se maksimizovati ili minimizovati. U najvećem broju primena se maksimizuje profit ili minimizuju troškovi.

Ograničenja postoje u svakom realnom problemu. To mogu biti ograničene količine resursa, rada ili zahtevi potražnje. Predstavljaju se u obliku jednačina i nejednačina, koje koriste deklarisane promenljive problema.

Predstavljanje problema u Excel tabeli

Svaki deo problema se jasno i pregledno definiše u Excel tabeli. Solver će interpretirati model na osnovu definisanih promenljivih, ciljne funkcije i ograničenja iz *Excel* tabele. Za definisanje modela se koriste formule u ćelijama tabele.

Ciljna funkcija se definiše unosom formule za ciljnu jednačinu u izabranu ćeliju tabele. Dobro je u susedne ćelije uneti i tekst sa nazivom i objašnjenjem značenja ciljne funkcije. Elementi jednačine su reference na ćelije ulaznih promenljivih. Vrednosti ulaznih ili promenljivih odlučivanja algoritam *Solvera* menja, čime se automatski menja i vrednost ciljne funkcije.

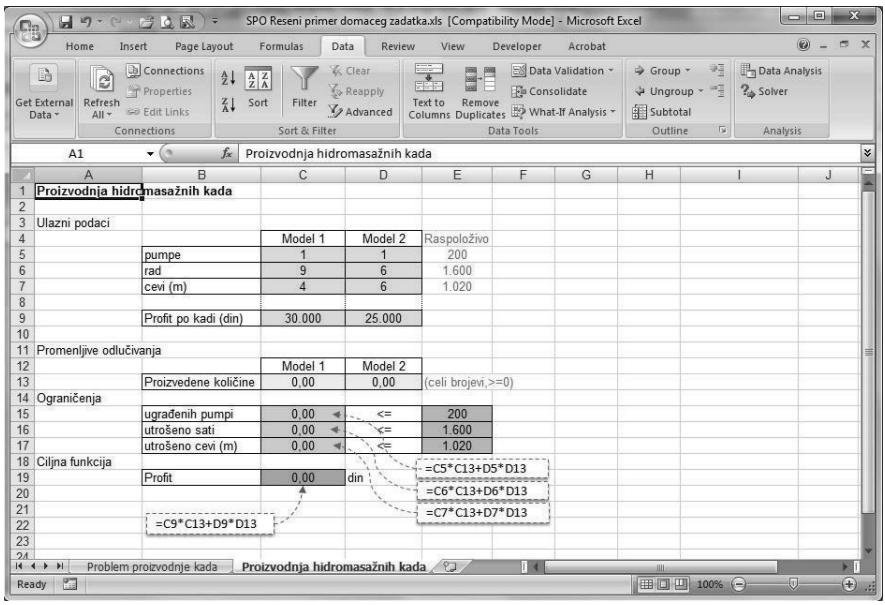
Ograničenja se unose kao niz jednačina ili nejednačina, pri čemu je poželjno u susednu ćeliju za svako ograničenje uneti tekstualni opis.

Najvažniji deo izrade tabele ograničenja je unos formula, koje obično predstavljaju levu stranu jednakosti. Sva ograničenja se odnose na vrednosti ulaznih promenljivih, koje *Solver* menja u toku traženja rešenja, pa se vrednosti ograničenja izračunavaju automatski.

Desna strana ograničenja su vrednosti, koje se unose desno od (ne)jednakosti. Između njih se, samo radi preglednosti, u posebne ćelije unose oznake (ne)jednakosti.

Za složenije probleme treba koristiti mogućnost imenovanja adesa grupa ćelija, kao što su reference na intervale vrednosti u formulama ograničenja i ćelije u kojima se nalaze promenljive odlučivanja. Imenovanje je posebno korisno kada se *Excel* tabele dopunjavju VBA kodom.

Na Sl. 28 prikazana je moguća organizacija Excel tabele za prikaz problema proizvodnje hidromasažnih kada. Vizuelno su odvojeni i istaknuti različitim bojama ulazni podaci, promenljive odlučivanja, ograničenja i ciljna funkcija. Dobra praksa je da se na prethodnom listu sačuva originalni tekst problema i opiše matematički model koji je iz njega izveden.



Sl. 28. Prvi deo računarskog modela: Excel tabela

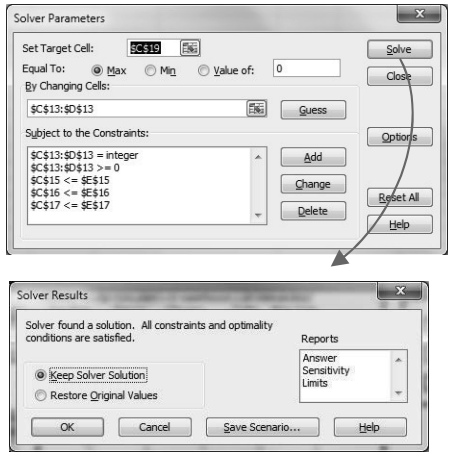
Unos modela linearnog programiranja

Računarski podržano rešenje se dobija korišćenjem dodatka Microsoft Excel *Solver*. Definicija problema linearnog programiranja čuva se kao objekt uz Excel tabelu sa podacima.

Za kreiranje računarskog modela linearnog programiranja prvo se unosi adresa ćelije u kojoj se nalazi formula ciljne funkcije, Sl. 29. Za tu ćeliju se definiše cilj optimizacije, minimum, maksimum ili neka određena vrednost.

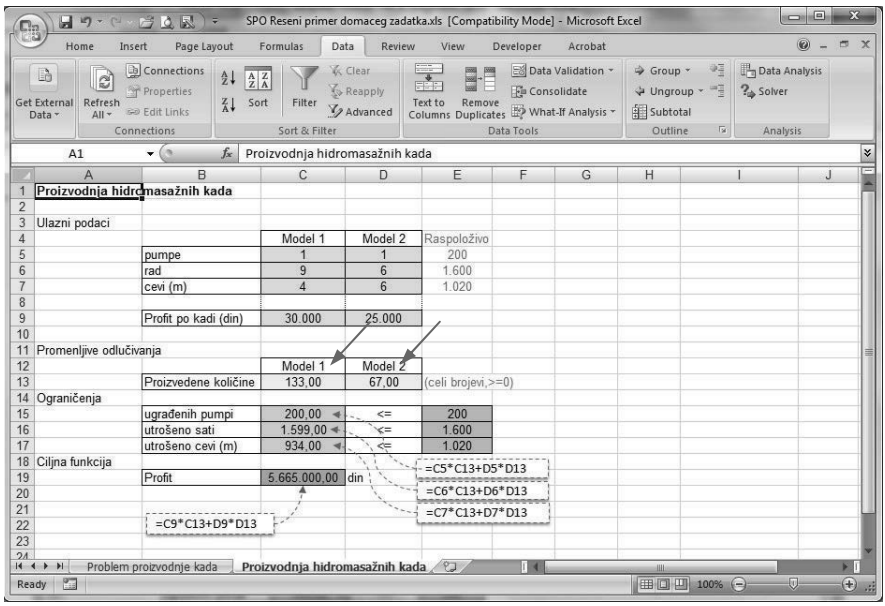
Nakon toga se unose adrese ćelija tabele koje predstavljaju promenljive sistema u koje će sistem uneti rešenje.

Poslednji korak je unos ograničenja nad skupom promenljivih, koji se posebno dodaju i ažuriraju korišćenjem tastera i posebnog programskog interfejsa. Model se izvršava komandnim tasterom *Solve*, Sl. 29.



Sl. 29. Drugi deo računarskog modela: Solver model

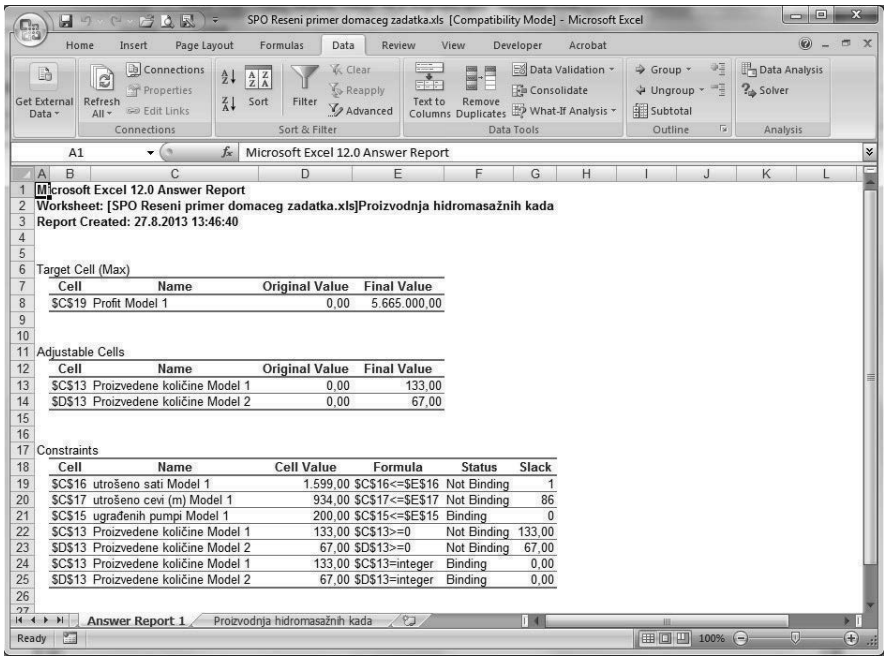
Rešenje problema se dobija u ćelijama predviđenim za promenljive sistema, Sl. 30. Na osnovu njih se izračunavaju sve ostale vrednosti, uključujući vrednost ciljne funkcije.



Sl. 30. Rešenje problema linearnog programiranja metodom Solver

Osim vrednosti promenljivih odlučivanja, koje se dobiju u izabranim ćelijama elektronske tabele, može se u posebnom listu dobiti i detaljniji izveštaj o procesu rešavanja (*Answer Report*), Sl. 31.

U izveštaju se, između ostalog, vide i vrednosti izravnavajućih promneljivih (*Slack*) koje je koristio simpleks metod prilikom rešavanja sistema.



Sl. 31. Detaljni izveštaj sistema Solver

Simpleks metod

Najčešći metod rešavanja sistema linearnih jednačina, kojima se opisuju problemi linearnog programiranja, je tzv. *simpleks* metod. Za primenu simpleks metoda, problem je prvo potrebno formulisati u standardnoj formi, tako da se sva ograničenja pretvore u oblik jednakosti, a ciljna funkcija u takav oblik da se traži njen maksimum [8].

Pretvaranje ograničenja u obliku nejednakosti u oblik jednakosti

Priprema za rešavanje problema simpleks metodom je pretvaranje ograničenja iz oblika nejednakosti u oblik jednakosti. Postupak se sastoji u dodavanju svakoj nejednakosti oblika više/više ili jednako izravnavajuće promenljive (*slack variable*), odnosno oduzimanju izravnavajuće promenljive svakoj nejednakosti oblika manje/manje ili jednako. Npr. za ograničenje oblika manje/manje ili jednako:

$$a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kn}x_n \leq b_k$$

dodaje se nenegativna promenljiva s_k, koja predstavlja razliku između vrednosti izraza s leve strane i konstante b_k, koja je veća ili jednaka od vrednosti izraza:

$$a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kn}x_n + s_k = b_k$$

Za ograničenje oblika veće/veće ili jednako, od vrednosti izraza s leve strane nejednakosti oduzima se nenegativna promenljiva s_k, koja predstavlja razliku između vrednosti izraza s leve strane i vrednosti konastante b_k:

$$a_{k1}x_1 + a_{k2}x_2 + \ldots + a_{kn}x_n - s_k = b_k$$

Za rešavanje problema proizvodnje hidromasažnih kada simpleks metodom, *Solver* rešava modifikovani problem, koji koristi tri izravnavajuće promenljive:

$$\begin{aligned} max \quad F = 30.000x_1 + 25.000x_2 \quad & profit \\ 1x_1 + 1x_2 + s_1 = 200 \quad & ograničen\ broj\ pumpi \\ 9x_1 + 6x_2 + s_2 = 1.600 \quad & ograničenje\ radnih\ sati \\ 4x_1 + 6x_2 + s_3 = 1.020 \quad & ograničenje\ dužine\ cevi\ (m) \\ x_1, x_2, s_1, s_2, s_3 \geq 0 \quad & uslov\ nenegativnosti \end{aligned}$$

Promenljive x_1 i x_2 su strukturne, a s_1, s_2 i s_3 su izravnavajuće promenljive modela. U *Solver* modelu potrebno je definisati samo strukturne promenljive. Izravnavajuće promenljive metoda generišu se automatski za svaki problem, a mogu se videt samo u sumarnom izveštaju (*Answer Report*), Sl. 31.

1. Osnovna dopustiva rešenja

Nakon pretvaranja ograničenja iz nejednakosti u oblik jednakosti, ograničenja predstavljaju sistem linearnih jednačina. Rešenje sistema od *m* jednačina i *n* promenljivih može se pronaći tako što se izabere *m* promenljivih i postave vrednosti koje zadovoljavaju sistem, dok se ostale promenljive postave na minimalnu vrednost (obično vrednost nula). Izabranih *m* promenljivih su osnovne (bazične) promenljive sistema, a rešenje koje se dobije korišćenjem samo osnovnih promenljivih, dok su ostale postavljene na nulu, naziva se osnovno dopustivo rešenje (*basic feasible solution*).

Svako osnovno dopustivo rešenje odgovara jednoj ekstremnoj tački regiona dopustivih rešenja problema linearnog programiranja. Optimalno rešenje problema se takođe nalazi u nekoj od ektremnih tački regiona, tako da se rešenje problema linarnog programiranja svodi na pronalaženje skupa osnovnih promenljivih i njihovih optimalnih vrednosti, koje daju osnovno dopustivo rešenje koje odgovara optimalnoj tački regiona dopustivih rešenja.

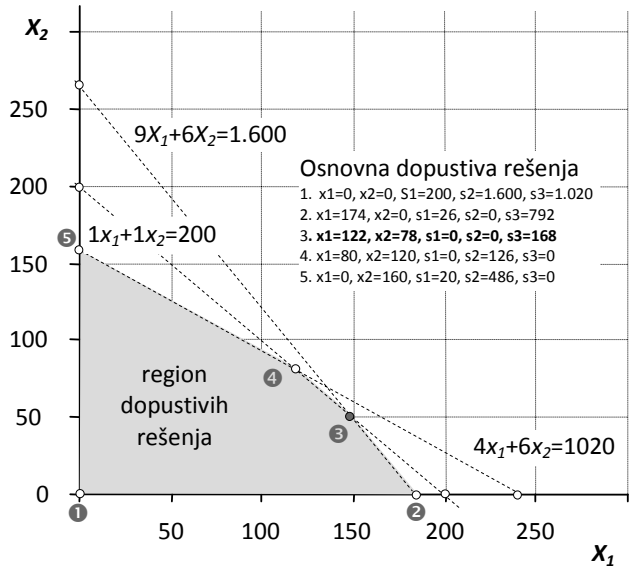
Modifikovani model problema proizvodnje hidromasažnih kada ima tri ograničenja i pet promenljivih [8]. Osnovno dopustivo rešenje od tri osnovne

promenljive može se formirati, izborom tri od pet promenljivih, na $\binom{n}{k}$= $\frac{n!}{k!(n-k)!} = \binom{5}{2} = \frac{5!}{2!3!} = \frac{120}{12} = 10$ različitih načina, Sl. 32.

	Osnovne promenljive	Ostale promenljive	Rešenje	Vrednost ciljne funkcije
1	s_1, s_2, s_3	x_1, x_2	x_1=0, x_2=0, s_1=200, s_2=1.566, s_3=2.880	0
2	x_1, s_1, s_3	x_2, s_2	x_1=174, x_2=0, s_1=26, s_2=0, s_3=792	60.900
3	x_1, x_2, s_3	s_1, s_2	x_1=122, x_2=78, s_1=0, s_2=0, s_3=168	**66.100**
4	x_1, x_2, s_2	s_1, s_3	x_1=80, x_2=120, s_1=0, s_2=126, s_3=0	64.000
5	x_2, s_1, s_2	x_1, s_3	x_1=0, x_2=180, s_1=20, s_2=486, s_3=0	54.000
6	x_1, x_2, s_1	s_2, s_3	x_1=108, x_2=99, s_1=-7, s_2=0, s_3=0	67.500
7	x_1, s_1, s_2	x_2, s_3	x_1=240, x_2=0, s_1=-40, s_2=-594, s_3=0	84.000
8	x_1, s_2, s_3	x_2, s_1	x_1=200, x_2=0, s_1=0, s_2=-234, s_3=480	70.000
9	x_2, s_2, s_3	x_1, s_1	x_1=0, x_2=200, s_1=0, s_2=366, s_3=-320	60.000
10	x_2, s_1, s_3	x_1, s_2	x_1=0, x_2=261, s_1=-61, s_2=0, s_3=-1.296	78.300

Sl. 32. Moguća osnovna dopustiva rešenja za problem proizvodnje kada (1-5)

Prvih pet rešenja iz tabele su dopustiva i predstavljaju osnovna dopustiva rešenja. Ostala rešenja nisu dopustiva, jer imaju negativne vrednosti promenljivih, pa narušavaju uslov nenegativnosti.



Sl. 33. Ilustracija veze između osnovnih dopustivih rešenja i ekstremnih tački

Najbolje dopustivo rešenje u tabeli je optimalno rešenje problema, gde su osnovne promenljive x_1, x_2 i s_3, a izravnavajuće s_1 i s_2, kojima je dodeljena minimalna vrednost nula.

Traže se vrednosti za x_1, x_2 i s_3, koja zadovoljavaju ograničenja

$$1x_1 + 1x_2 = 200 \quad ograničen\ broj\ pumpi$$
$$9x_1 + 6x_2 = 1.600 \quad ograničenje\ radnih\ sati$$
$$4x_1 + 6x_2 + s_3 = 1.020 \quad ograničenje\ dužine\ cevi\ (m)$$

Izravnavajuće promenljive s_1 i s_2 izostavljene su u modifikovanim ograničenjima s jednakostima, jer imaju vrednost nula.

Primena simpleks metoda daje rešenje s vrednostima x_1 = 122, x_2 = 78 i s_3 = 168, koje zadovoljava ograničenja i daje najveću vrednost ciljne funkcije F = 66.100.

Slika 33 prikazuju relacije između osnovnih dopustivih rešenja sa Sl. 32 i ekstremnih tačaka regiona dopustivih rešenja problema.

2. Pronalaženje dopustivih rešenja

Simpleks metod prvo identifikuje sva osnovna dopustiva rešenja, odnosno ekstremne tačke problema linearnog programiranja, nakon čega prelazi na susedne ekstremne tačke, ako daju bolju vrednost ciljne funkcije od tekuće tačke. Ako nema susednih tačaka s boljom vrednošću ciljne funkcije, tekuća tačka je optimalna i simpleks metod okončava.

Proces premeštanja s jedne ekstremne tačke na drugu vrši se tako što se neka od osnovnih promenljivih zameni jednom izravnavajućom promenljivom i tako kreira novo osnovno dopustivo rešenje, koje odgovara susednim ekstremnim tačkama. Na Sl. 34, premeštanje s prvog osnovnog dopustivog rešenja (tačka ❶) na drugo osnovno dopustivo rešenje (tačka ❷) vrši se tako što x_1 postaje osnovna, a s_1 izravnavajuća promenljiva. Simpleks metod se može premeštati od tačke ❶, preko tačke ❷ do tačke ❺, ali i obrnutim redosledom. Metod pronalazi optimalno rešenje, ali put do rešenja ne mora biti najkraći.

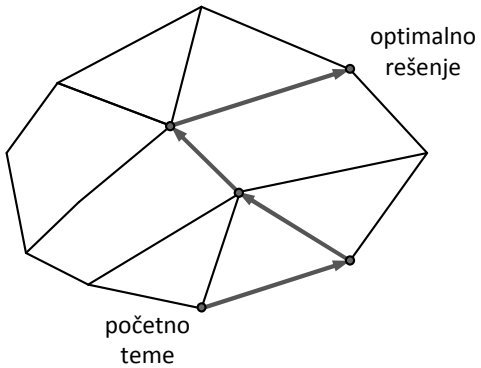
Za određivanje da li će zamena osnovnih i izravnavajućih promenljivih dati bolju vrednost ciljne funkcije, simpleks metod računa redukovanu cenu za sve izravnavajuće promenljive, da odredi da li će se ciljna funkcija poboljšati ako se neka od izravnavajućih promenljivih zameni osnovnom. Proces se nastavlja sve dok su moguća poboljšanja ciljne funkcije.

3. Geometrijska interpretacija algoritma

Simpleks metod je efikasan način rešavanja problema linearnog programiranja, jer razmatra samo ekstremne tačke regiona dopustivih rešenja. Nove, izravnavajuće promenljive, uvode se radi pretvaranja ograničenja iz oblika nejednakosti u oblik jednakosti. Metod se sistematski premešta na susedne tačke na uglovima regiona dopustivih rešenja, koji daju sve bolje vrednosti ciljne

funkcije. Geometrijski, region dopustivih rešenja za problem s *n* promenljivih predstavlja *n*-dimenzionalni politop, Sl. 35 (a).

Simpleks algoritam traži optimalno rešenje na *temenima* politopa. Opšti oblik simpleks algoritma pretraživanja je na Sl. 35 (b) [61]. Algoritam počinje od nekog temena dopustivog politopa i premešta se duž njegovih ivica dok ne pronađe teme koje predstavlja optimalno rešenje, kada okončava.



(a) Grafički prikaz rada simpleks algoritma

```
begin
  Generiši početno teme regiona dopustivih rešenja X0
  for k = 0..n
    if  teme Xk bolje od svih susednih u regionu dopustivih rešenja
    then  teme Xk je optimalno rešenje (kraj)
    else
      k = k + 1;
      Generiši novo rešenje Xk (teme regiona dopustivih rešenja)
      za koje je vrednost ciljne funkcije bolja.
end
```

(b) Opšti oblik simpleks algoritma

Sl. 34. Prikaz Simpleks algoritma

3.2.2 Tabele i stabla odlučivanja

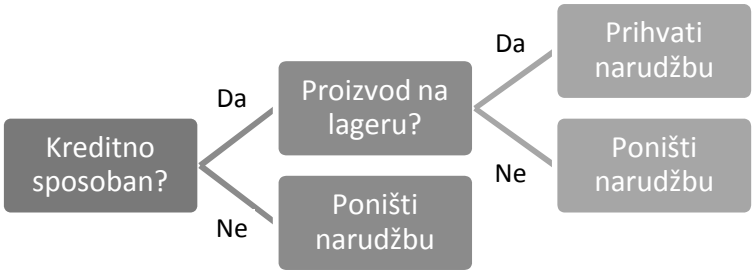
Problemi odlučivanja se mogu predstaviti i efikasno analizirati u tabelarnom obliku, npr. pomoću tabela odlučivanja (*decision tables*) i matrica isplate (*payoff matrix*), ali se takođe veoma pregledno mogu predstaviti i grafički, pomoću modela poznatog kao stablo odlučivanja (*decision tree*). Tabele odlučivanja i stabla odlučivanja su *ekvivalentni* modeli, kao što se vidi u primeru modela donošenja odluke o potvrdi narudžbe:

a) model odlučivanja prikazan tabelom odlučivanja

Proces potvrde narudžbe

	1	2	3	4
Kreditno sposoban	Da	Da	Ne	Ne
Proizvod na lageru	Da	Ne	Da	Ne
Prihvati narudžbu	X			
Poništi narudžbu		X	X	X

b) model odlučivanja prikazan stablom odlučivanja

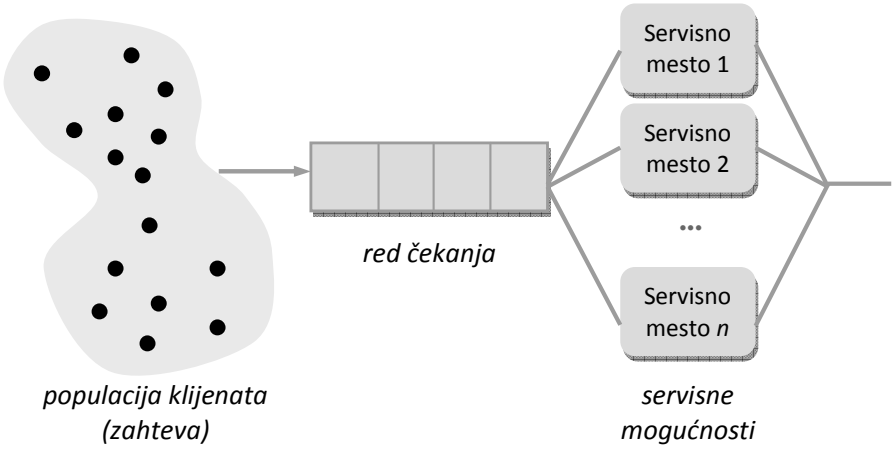


3.2.3 Modeli redova čekanja

Postoje analitički modeli za dinamičke sisteme, koji modeliraju promene sistema u vremenu. Primer su modeli *redova čekanja* (*queues*) ili mreža redova čekanja, koji mogu da modeliraju određene klase dinamičkih sistema. Za složenije situacije, koriste se stohastički simulacioni modeli s istim osnovnim komponentama modela, ali različitim parametrima [10], [22].

Model reda čekanja se sastoji od tri osnovne komponente: *izvora* zahteva, koji generiše stohastički proces u kome entiteti ili klijenti, stižu na datu lokaciju radi dobijanja usluge, skupa *resursa* koji pružaju usluge i *reda* čekanja (*queue*, *waiting area*) koji može da primi entitete čiji se zahtevi ne mogu odmah zadovoljiti.

Osnovne komponente sistema jednog reda čekanja su *populacija*, proces pristizanja zahteva (*arrivals*), proces opsluživanja (*service*), broj servisnih mesta (*stations*) i servisna disciplina ili pravila reda čekanja (*waiting line rules*), Sl. 35.



Sl. 35. Osnovni elementi reda čekanja

Populacija može biti konačna i beskonačna, a predstavlja izvor iz koga dolaze potencijalni klijenti i u koji se vraćaju oni koji su tražili i dobili uslugu. Proces dolaska zahteva opisuje kako korisnici dolaze na ulaz sistema. To je stohastički proces, opisan verovatnoćom raspodele vremena između dolazaka, odnosno vremenskog intervala između dolazaka dva uzastopna klijenata.

Proces usluživanja opisuje kako davaoci usluga zadovoljavaju zahteve kupaca koji čekaju u redu. Ovo je stohastički proces, definisan verovatnoćom raspodele vremena opsluživanja, odnosno, vreme koje klijenti provode sa resursima koji pružaju usluge.

Broj postojećih stanica i broj davalaca usluga za svaku stanicu su dodatni relevantni parametri reda čekanja sistema. Pravila reda čekanja definišu redosled kojim se klijenti izuzimaju iz linije čekanja radi pružanja usluga. Šema prioriteta određuje nivo prioriteta svakog klijenta. Klijent sa najvišim prioritetom se opslužuje pre ostalih klijenata koji čekaju u redu.

Kratki opis svojstava reda čekanja se može dati pomoću Kendalove notacije u formi:

$$A/B/m/N - S$$

gde je *A* raspodela nailaska zahteva, *B* raspodela vremena opsluživanja, *m* je broj servisnih mesta, *N* maksimalni broj zahteva u redu čekanja (ako je 1, može se izostaviti).

Raspodele A i B su najčešće *M* (Markov, eksponancijalna distribucija), *D* (*Deterministic*), E_k (*Erlang*-k), H_k (*Hyper*-k) i *G* (*General*, nedefinisana distribucija). Servisna disciplina *S* se može izostaviti, kada se podrazumeva FIFO. Najčešće se koriste servisne discipline:

- FIFO (*First In, First Out*), gde zahtevi idu na kraj reda čekanja, a opslužuje se prvi u redu;
- LIFO (*Last In, First Out*), gde zahtev ide na čelo reda i opslužuje se prvi, ako ne naiđe sledeći;
- slučajni (*random*), gde se zahtevi iz reda čekanja opslužuju slučajnim redosledom određeno vreme, nakon čega se vraćaju u red čekanja ako nisu do kraja realizovani;
- prioritetni (*priority*), gde svaki klijent ima statički ili dinamički prioritet, po kome se vrši opsluživanje.

Najjednostavniji model reda čekanja je M/M/1, u kome su distribucija zahteva (parametar λ) i vremena opsluživanja (parametar μ) eksponencijalni, red čekanja neograničen, jedno servisno mesto i FIFO disciplina opsluživanja.

Osnovni cilj teorije linija čekanja je da ispita pojave zagušenja (*congestion*), koje se dešavaju kada su potražnja i obezbeđivanje usluga stohastičke prirode. Npr. ako vreme dolaska kupaca i vreme trajanja opsluživanja nisu poznati unapred na deterministički način, mogu nastati konflikti između klijenata u korišćenju ograničenih zajedničkih resursa. Posledica je da neki korisnici moraju da čekaju u redu.

Modeli redova čekanja omogućavaju evaluaciju performansi sistema definisane strukture, pa se uglavnom koriste u fazi projektovanja sistema. U cilju utvrđivanja odgovarajućih vrednosti parametara koji karakterišu novi sistem, razmatraju se relevantni ekonomski faktori, koji zavise od nivoa usluga koje sistem treba da garantuje kada radi u optimalnim uslovima. Model uzima u obzir troškove zadovoljavanja zahteva, koji se povećavaju kako se povećava nivo usluga i troškove vremena provedenog u redu čekanja, koji se smanjuju kako se nivo pružanja usluga smanjuje. Klijenti očekuju kraće čekanje, dok je davalac usluga zainteresovan za niže troškove pružanja usluga.

Struktura sistema se definiše tako da se dobije optimalno rešenje, koje predstavlja kompromis između troškova pružanja usluga i troškova linije čekanja. Optimalni nivo usluga, koji određuje idealnu strukturu sistema, dobija se kao minimum krive koja predstavlja sumu ove dve vrste troškova [22].

3.2.4 Opšti metodi rešavanja problema

Opšti metodi rešavanja problema koriste se za probleme za koje ne postoje ili nisu dovoljno efikasni drugi modeli i metodi rešavanja. Izučavaju se u okviru veštačke inteligencije, koja pretpostavlja da ove probleme treba da reši računarski program, koji se naziva agent, obično racionalni ili inteligentni agent [4]. Racionalni i inteligentni agenti maksimizuju svoje performanse, koje se

iskazuju ciljem i nekom merom performansi, koja meri stepen dostizanja postavljenog cilja.

Problem predstavlja *niz stanja* kroz koje agent treba da prođe da bi dostigao cilj, tako da *maksimizuje* svoju *meru kvaliteta*. Formulacija problema je proces određivanja akcija i stanja koje treba razmotriti za postizanje zadanog cilja.

Rešavanje problema (*problem solving*) je pronalaženje niza akcija koji vodi od početnog do ciljnog stanja. Rešenje problema je niz akcija ili *putanja* kroz prostor stanja, a kvalitet rešenja je definisan funkcijom cene puta. Optimalno rešenje je rešenje s najmanjom cenom od svih mogućih rešenja.

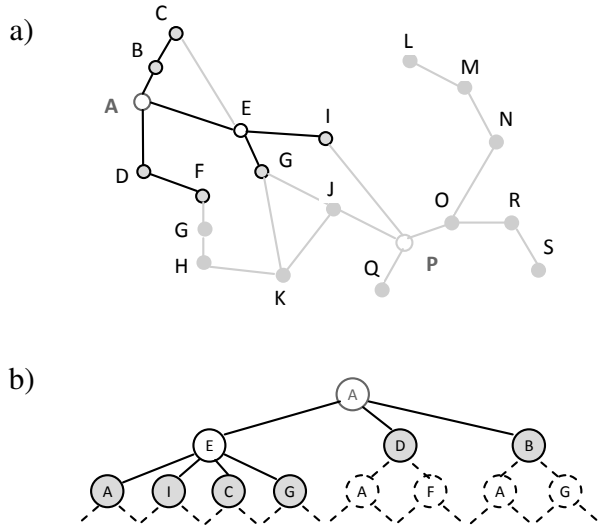
Pretraživanje je proces pronalaženja niza akcija ili putanje kroz prostor stanja, koja vodi od početnog do ciljnog stanja. Ulaz algoritma pretraživanja je definicija problema, a izlaz je rešenje, odnosno sekvenca akcija agenta. Prostor stanja može biti u celini poznat, delimično poznat ili sasvim nepoznat, što zahteva različite pristupe pretraživanju.

Mogući niz akcija, od početnog stanja do cilja, ima oblik stabla, čiji koren je početno stanje, grane stabla su akcije, a čvorovi su stanja iz prostora stanja problema. Takvo stablo se naziva *stablo pretraživanja* (*search tree*). Primer dela stabla pretraživanja za problem pronalaženja optimalnog puta između dve lokacije je na Sl. 36 [4].

Na Sl. 36 (a) prikazana je saobraćajna mreža, s poznatim rastojanjima između naselja *A* do *S*. Problem je da se pronađe najkraći put od naselja *A* do naselja *P*. Na Sl. 36 (b) prikazan je samo vrh stabla pretraživanja, koje se dobije nakon otvaranja čvora *E*. U svakom čvoru se zna ukupna dužina pređenog puta, kao i rastojanje do susednih mesta. Postoje različite heuristike za izbor sledećeg čvora koji će se posetiti, odnosno otvoriti, što znanto utiče na performanse pretraživanja.

Opšti metodi rešavanja problema su metodi pretraživanja prostora mogućih rešenja [1]:

1. analitičko pretraživanje
2. slepo pretraživanje
3. heurističko pretraživanje



Sl. 36. Prostor stanja (a) i vrh stabla pretraživanja (b) optimalnog puta

Analitičko pretraživanje

Analitičke tehnike pretraživanja koriste matematičke formule za direktno izvođenje ili predviđanje rezultata prilikom rešavanja strukturiranih problema [1], [4].

Slepo pretraživanje

Koristi se za rešavanje složenih, nestrukturiranih problema [1]. Prethodno je potrebno definisati, odnosno opisati *cilj* pretraživanja. Prilikom pretraživanja, vrši se detaljno ispitivanje (enumeracija) svih mogućnosti, korak po korak, sve dok se ne dostigne cilj. Pretraživanje može biti *kompletno* ili *delimično* (parcijalno). Prilikom delimičnog pretraživanja, ispituje se samo deo mogućnosti, sve dok se ne pronađe dovoljno dobro rešenje po zadanom kriterijumu, odnosno *suboptimalno* rešenje. Pošto je prostor mogućnosti često enormno velik (npr. u šahu), postoje praktična ograničenja prilikom slepog pretraživanja, pre svega raspoloživo vreme i ograničeni memorijski resursi.

Heurističko pretraživanje

Broj razmotrenih mogućnosti u procesu pretraživanja može se smanjiti korišćenjem dodatnih informacija iz predznanja o problemu ili heuristika. Heuristika je neformalno znanje, koje pomaže donošenju odluka u nekoj oblasti primene. Heurističko pretraživanje ograničava prostor mogućnosti koje treba pretražiti i usmerava pretraživanje na one mogućnosti za koje se očekuje da su bliže optimalnom rešenju. Na taj način, korak po korak, približava se optimalnom rešenju i eventualno ga pronalazi [1].

Primeri poznatih algoritama heurističkog pretraživanja su [4]:

- *Best-First* algoritam koristi heurističku evaluaciju čvorova stabla $f(n)$, koja predstavlja ocenu poželjnosti čvora kandidata; na svakom koraku uređuje listu čvorova stabla prema opadajućoj vrednosti ove funkcije.
- *Hill-climbing* je algoritam lokalnog pretraživanja, koji u memoriji čuva samo tekuće stanje, koje se pokušava unaprediti tako što se bira najbolje ocenjeni susedni čvor kao sledeći kandidat.
- *Simulated annealing* je metod koji pokušava da izbege padanje u lokalni maksimum kod metoda lokalnog pretraživanja tako što dopušta izbor lošijih poteza, čija se frekvencija postepeno smanjuje; pod određenim uslovima, algoritam pronalazi globalni maksimum s verovatnoćom koja teži jedinici.
- *Genetički algoritmi*, gde se sledeća stanja ne generišu na osnovu jednog tekućeg stanja, već kombinacijom dva prethodna (roditeljska) stanja. Početak je k slučajno generisanih stanja, tzv. *populacija*, koja se predstavljaju nizovima konačnog alfabeta, npr. 0,1; funkcija evaluacije stanja (*fitness function*) ima veću vrednost za poželjnija stanja; sledeća generacija stanja proizvodi se selekcijom, ukrštanjem i mutacijom (*selection*, *crossover*, *mutation*) parova prethodno razmotrenih stanja.

3.2.5 Simulacioni modeli

Simulacija je imitacija realnosti, kroz eksperiment, koji omogućava da se vreme eksperimenta skrati. Koristi se za računarsku podršku rešavanja nestrukturiranih problema. Simulacioni modeli ne garantuju optimalno rešenje.

Osnovne vrste su *stohastičke* simulacije, *vremenski zavisne* i *vremenski nezavisne* simulacije, *objektno orijentisane* i *vizuelne* simulacije [1]. Stohastičke simulacije mogu biti *diskretne* i *kontinualne*, u zavisnosti od vrste distribucije koja se koristi za generisanje slučajnih događaja. Standardna metodologija rešavanja nekog problema korišćenjem simulacije je:

1. definisanje problema
2. kreiranje modela
3. testiranje i validacija
4. dizajn eksperimenta
5. *eksperiment*
6. ocena (evaluacija)
7. implementacija rešenja

Metod Monte Karlo

Metod *Monte Karlo* je metod stohastičke simulacije, koji koristi generator (pseudo)slučajnih brojeva za generisanje velikog broja eksperimenata u cilju

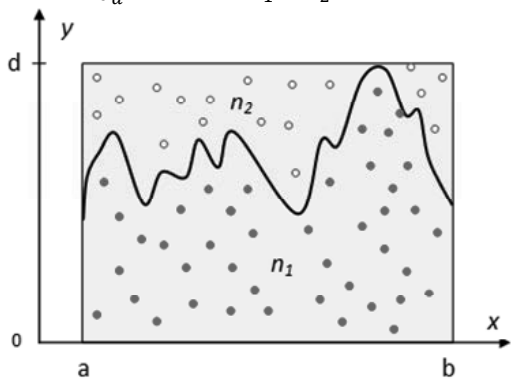
procene vrednosti neke veličine. Metod je uveden u okviru istraživanja vezanih za razvoj hidrogenske bombe i vezuje se za ime naučnika Džona fon Nojmana, koji je 1947. godine detaljno razradio predlog metoda, na osnovu ideja svojih saradnika. Naziv metoda naglašava korišćenje slučajnosti, kao kod bacanja kocke u kockarnici.

U ekonomiji se koristi za ocenu poslovnog rizika, promena vrednosti investicija, strategijsko planiranje i rešavanje drugih problema koji uključuju neizvesnost.

Primer: Integracija empirijske funkcije

Na Sl. 37 je klasična ilustracija izračunavanja površine ispod empirijske krive $f(x)$, $a \le x \le b$, $0 \le y \le d$, $d = \max f(x)$, koja nema pogodan analitički oblik za računanje integrala. Površina ispod krive na segmentu $[a,b]$ računa se kao odgovarajući deo površine pravougaonika $(b\text{-}a)\cdot d$, koji se dobija na osnovu relativnog odnosa broja slučajno generisanih tački koje padaju iznad i ispod krive.

$$P = \int_a^b f(x)dx \approx \frac{n_1}{n_1 + n_2}(b-a)\cdot d$$



Sl. 37. Monte Karlo metod računanja integrala empirijske krive

Greška ovakve procene površine zavisi od broja slučajno generisanih tačaka n. Potreban broj tačaka (eksperimenata) za željenu tačnost ε može se izračunati na osnovu izraza koji daje ocenu greške procene $\varepsilon \sim 1/\sqrt{n}$ [6].

Postoji veliki broj softverskih alata za sve vrste simulacija. Poznati tradicionalni softverski alati zasnovani su na posebnim jezicima, kao što su GPSS (*General Purpose Simulation Ssstem*) i SIMSCRIPT [61]. Većina novijih softverskih alata podrazumeva Web okruženje. Poznatiji Web alati su *Simul8* [76] i WebGPSS. Niz alata su dodaci nekih programskih okruženja, kao što su npr. *Excel* dodaci *Analytica* i Palisade *@Risk* [75] i Matlab dodatak *Simulink* [1].

3.3 Softverska podrška modeliranju

Softver za modeliranje i analizu koristi se kao neposredna podrška odlučivanju ili za razvoj sistema za podršku odlučivanju. Podrška rešavanju različitih vrsta problema odlučivanja ostvaruje se primenom odgovarajućih modela, za čije rešavanje su razvijene brojne tehnike i odgovarajući sistemi i alati.

Neki od značajnih kategorija modela za podršku odlučivanju, koji imaju dobru softversku podršku su: optimizacioni modeli, simulacioni modeli, prediktivni modeli, modeli mašinskog učenja (*machine learning*), modeli upravljanja projektima (*project management*), modeli analize rizika (*risk analysis*) i modeli mreža repova ili linija čekanja (*queue*, *waiting line*) [10]. U ovom materijalu koriste se softverski alati za podršku razvoju prediktivnih, optimizacionih i modela mašinskog učenja.

Optimizacioni modeli su primenjivi na veliki broj strukturiranih problema odlučivanja, zbog mogućnosti jasne matematičke formulacije i postojanja efikasnih metoda za njihovo rešavanje. Primeri matematičkih modela optimizacije su *linearna* optimizacija, *celobrojna* optimizacije, *konveksna* optimizacija, *mrežna* optimizacija i *višekriterijumska* optimizacija. U ovom materijalu se kao alat za linearnu optimizaciju koristi Microsoft Excel *Solver*, dok se za višekriterijumsku optimizaciju koristi besplatni program *DEXi* [27].

Prediktivni modeli obuhvataju dve osnovne kategorije modela: *eksplanatorni* modeli (modeli regresije i klasifikacije, koji istovremeno spadaju u grupu modela mašinskog učenja i prepoznavanja uzoraka) i modeli *vremenskih serija* (*time series*), koji funkcionalno identifikuju vremenske uzorke najčešće numeričkih vrednosti. Koriste se u sistemima poslovne inteligencije, a u ovom materijalu su zastupljeni u okviru alata mašinskog učenja, npr. Pentaho/*Weka*, *SQL Server* i Microsoft Excel *Data Mining Add-In*.

Modeli mašinskog učenja koriste se za kreiranje inteligentnih sistema, koji mogu da uče na osnovu opažanja prošlosti i mogu da razvijaju nova pravila za primenu u budućnosti. Osnovni metodi su metodi *učenja s učiteljem* (*supervised learning*), gde spadaju regresioni i klasifikacioni modeli i *učenja bez učitelja* (*unsupervised learning*), gde spadaju metodi klasteringa. U ovom materijalu se koriste sistemi *Weka*, *SQL Server* i Microsoft Excel *Data Mining Add-In*, kao i programske biblioteke sistema *R*, u kojima su implementirani brojni metodi mašinskog učenja.

3.4 Primeri rešavanja strukturiranih problema

Praktična pimena matematičkih modela će se demonstrirati na dva primera strukturiranih problema, koji se rešavaju metodom linearnog programiranja i Monte Karlo simulacije.

Primer 1: Planiranje proizvodnje (linearno programiranje)

Kompanija proizvodi šest različitih vrsta proizvoda. Proizvodnja svakog proizvoda zahteva utrošak rada i sirovina, prema tabeli.

	Proizvod 1	Proizvod 2	Proizvod 3	Proizvod 4	Proizvod 5	Proizvod 6
rad	6	5	4	3	2,5	1,5
sirovine	3,2	2,6	1,5	0,8	0,7	0,3

Kompanija ima određene količine ovih resursa.

Postoje ograničene potrebe za količinama svake vrste proizvoda, koje su istovremeno i maksimalne količine koje se mogu proizvesti.

	Proizvod 1	Proizvod 2	Proizvod 3	Proizvod 4	Proizvod 5	Proizvod 6
potražnja	960	928	1.041	977	1.084	1.055

Zadane su tabele neophodnih resursa i potreba za proizvodima, po vrstama, kao i profit za svaki proizvod.

	Proizvod 1	Proizvod 2	Proizvod 3	Proizvod 4	Proizvod 5	Proizvod 6
cena/kom	12,5	11	9	7	6	3
troškovi	6,5	5,7	3,6	2,8	2,2	1,2
profit /kom	6	5,3	5,4	4,2	3,8	1,8

Potrebno je planirati proizvodnju određenih količina svakog proizvoda, tako da se *maksimizuje* profit.

Postavka rešenja

1. *Ulazne promenljive*

U ovom problemu je potrebno odrediti nepoznate optimalne količine svake od 6 vrsta proizvoda koji će se proizvesti:

$x_1, x_2, x_3, x_4, x_5, x_6$

2. *Ciljna funkcija*

Cilj je maksimizacija profita. Profit se računa pomoću zadanog profita p (tabela *Profit po jed. proizvoda*) i proizvedene količine x određene vrste proizvoda:

$$P = p_1*x_1 + p_2*x_2 + p_3*x_3 + p_4*x_4 + p_5*x_5 + p_6*x_6$$

Treba maksimizovati funkciju P, što se (programski) postiže podešavanjem proizvedenih količina svakog proizvoda.

3. *Ograničenja*

a) ograničenja raspoloživog obima rada L:

$$L = l_1*x_1 + l_2*x_2 + l_3*x_3 + l_4*x_4 + l_5*x_5 + l_6*x_6 \leq obim\ rada = 4500$$

b) ograničene količine sirovina R, odmosno raspoloživog materijala:

$$R = r_1*x_1 + r_2*x_2 + r_3*x_3 + r_4*x_4 + r_5*x_5 + r_6*x_6 \leq materijal = 1600$$

Osim toga, moraju se zadovoljiti ukupni zahtevi D, s tim da se veće količine od zahtevanih se ne mogu proizvesti. Ograničenje zahtevanih količina je:

$$x_i \leq d_i, \text{ za } i = 1..6$$

Računarsko rešenje

1. Pregledno se u Excel tabeli rasporede i unesu ulazni podaci, odrede ćelije za vrednosti promenljivih odlučivanja, unesu formule za ograničenja (dve) i formula ciljne funkcije. Na slici je mogući raspored.

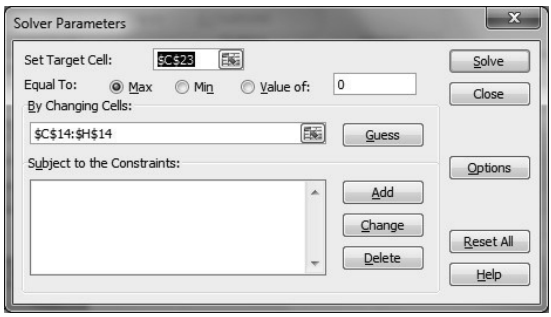
C23 =C10*C14+D10*D14+E10*E14+F10*F14+G10*G14+H10*H14

	A	B	C	D	E	F	G	H	I
1	**Planiranje proizvodnje**								
2									
3	Ulazni podaci								
4			Proizvod 1	Proizvod 2	Proizvod 3	Proizvod 4	Proizvod 5	Proizvod 6	Raspoložive količine
5		rad	6	5	4	3	2,5	1,5	4.500
6		sirovine	3,2	2,6	1,5	0,8	0,7	0,3	1.600
7		jedinična cena	12,5	11	9	7	6	3	
8		ostali troškovi	6,5	5,7	3,6	2,8	2,2	1,2	
9									
10		Profit po jed. proiz.	6	5,3	5,4	4,2	3,8	1,8	
11									
12	Ulazne promenljive								
13			Proizvod 1	Proizvod 2	Proizvod 3	Proizvod 4	Proizvod 5	Proizvod 6	
14		Proizvedene količ.	0,00	0,00	0,00	0,00	0,00	0,00	
15			<=	<=	<=	<=	<=	<=	
16		Tražene količine	960	928	1041	977	1084	1055	
17									
18	Ograničenja								
19		utrošeni rad	0,00	<=	4.500				
20		utrošene sirovine	0,00	<=	1.600				
21									
22	Ciljna funkcija								
23		Profit	0,00						

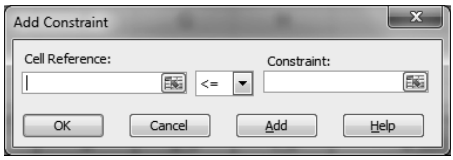
=C5*C14+D5*D14+E5*E14+F5*F14+G5*G14+H5*H14

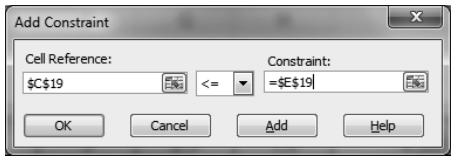
=C10*C14+D10*D14+E10*E14+F10*F14+G10*G14+H10*H14

2. Aktivira se Solver interfejs (**Tools/Solver**), označi ćelija funkcija cilja i vrsta optimizacije (minimum, maksimum ili ciljna vrednost) i promenljive odlučivanja.

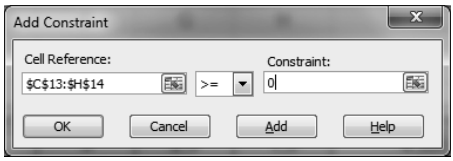


3. Uključe se ranije pripremljeni elementi ograničenja (jedno po jedno), npr. unos ograničenja obima rada $L \leq 4.500$:

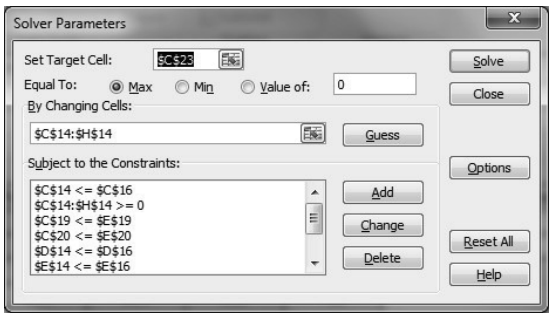




... i uslov pozitivnosti promenljivih odlučivanja, kao opšte ograničenje:



Konačni model linearnog programiranja (*Solver*) je kao na slici:



Rešavanje se pokreće tasterom *Solve*. Rezultati su kao na slici.

C23 fx =C10*C14+D10*D14+E10*E14+F10*F14+G10*G14+H10*H14

	A	B	C	D	E	F	G	H	I
1	**Planiranje proizvodnje**								
2									
3	Ulazni podaci								
4			Proizvod 1	Proizvod 2	Proizvod 3	Proizvod 4	Proizvod 5	Proizvod 6	Raspoložive količine
5		rad	6	5	4	3	2,5	1,5	4.500
6		sirovine	3,2	2,6	1,5	0,8	0,7	0,3	1.600
7		jedinična cena	12,5	11	9	7	6	3	
8		ostali troškovi	6,5	5,7	3,6	2,8	2,2	1,2	
9									
10		Profit po jed. proiz.	6	5,3	5,4	4,2	3,8	1,8	
11									
12	Ulazne promenljive								
13			Proizvod 1	Proizvod 2	Proizvod 3	Proizvod 4	Proizvod 5	Proizvod 6	
14		Proizvedene količ.	0,00	0,00	0,00	596,67	1.084,00	0,00	
15			<=	<=	<=	<=	<=	<=	
16		Tražene količine	960	928	1041	977	1084	1055	
17									
18	Ograničenja								
19		utrošeni rad	4.500,00	<=	4.500				
20		utrošene sirovine	1.236,13	<=	1.600				
21									
22	Ciljna funkcija								
23		Profit	6.625,20						

Maksimalni profit iznosi **6.625,20** i postiže se ako se svi raspoloživi radni i materijalni resursi upotrebe za proizvodnju **Proizvoda 4** i **Proizvoda 5**.

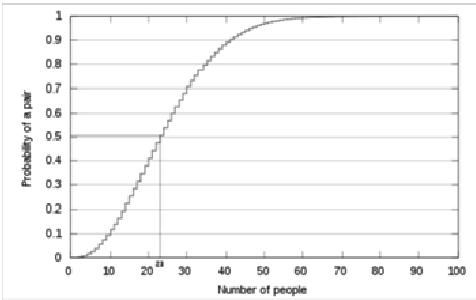
Primer 2: Problem rođendana (Monte Karlo simulacija)

Metodom Monte Karlo može se eksperimentalno oceniti verovatnoća da je dvoje slučajno izabranih ljudi iz neke grupe rođeno istog dana.

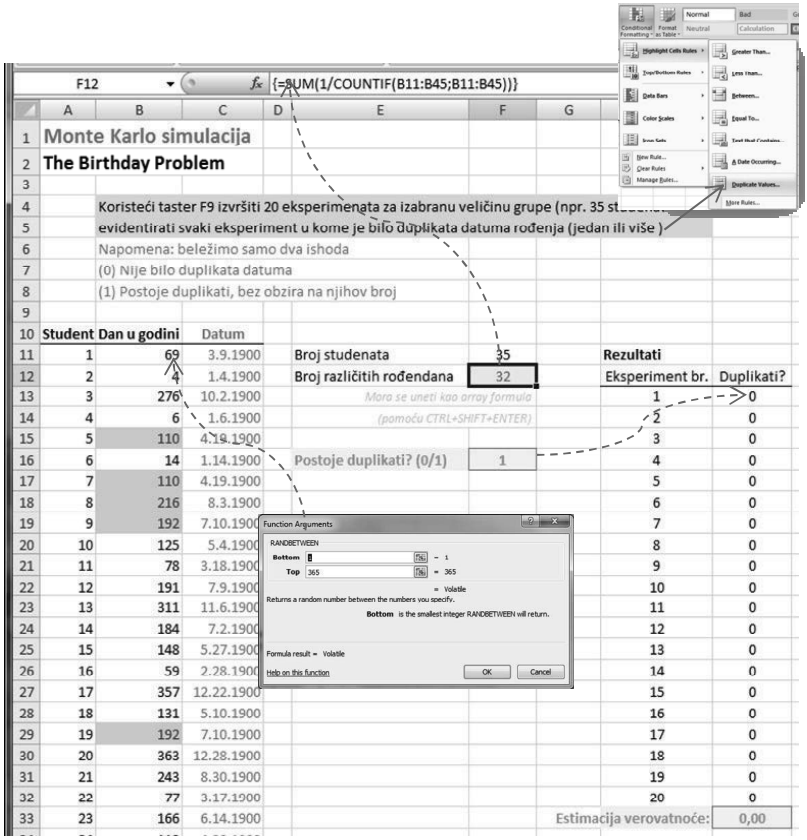
U pitanju je elementarni problem teorije verovatnoće, za koji se verovatnoća može egzaktno izračunati. Kada su događaji međusobno nezavisni, verovatnoća svih događaja je jednaka proizvodu verovatnoće svakog događaja ponaosob. Npr. za grupu od n = 35 studenata, verovatnoća da *ne* postoje dva studenta koji imaju rođendan istog dana može se izračunati kao $P(\bar{A})$= 365/365 × 364/365 × 363/365 × 362/365 × ... × 331/365. Tada je verovatnoća istih rođendana

$$P(A) = 1 - P(\bar{A}) = 1 - \prod_{i=1}^{n} \frac{365 - i + 1}{365} = 1 - \frac{1}{365^n} \prod_{i=1}^{n} (365 - i + 1)$$

Grafički prikaz zavisnosti verovatnoće $P(A)$ od veličine grupe, za n=1..100 ljudi je na slici [61]:



Kad je nepoznat način računanja ovakve veličine ili je računanje teško, za estimaciju verovatnoće se može koristiti metod Monte Karlo. Ovaj pristup je moguće realizovati korišćenjem Microsoft *Excel* elektronske tabele, pošto postoje *Excel* funkcije RANDOM(), za generisanje (pseudo) slučajnih brojeva u dijapazonu 0..1 i RANDBETWEEN(a, b), koja generiše slučajne cele brojeve u zadanim granicama [a..b].



Postupak je:

1) Kreira se radni list, koji je prikazan na sledećoj slici. Za broj studenata u svojoj nastavnoj grupi (u primeru na slici je uzeto n=35 studenata), funkcijom RANDBETWEEN(1,365) generiše se lista od n slučajnih celih brojeva 1..365 (broj dana u godini). Funkcija SUM pronalazi broj različitih dana u slučajno generisanoj listi.

2) Koristeći funkcijski taster F9, pokrene se m=20 eksperimenata za konkretnu nastavnu grupu.

Rezultat svakog eksperimenta (0/1) unosi se u kolonu "Duplikati?", tako da se nakon m ponavljanja dobija m_1, broj ekperimenata koji daju duplikate rođendana.

Estimacija verovatnoće vrši se kao $P(m) \approx m_1/m$ i automatski ažurira posle svakog eksperimenta.

Zanimljivo je uporediti rezultat dobijen Monte Karlo metodom s egzaktnom vrednošću ove verovatnoće, kao i stvarnim brojem istih rođendana u svojoj nastavnoj grupi.

Za dovoljno malu grešku estimacije, obično se mora izvršiti znatno veći broj eksperimenata (n>20).

Pitanja za ponavljanje

1. Koji je osnovni razlog modeliranja procesa odlučivanja?
2. Navedite osnovne vrste modela odlučivanja.
3. Navedite bar po jedan primer svake od vrsta modela odlučivanja.
4. Koje vrste matematičkih modela za podršku odlučivanju poznajete?
5. Šta se rešava postupkom optimizacije?
6. Objasnite pojam linearnog programiranja.
7. Šta je dualni problem linearnog programiranja?
8. Kakav je odnos između modela tabela odlučivanja i stabala odlučivanja?
9. Koje opšte metode rešavanja problema poznajete?
10. Kada se koriste simulacioni modeli?
11. Šta predstavlja metod Monte Karlo?

Zadaci za vežbanje

1. Zaposleni u fotokopirnici zarađuje 85.000 din mesečno na poslu upravljanja fotokopirnicom. Obavezan je da radi najmanje 135, a najviše od 180 sati mesečno. Kao dodatni posao, zaradi 1.400 din po satu za slaganje brošura za lokalne poslovne klijente.

   Po njegovoj proceni, za održavanje životnog standarda mu je neophodna zarada od 110.000 dinara mesečno. No, radi održavanja zadovoljavajućeg kvaliteta života, ograničio je svoj rad na ukupno 200 sati mesečno.

   Podrazumevajući da ne može da utiče na broj sati koji mora da radi u centru za kopiranje, da li će biti u mogućnosti da istovremeno zadovolji svoje ciljeve i ostvari dovoljne prihode? Objasniti.

2. Online apoteka prodaje vitaminske tablete u raznim pakovanjima. Pakovanje od 250 komada košta 1.500 din, a od 150 komada 1.200 din.

   Kupac planira da kupi najmanje 750 komada. Koliko pakovanja svake vrste (150 i 250 komada) treba da kupi, ako želi da minimizuje troškove?

3. Za opremanje kancelarije treba nabaviti kombinovane kancelarijske ormare. Cena jednog tipa ormara, koji zauzima 6 m^2 površine i ima 8,2 m^3 prostora za dokumente je 10.000 din po komadu. Cena drugog tipa ormara, koji zauzima 8,4 m^2 površine i ima 12 m^3 prostora za odlaganje je 14.000 din po komadu. Za nabavku je ukupno odobreno 150.000 din.

U kancelariji ima mesta za više od 72 m^2 ormara. Koliko treba nabaviti svakog od modela kancelarijskih ormara da se dobije maksimalni kapacitet ormara?

4. Svaki dan se u tri regiona zemlje utroši po 100 milijardi litara vode. Severni i centralni region imaju na raspolaganju po 120 milijardi litara vode, dok južni region ima na raspolaganju svega 40 milijardi litara vode.

   Troškovi dostave (u din) jedne milijarde litara vode u/između tri regiona su:

	Severni	Centralni	Južni
Severni	550.000	750.000	1.000.000
Centralni	700.000	500.000	600.000
Južni	1.000.000	6.000	500.000

   Pošto se ne mogu zadovoljiti svi zahtevi za vodom, svaka milijarda litara nezadovoljene tražnje stvara nestašice i sledeće troškove:

	Severni	Centralni	Južni
Cena po nedostajućoj milijardi litara	600.000	550.000	900.000

   Kako bi trebalo distribuirati vodu u tri regiona da se minimizuju ukupni troškovi isporuke i nedostajuće količine vode za piće?

5. Atletski gimnastičar dnevno treba da unese određene količine kalcijuma, proteina i kalorija. Njegov dijetalni doručak pretežno se sastoji od mleka i žitarica, čija su cena i hranljivi sadržaj u tabeli:

Sastojak	Mleko (1 dl)	Žitarice (1 dl)
Kalcijum	3,33	3,33
Proteini	6,67	3,33
Kalorije	3,33	3,33
Cena (din)	7,5	11,0

   Dnevni obrok treba da obezbedi najmanje 50 jedinica kalcijuma, 90 jedinica proteina i 66 kalorija. Odrediti dnevne količine mleka i žitarica, tako da se *minimizuju* troškovi sportske ishrane atletskog gimnastičara.

6. U pogonu proizvodnje čipova četiri tehničara (A, B, C i D) proizvode tri vrste čipova (Tip 1, Tip 2 i Tip 3). Mesečna prodaja je 120 čipova Tip 1, 75 čipova Tip 2 i do 50 čipova Tip 3. Tehničar A proizvodi samo čipove Tip 1 i 3, tehničar B samo čipove Tip 1 i 2, tehničar C čipove Tip 1 i 3, a tehničar D proizvodi samo čip Tip 2.

   Profit za čip Tip 1 je 600 din, za Tip 2 je 700 din, a za Tip 3 je 1.000 din.

   Uloženi rad (u radnim satima) svakog od tehničara je:

Proizvod	Tehničar

	A	B	C	D
Tip 1	2	2,5	-	-
Tip 2	-	3	-	3,5
Tip 3	3	-	4	-

Ako svaki tehničar mesečno efektivno radi najviše 120 sati, kako proizvođač čipova može da *maksimizuje* svoj ukupni profit?

7. Farma ima 130 hektara zemlje na kojoj seje dve kulture: kukuruz i soju. Troškovi setve 1 hektara kukuruza su 2.500 dinara, a soje 4.000 dinara. Svaki hektar kukuruza zahteva 3,6 m^3 skladišnog prostora i donosi profit od 2.500 dinara po hektaru, dok svaki hektar soje zahteva 1,5 m^3 za skladištenje, uz profit od 3.500 dinara po hektaru. Ukupan iznos prostora za skladištenje farme je 690 m^3.

   Za finansiranje setve farma je obezbedila finansijska sredstva u iznosu od 4.000.000 dinara.

   Koliko hektara svake kulture treba posejati kako bi se *maksimizovao* profit farme? Koliki iznos profita se može očekivati?

8. Otkupna cena 1 kg belog grožđa je 0,42 EUR i još 0.02 EUR po stepenu slatkoće, a cena 1 kg crnog grožđa je 0,4 EUR i još 0.02 EUR po stepenu slatkoće.

   Vinogradaru je sazrelo je 1800 kg belog i 1200 kg crnog grožđa. Svaka 4 dana grožđe postaje jedan stepen slađe. Vinogradar mora da organizuje berbu i transport u tri uzastopne subote, tako da u prvom terminu isporuči najviše 1500kg, u drugom 1100kg, a u trećem do 800 kg grožđa.

   Ako je slatkoća belog grožđa prve subote 17, a crnog 18,5 stepeni, koji je optimalan plan berbe, tako da zarada bude *najveća*?

9. Poznata grupa planira gostovanje u sportskoj dvorani, koja prima 5.000 gledalaca. Prema zahtevu organizatora, grupa može da proda najviše 3.000 opštih, odnosno do 4.000 povlašćenih studentskih ulaznica. Cena opšte ulaznice je 700, a studentske 400 dinara.

   Predviđena sredstva za reklamu nastupa su 300.000 din. Troškovi reklamiranja po jednoj ulaznici su 95 dinara po opštoj, odnosno 45 dinara po svakoj studentskoj ulaznici.

   Koliki je *najveći* profit koji se može ostvariti i koliko studentskih ulaznica je neophodno prodati?

10. Banka ima na raspolaganju 650.000 EUR koje treba da raspodeli na ulaganja u obveznice, stambene kredite, auto kredite i gotovinske pozajmice. Od obveznica se očekuje prinos od 10%, od hipoteka 8,5%, auto

kredita 9,5%, a gotovinskih kredita 12,5%. Da bi smanjila rizik ulaganja, banka želi da ograniči gotovinske pozajmice na ne više od 25% od ukupnog portfolija. Osim toga, banka želi da osigura da se više ulaže u hipoteke nego gotovinske kredite, kao i da ulaganja u obveznice budu veća nego u gotovinske kredite.

Kako banka treba da rasporedi raspoloživa sredstva da ostvari *najveći* mogući povrat uloženog novca?

4 Računarska podrška odlučivanju: koncept i tehnologija sistema

1. Koncept i arhitektura sistema za podršku odlučivanju
2. Komponente sistema za podršku odlučivanju
3. Klasifikacija sistema za odršku odlučivanju
4. Primeri sistema za podršku odlučivanju

U ovom poglavlju izlažu se osnovni koncepti sistema za podršku odlučivanju i objašnjava struktura i funkcionisanje njihovih standardnih komponenti. Daju se osnovne klasifikacije i primeri realizovanih sistema za podršku odlučivanju u različitim oblastima primene.

4.1 Koncept i arhitektura sistema za podršku odlučivanju

Sistemi za podršku odlučivanju su usmereni na podršku donošenju odluka, po čemu se razlikuju od standardnih informacionih sistema, kao i drugih sistema za obradu podataka različite namene.

Sistem za podršku odlučivanju je sistem zasnovan na *modelima* ili *znanju*, čija je namena podrška menadžerima u donošenju odluka u delimično strukturiranim i nestrukturiranim situacijama.

Podrška se obično realizuje pomoću modeliranja i primene modela za analizu mogućih rešenja. Osnovna namena sistema nije da zameni ljude u procesu donošenja odluka, već da poboljša njihovu sposobnost donošenja odluka.

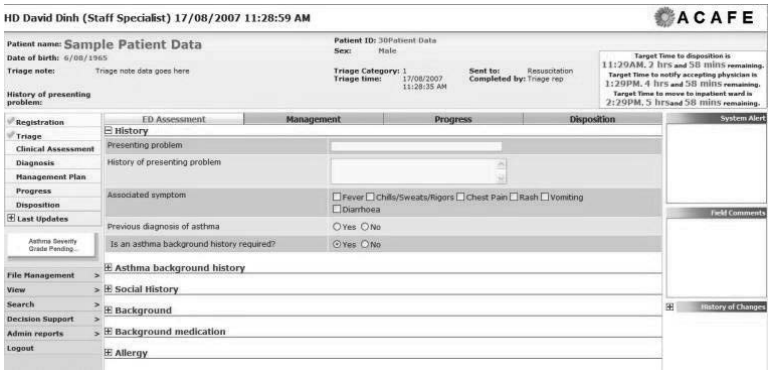
Ilustracija: podrška procesima u zdravstvu

U zdravstvu postoje u elektronskoj formi podaci o pacijentima (e-karton) i procesu lečenja, npr. zakazivanje, dijagnostika, terapija, nadzor i sl.

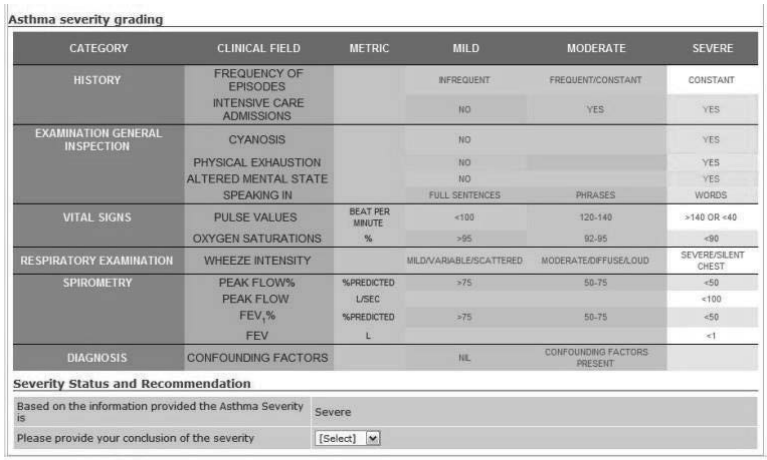
Sistem otvorenog koda ACAFE (*A Computerized Advancement For Emergency*) je namenjen podršci lekarima hitne pomoći u donošenju odluka prilikom hitnih intervencija na terenu [71]. Na slici je ekran za prijavu korisnika sistema:



Sistem omogućava pristup i uvid u medicinske podatke o pacijentima, kao klasičan informacioni sistem:



Podrška odlučivanju se vrši na više načina, npr. kroz ocenu stepena ozbiljnosti unesenih simptoma za konkretnog pacijenta i aktiviranje odgovarajućih upozorenja istaknutih drugom bojom pozadine:



Osnovna svojstva sistema za podršku odlučivanju

Sistemi za podršku odlučivanju obezbeđuju podršku u rešavanju delimično strukturiranih i nestrukturiranih problema na svim nivoima individualnog i grupnog odlučivanja, direktnom ili sekvencijalnom odlučivanju u svim fazama odlučivanja.

Osnovna svojstva ovih sistema su:

- obezbeđuju podršku različitim načinima odlučivanja;
- velika prilagodljivost, interaktivnost i jednostavnost upotrebe;

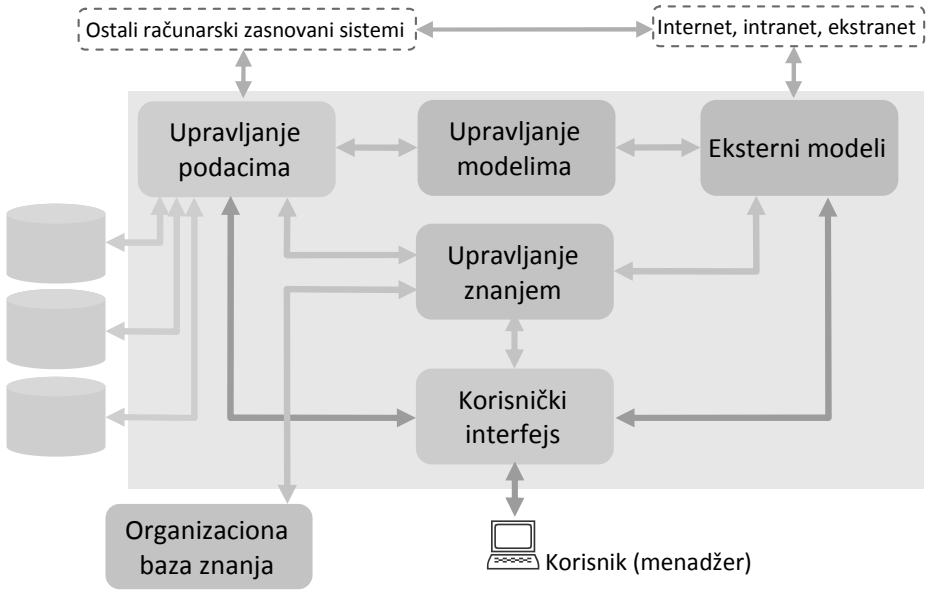
- pre svega poboljšavaju efektivnost (tačnost, pravovremenost, kvalitet), a ne efikasnost odlučivanja (cena odluke);
- omogućavaju samostalni razvoj jednostavnijih sistema i krajnjim korisnicima, pošto omogućavaju modeliranje i eksperimentisanje;
- omogućavaju pristup različitim podacima u bazama podataka, različitim navigacionim i multimedijskim informacijama.

4.2 Komponente sistema za podršku odlučivanju

Osnovne komponente sistema za podršku odlučivanju su podsistemi, Sl. 38 [1]:

- za upravljanje podacima (*Database Management System*, DBMS),
- za upravljanje modelima (*Model Base Management System*, MBMS),
- za upravljanje znanjem (*Knowledge Base Management System*, KBMS) i
- korisnički interfejs.

Pojedini sistemi imaju samo neke od ovih komponenti. Podsistem za upravljanje znanjem imaju sistemi za podršku odlučivanju zasnovani na znanju.



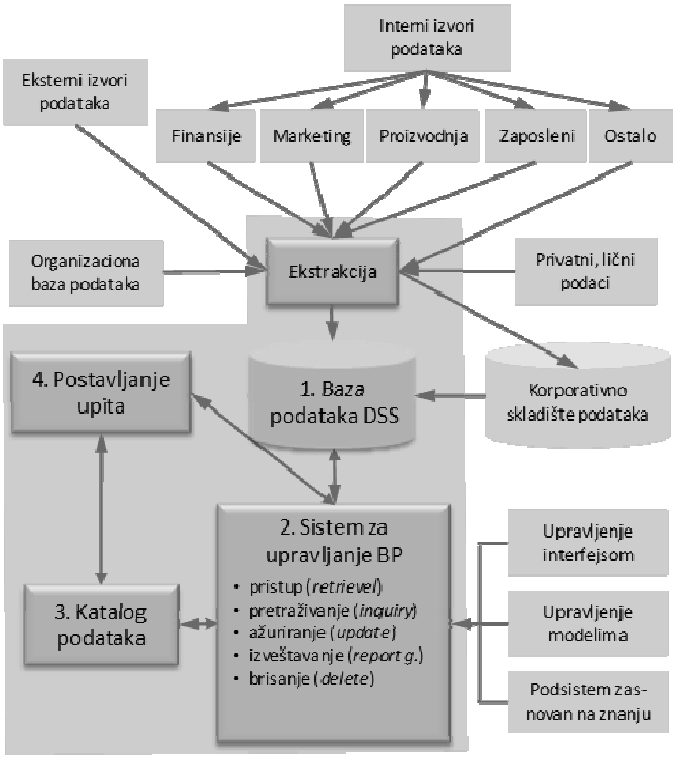
Sl. 38. Struktura sistema za podršku odlučivanju

U ovoj tački će se ukratko prikazati samo osnovna struktura i opšte funkcije komponenti sistema za podršku odlučivanju, dok će se teorijski i praktični aspekti razvoja i implementacije više kategorija sistema za podršku odlučivanju razmotriti u narednim poglavljima.

4.2.1 Podsistem za upravljanje podacima

Osnovne funkcije sistema za upravljanje bazama podataka (DBMS) su:
- izdvajanje (ekstrakcija) podataka,
- upravljanje podacima i relacijama,
- ažuriranje podataka (*add*, *delete*, *edit*, *change*),
- pristup podacima putem pretraživanja,
- realizacija svih vrsta upita,
- katalog podataka.

Komponente podsistema za upravljanje podacima u sistemu za podršku odlučivanju su, Sl. 39:
- baza podataka (*Database*),
- sistem za upravljanje bazama podataka (*DBMS*),
- katalog podataka (*Data directory*),
- sistem ili mehanizam za postavljanje upita (*Query facility*).



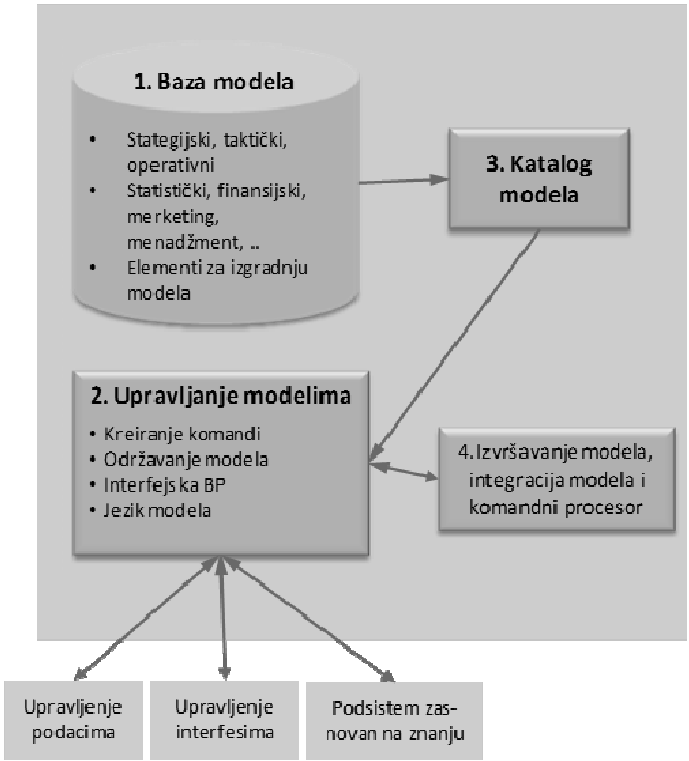
Sl. 39. Struktura podsistema za upravljanje podacima

4.2.2 Podsistem za upravljanje modelima

Osnovne komponente podsistema za upravljanje modelima (*model management*) i njihov međusobni odnos prikazani su na Sl. 40.

Podsistem za upravljanje modelima čine:

- baza modela,
- komponenta za upravljanje bazom modela (MBMS),
- jezik modela,
- katalog modela,
- komponenta za izvršavanje modela, integraciju modela i procesiranje komandi.



Sl. 40. Struktura podsistema za upravljanje modelima

Prena nameni, u sistemima za podršku odlučivanju koristi se više kategorija modela:

- *Strategijski*, za podršku donošenju odluka najvišeg nivoa menadžmenta organizacije;
- *Taktički*, za podršku odlučivanju o alokaciji resursa na srednjem nivou menadžmenta;
- *Operativni*, za podršku svakodnevnim aktivnostima organizacije;
- *Analitički*, koji se koriste za analizu podataka.

Komponenta za upravljanje bazom modela

Funkcije komponente za upravljanje bazom modela (*Model Base Management System*, MBMS) su:

- kreiranje modela,
- ažuriranje modela,
- manipulisanje podacima modela,
- generisanje novih procedura.

Katalog modela (*model directory*) sadrži same modele i druge definicije, neophodne za njihovo kreiranje, ažuriranje i izvršavanje.

4.2.3 Podsistem za upravljanje znanjem

Primena računara u kognitivnim zadacima predviđana je na samom početku njihovog razvoja. Naučna disciplina koja se bavi problemom predstavljanja i upotrebe znanja za podršku različitim aktivnostima, uključujući podršku donošenju odluka, naziva se veštačka inteligencija (*Artificial Intelligence*, AI).

Posebna kategorija softvera, koji se razvija od šezdesetih godina dvadesetog veka, bavi se problemom podrške odlučivanju u nestrukturiranim problemima. Osnovna ideja je da logičko rasuđivanje nije dovoljno za rešavanje ovakvih problema, već je neophodno specifično znanje o samom problemu i njegovom rešavanju, koje poseduju eksperti na određenom području.

Fragmentarno i nekompletno znanje eksperata koristi se u posebnoj kategoriji sistema, koji se nazivaju ekspertni sistemi (*expert systems*). Osnovna namena ekspertnih sistema je pružanje podrške u donošenju odluka.

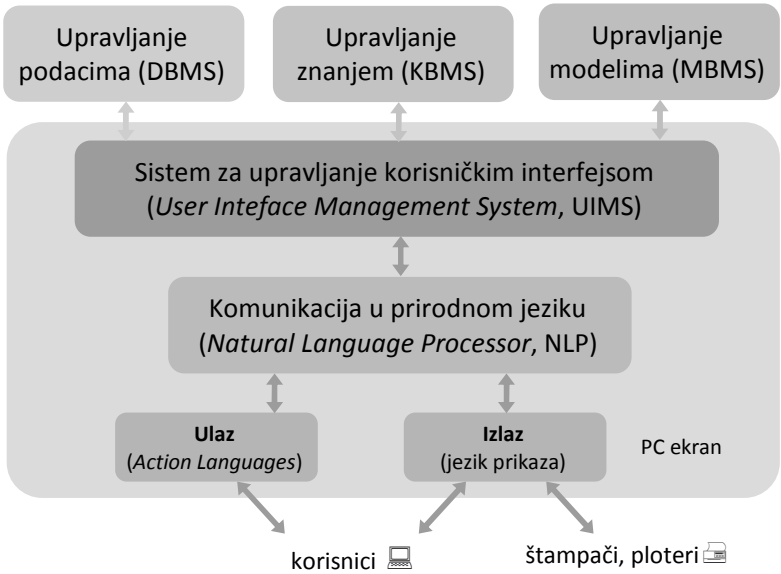
Više o ekspertnim sistemima i drugim kategorijama softvera čiji rad se zaniva na predstavljanju i upotrebi znanja iznosi se u poglavljima posvećenim inteligentnim sistemima.

4.2.4 Podsistem korisničkog interfejsa

Podsistem korisničkog interfejsa je veoma važna komponeta, jer korisnici računarskih sistema doživljavaju korisnički interfejs kao sam sistem. Način komunikacije u velikoj meri određuje mogućnost i kvalitet podrške odlučivanju.

Grafički interfejsi i multimedijska komunikacija poboljšavaju obim i kvalitet komunikacije čoveka i računara. Struktura podsistema korisničkog interfejsa prikazana je na Sl. 41.

Primeri različitih načina komunikacije i odgovarajućih interfejsa koje koriste sistemi za podršku odlučivanju su komunikacija glasom i interefjsi koji prepoznaju govor (*speech recognition*) i vrše sintezu govora (*speech synthesis*), komunikacija gestovima i interfejsi koji prepoznaju pokrete snimljene kamerom, panel displeji i taktilni interfejsi.



Sl. 41. Struktura podsistema korisničkog interfejsa

Komunikacija u prirodnom jeziku

Komunikacija u prirodnom jeziku je prirodan način ljudske komunikacije i predstavlja obrazac prema kome su kreirani deklarativni programski jezici, kao što je jezik SQL. Računarska komunikacija u prirodnom jeziku je kontekstno zavisna, specifična za pojedine jezike i grupe jezika, tako da ne postoji univerzalni sistem za komunikaciju s računarom. Sistemi za komunikaciju u prirodnom jeziku su razvijeni uglavnom za engleski i neke druge svetske jezike, dok podrška za veliki broj jezika ne postoji, veoma je ograničena ili se tek razvija.

Komunikacija glasom

Tekstualna komunikacija je ekvivalent govora. Govorna komunikacija s računarskim sistemom je moguća korišćenjem metoda prepoznavanja i sinteze govora.

Prepoznavanje govora (*speech recognition*) prevodi govorni signal mikrofona u tekst, a programi za sintezu govora prevode tekst u zvučni signal koji se reprodukuje kao govor, pri čemu je sinteza govora jednostavniji postupak, pa je ugrađen u brojne aplikativne programe, npr Microsoft *Windows*, *Google Translate*.

Podrška i kvalitet komunikacije zavise od jezika. Najrazvijenija je podrška za engleski i nekoliko drugih vodećih svetskih jezika, dok se za ostale tek pojavljuju parcijalna rešenja.

Prepoznavanje govora ne treba poistovećivati s razumevanjem govora (*speech understanding*), gde je cilj razumevanje smisla izgovorenog.

Primer: Programi za ćaskanje

Zanimljiva kategorija softvera, koja koristi komunikaciju u prirodnom jeziku su programi za ćaskanje (*chatterbots*). Nastali su u oblasti veštačke inteligencije, kao pokušaj kreiranja inteligentnih softverskih agenata, koji mogu da komuniciraju u prirodnom jeziku. Koriste se za zabavu i neke rutinske zadatke, kao što je automatsko odgovaranje na telefonske pozive i davanje informacija.



Sl. 42. Primer komunikacije u prirodnom jeziku s inteligentnim agentom

Programi za ćaskanje omogućavaju razgovor sa veštačkim sagovornikom, koji je često podržan animacijom i sintezom govora, Sl. 42 [74].

4.3 Klasifikacija sistema za podršku odlučivanju

Postoji više sličnih klasifikacija sistema za podršku odlučivanju, od kojih je najsažetija podela na [1]:

- komunikaciono orijentisane i grupne sisteme (*GSS*),
- sisteme zasnovane na podacima (*data-driven*),
- sisteme zasnovane na dokumentima (*document-driven*),
- sisteme zasnovane na znanju (*knowledge-driven*),
- sisteme zasnovane na modelima (*model-driven*).

Slična klasifikacija, koja samo posebno izdvaja pojedine kategorije sistema, koji su česti u praksi je [1]:

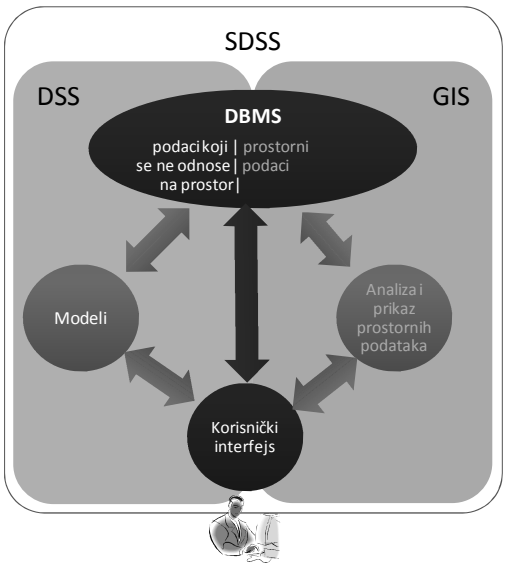
- tekstualno orijentisani sistemi (*text-oriented*),
- sistemi zasnovani na podacima (*database-oriented*),
- sistemi zasnovani na elektronskim tabelama (*spreadsheet-oriented*),
- sistemi zasnovani na modelima (*solver-oriented*),
- sistemi zasnovani na pravilima (*rule-oriented*),
- hibridni sistemi (*compound*), koji koriste više metoda istovremeno.

Najčešći su u praksi kombinovani ili hibridni sistemi, pri čemu je neki od metoda realizacije dominantan.

4.4 Prostorni sistemi za podršku odlučivanju

Prostorni sistemi za podršku odlučivanju (*Spatial DSS*, SDSS) su kategorija sistema posebne arhitekture, koja kombinuje tradicionalne sisteme za podršku odlučivanju i komponente geografskog informacionog sistema (GIS), Sl. 43:

- tradicionalni sistem za podršku odlučivanju se bavi modelima i prikupljanjem, pronalažnjem, upravljanjem i analizom podataka koji nisu prostorni;
- geografski informacioni sistem obezbeđuje prikupljanje, skladištenje, upravljanje i kartografski prikaz prostornih podataka.



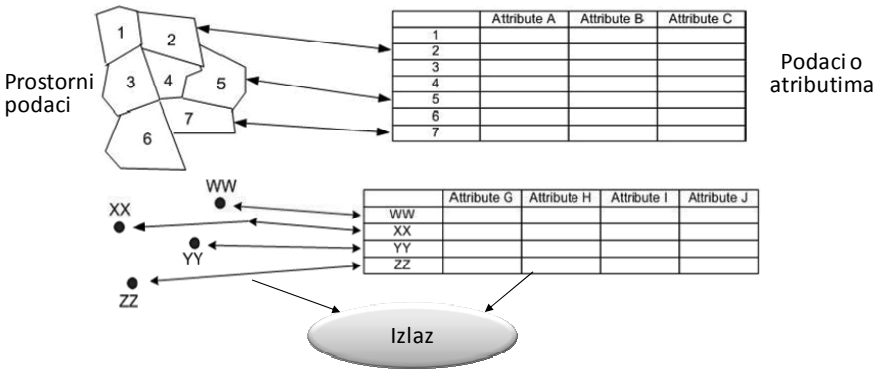
Sl. 43. Struktura prostornog DSS

Prostorni sistemi za podršku odlučivanju predstavljaju velike skupove podataka pomoću mapa. Omogućavaju lakši pristup, prikaz i analize podataka koji imaju

geografski sadržaj i značenje, kao što su npr. demografske analize klijenata, politički obrasci glasanja po izbornim jedinicama, kriminološke analize i prikazi [1], [23]. Osnovna ideja ovih sistema je proširenje mogućnosti standardnih geografskih informacionih sistema u rešavanju kompleksnih, nestrukturiranih, prostorno orijentisanih problema. U praksi se za ovu klasu sistema koriste različiti nazivi i oznake, npr. SDSS, grupni SDSS, DSS životne sredine, prostorni sistemi zasnovani na ekspertnom znanju, kolaborativni GIS (*Public participation GIS*, PPGIS) i sl.

Geografski informacioni sistemi (GIS) su posebno projektovani za rad sa prostornim informacijama, kako bi unapredili donošenje odluka. Omogućavaju obradu različitih vrsta geografskih podataka (mapa), kao i drugih važnih podataka vezanih za geografsku lokaciju: stanovništvo, prodaja, vreme, saobraćaj, kriminal i sl.

Mnoge odluke zasnivaju na geografskim lokacijama, npr. gde otvoriti maloprodajne objekte, gde kupiti kuću ili stan, gde se može više zaraditi. Vizuelni prikaz podataka pomoću mapa je efikasniji i razumljiviji od ostalih oblika. Korisnik kreira mape prema svojim potrebama (krive - poligoni, linije) i unosi podatke. GIS vizualizuje podatke po slojevima (*layers*), Sl. 44.



Sl. 44. Funkcionisanje GIS

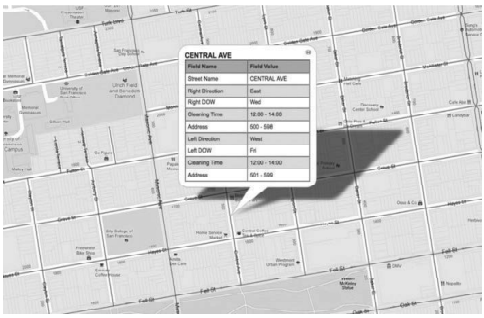
Neki primeri GIS aplikacija su dinamičke mape, koje omogućavaju različite prostorne analize:

- Google Maps http://maps.google.com/
- Yahoo!Maps http://maps.yahoo.com/
- MapQuest http://www.mapquest.com/
- DTC Wildfire Incident Viewer http://demo.dtsagile.com/wildfire/
- Microsoft *Streets and Trip* (map&travel planning; slojevi restorana, hotela);
- Via Michelin (besplatan Web map&travel planning).

Primer: Mape čišćenja ulica i plan dozvoljenog parkiranja u San Francisku

Primer posebne GIS aplikacije namenjene gradskom i regionalnom planiranju, koja ima aplikaciju za podršku odlučivanju, je *San Francisco Enterprise GIS* (http://www.ci.sf.ca.us/site/gis_index.asp).

Sistem rešava problem izbora mesta za parkiranje, kao jedan od važnih praktičnih problema funkcionisanja grada, jer postoji više od 50 ruta čišćenja ulica, tokom koga je zabranjeno parkiranje u okolnim ulicama.



Kazne i nezadovoljstvo građana su dovele do kreiranja posebnog rešenja, koje koristi prostorni sistem za podršku odlučivanju o parkiranju i dostupan je građanima preko mobilnih uređaja. Izrađene su javne mape dnevnog rasporeda čišćenja i parkiranja, tako da građani pravilno da izaberu parking i izbegnu kazne.

Postoje i druge kategorije softvera koje se zasnivaju na GIS rešenjima, kao što je npr. računarska igra *SimCity*.

4.5 Primeri sistema za podršku odlučivanju

Kratko će se nabrojati primeri nekih kategorija komercijalnih Web sistema i nekih edukativnih sistema, kao što su školski primeri ekspertnih sistema.

- DSS za kupovinu

  mySimon - pomoć pri kupovini (*shopping agent*, probati "led tv")

- DSS za odlučivanje o transportu

  FedEx - međunarodni transport (kalkulator, praćenje pošiljki)

- Prostorni DSS

  NASA REASoN - sistem za podršku odlučivanju o alokaciji resursa radi zaštite granica SAD (*border Spatial decision support system*, BSDSS)

- Demonstracije ekspertnih sistema
  Grad Advisor - izbor poslediplomskog školovanja
  Audit Reporting - mišljenje revizora
  Processing Loan Applications - obrada kreditnih zahteva
  easydiagnosis.com/ - medical diagnostics examples
- edukativni primeri DSS
  www.uky.edu/BusinessEconomics/dsskba/instmat.htm

Pitanja za ponavljanje

1. Šta čini arhitekturu sistema za podršku odlučivanju?
2. Koje su osnovne komponente podsistema za upravljanje modelima?
3. Koje su osnovne funkcije podsistema za upravljanje podacima u sistemu za podršku odlučivanju?
4. Navedite klase modela koje koriste sistemi za podršku odlučivanju.
5. Šta je *chatterbot*?
6. Za koje oblike komunikacije je pogodan tekstualno orijentisan interfejs računarskog sistema?
7. Koje vreste intefejsa su pogodne za komunikaciju u prirodnom jeziku?
8. Navedite bar tri načina komunikacije koji se koriste u korisničkim interfejsima sistema za podršku odlučivanju.

5 Poslovna inteligencija, skladištenje podataka i otkrivanje znanja

1. Osnovni pojmovi
2. Poslovna inteligencija
3. Skladištenje podataka
4. Istraživanje podataka
5. Upravljanje znanjem
6. Primeri sistema poslovne inteligencije

U ovom poglavlju se izlažu savremeni metodi modeliranja i empirijske analize sistema, koji se koriste za podršku odlučivanju i izgradnju sistema za podršku odlučivanju. Izlaganje je usmereno na metode sladištenja podataka i otkrivanja znanja, a daje se i osvrt na problem upravljanja znanjem.

5.1 Osnovni pojmovi

U složenim organizacijama neprekidno se donose različite odluke, koje imaju kratkoročne i dugoročne efekte i utiču na veliki broj zaposlenih na različitim nivoima hijerarhije organizacije. Sposobnost individualnog i grupnog odlučivanja ljudi koji rade sa znanjem primarni je faktor uspešnosti organizacije [10]. Pri tome se za donošenje odluka koriste intuitivne metodologije, koje se zasnivaju na znanju, iskustvu i raspoloživim informacijama.

Složenost i dinamika savremenog odlučivanja zahtevaju primenu efikasnijih metoda, koji se zasnivaju na analitičkim metodima i matematčkim modelima. Sistemi poslovne inteligencije treba da omoguće efikasno i pravovremeno donošenje odluka.

Poslovna inteligencija se može definisati kao skup matematičkih modela i analitičkih metoda, koje koriste raspoložive podatke za generisanje informacija i znanja korisnih za složene procese donošenja odluka [10].

Osnovna arhitektura sistema poslovne inteligencija se sastoji od [10]:

- izvora podataka (*Data sources*)
- skladišta podataka (*Data Warehouses, Data Marts*)
- metoda poslovne inteligencije (*Business Intelligence methodologies*)

Izvori podataka mogu biti strukturirani, kao u operativnim informacionim sistemima, ali i nestrukturirani, kao npr. e-mail poruke i eksterni podaci.

Skladište podataka je skup podataka proizvedenih radi podrške odlučivanju. Ekstrakcija, punjenje i transformacija podataka (ETL) iz različitih izvora u novu

bazu podataka, stvaraju strukturu namenjenu podršci analiza poslovne inteligencije.

Metodi poslovne inteligencije za podršku odlučivanju obuhvataju, npr.
- interaktivne višedimenzionalne analize;
- analize podataka;
- analize vremenskih serija;
- induktivno učenje modela istraživanja podataka;
- optimizacione modele.

5.2 Poslovna inteligencija

U ovom materijalu će se koristiti definicija poslovne inteligencije koja ističe najvažnije komponente sistema poslovne inteligencije:

Poslovna inteligencija (*Business Intelligence*, BI) je naziv za arhitekturu i skup integrisanih alata, aplikacija i baza podataka, koji se koriste u analizi poslovnih podataka za podršku odlučivanju [1].

Komponente sistema poslovne inteligencije su:

- skladište podataka (*Data Warehouse, DW*);
- skup korisničkih alata za kreiranje upita i izveštaja i njihovu analizu i vizualizaciju (*Business Analytics, BA*);
- metodi za otkrivanje netrivijalnih relacija u podacima (*data mining* i *Web mining*);
- metodi i alati za upravljanje performansama poslovanja (*Business Performance Management, BPA*);

Međusobni odnos komponenti sistema poslovne inteligencije prikazan je na Sl. 45 [10].



Sl. 45 Osnovne komponente sistema poslovne inteligencije

Alati za analizu podataka (*data exploration*) služe za pasivne analize i uključuju metode postavljanja upita, sastavljanja izveštaja i statističke metode. Metodi su pasivni, jer podrazumevaju da donosioci odluka unapred postavljaju hipoteze i koriste analitičke metode za njihovu proveru.

Istraživanje podataka (*Data Mining*) se sastoji od skupa aktivnih metoda analize čija je osnovna namena izdvajanje informacija i znanja iz podataka. Sastoji se od matematičkih modela i metoda mašinskog učenja, prepoznavanja uzoraka i drugih metoda istraživanja podataka. Donosilac odluke ne mora unapred da postavlja hipoteze, a podrška odlučivanju se ostvaruje kroz povećanje njegovog znanja.

Modeli optimizacije omogućavaju određivanje najbolje alternative pre svega u različitim strukturiranim problemima, npr. u marketingu ili logistici.

Donošenje konkretne odluke ili izbor alternative se oslanja na metode poslovne inteligencije, ali predstavlja odgovornost donosioca odluke, koji u proces može da uključi neformalne i nestrukturirane informacije za prilagođavanje predloga odluke koji se dobija iz matematičkog modela.

Proces analize u poslovnoj inteligenciji zavisi od problema, donosilaca odluka i raspoloživih metoda, ali se odvija ponavljanjem aktivnosti analize problema, razumevanja problema, donošenja odluke i evaluacije rešenja [10]:

1. Analiza problema obuhvata razumevanje, opis i uočavanje važnih faktora problema, pri čemu se mogu koristiti različiti interaktivni alati za višedimenzionalne analize.
2. Razumevanje problema pretvara informacije iz analize problema u znanje, pri čemu se koristi intuicija, iskustvo i nestrukturirane informacije, kao i metodi induktivnog mašinskog učenja.
3. Donošenje odluke je faza u kojoj se znanje i rezultati faze razumevanja pretvaraju i odluke, čija posledica su određene akcije. Metodi poslovne inteligencije omogućavaju da se prethodne faze realizuju brzo, tako da se mogu donositi pravovremene i kvalitetne odluke.
4. Evaluacija se bavi merenjem performansi i evaluacijom odluka na osnovu skupa finansijskih pokazatelja i indikatora koje definišu različiti delovi organizacije.

5.3 Skladištenje podataka

Skladište podataka je skup podataka proizvedenih radi podrške odlučivanju. Izdvojeni tekući i istorijski podaci, koji su potencijalno od interesa za menadžment kompanije, struktuiraju se na pogodan način za upotrebu u različitim analizama za potrebe donošenja odluka.

Skladište podataka (*Data Warehouse*) je integrisana, pojmovno orijentisana, vremenski promenljiva i neizbrisiva kolekcija podataka, koja pruža podršku odlučivanju [1].

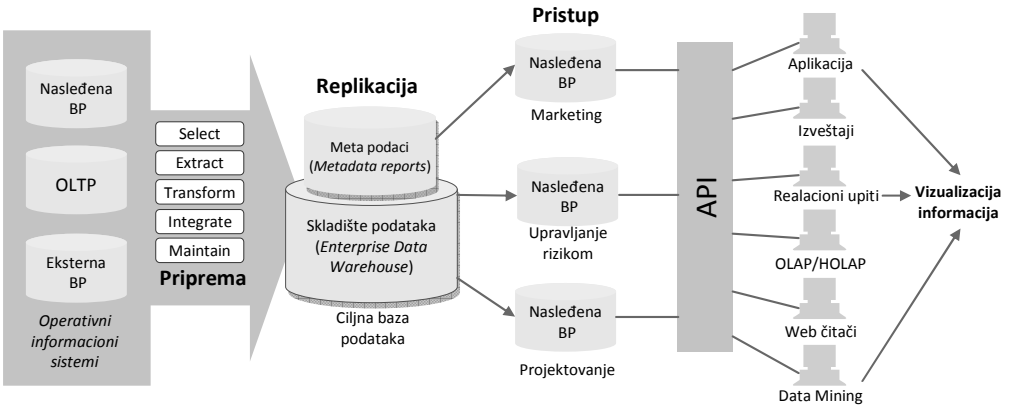
Skladišta podataka su neophodna za interaktivne analitičke obrade (OLAP) koje omogućavaju korisniku udobnije, manje formalne načine postavljanja složenih upita. Predstavljaju analitičke baze podataka, koje se razvijaju nezavisno od operativnih baza podataka informacionih sistema, jer su ciljevi njihove izgradnje različiti, što bi moglo da dovede do neefikasnosti ako se ne prilagode posebnim zahtevima. Koriste se i za smeštanje novih, proizvedenih informacija, koje se dobijaju iz više izvora i baza podataka.

Skladišta podataka su u savremenim organizacijama postala neophodna za održavanje tržišne prednosti, koriste se za izgradnju brendova, održavanje lojalnosti klijenata i druge poslovne primene.

Prema nameni, postoji više vrsta skladišta podataka [1]:

- *Enterprise Data Warehouses* (EDW) - velika skladišta podataka, namenjena celoj organizaciji;
- *Data Mart* - mali podskup skladišta posvećen nekoj tematici i namenjen određenoj grupi ljudi, npr. sektoru kompanije;
- *Operational Data Store* (ODS) - dinamička, privremena skladišta, za brzo donošenje odluka.

Na Sl. 46 je prikaz strukture i komponeti procesa skladištenja podataka [1].



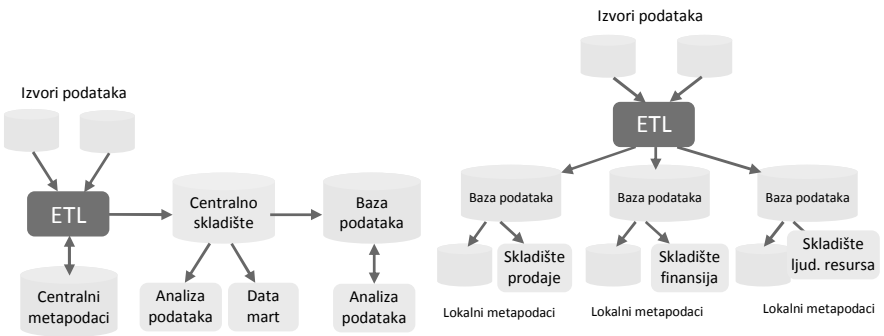
Sl. 46. Struktura i komponente skladišta podataka

Osnovne komponente procesa skladištenja podataka su izvori podataka (*data sources*), koji mogu biti transakcione baze, datoteke, eksterni izvori, softver za pripremu (ETL), baza podataka, metapodaci i *middleware* (SQL ili okruženje za postavljanje upita).

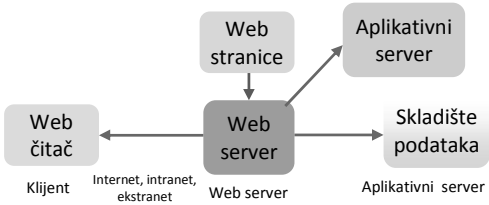
Arhitektura skladišta

Prema tome kako su raspoređeni podaci i softver skladišta, deo softvera skladišta za prikupljanje podataka (ekstrakciju, konsolidaciju, sumiranje i punjenje skladišta) i klijentski softver, skladište može imati dvoslojnu ili troslojnu arhitekturu.

Prema organizaciji podataka, postoje različite varijante arhitekture skladištenja, od centralizovanih integrisanih skladišta do decentralizovanih rešenja (*Data Marts*), gde se samo dele različiti pogledi na podatke, Sl. 47.



a) centralizovano b) decentralizovano (*Data Mart*)



c) Web

Sl. 47. Različite arhitekture skladištenja

Integracija podataka

Integracija podataka obuhvata pripremu podataka za upotrebu od strane drugih alata za skladištenje: pristup različitim izvorima podataka (*access*), objedinjavanje podataka iz različitih internih i eksternih izvora i sistema (*federation*), kao i izdvajanje promena (*change capture*).

Tehnologije koje se koriste u integraciji podataka su *integracija aplikacija* (*Enterprise Application Integration*, EAI, koja se realizuje korišćenjem servisno orijentisane arhitekture, SOA), *integracija informacija* (*Enterprise Information Integration*, EII) i tehnologija izdvajanja, transformacije i punjenja skladišta podataka (*Extract, Transformation and Load*, ETL).

Proces ekstrakcije, transformacije i punjenja (ETL)

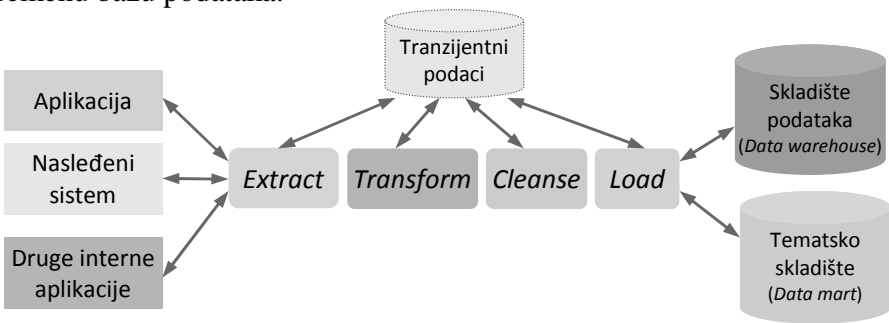
Priprema podataka za analize obuhvata izdvajanje podataka, njihovu transformaciju i punjenje u kreirano skladište podataka, Sl. 48.

Ekstrakcija predstavlja proces čitanja više baza podataka, gde spadaju transakcione baze, fajlovi, elektronske tabele i lične baze podataka.

Transformacija predstavlja promenu oblika podataka, koja obuhvata različita ujednačavanja tipova i formata ulaznih podataka radi prilagođavanja ciljnom tipu i formatu podatka u skladištu.

Punjenje (*load*) je proces smeštanja izdvojenih i transformisanih podataka u skladište.

Zbog računske i prostorne složenosti operacija, proces koristi pomoćnu, privremenu bazu podataka.



Sl. 48. Proces ekstrakcije, transformacije i punjenja (ETL proces)

Poslovna pravila se smeštaju centralizovano, kao metapodaci.

Za proces pripreme podataka koriste se namenski alati (*Microsoft*, *Oracle*, *IBM*, *Informatica*, *Tibco*) ili se razvijaju posebni programi.

5.3.1 Razvoj skladišta podataka

Primer efekata razvoja i uvođenja skladišta podataka u složenoj organizaciji je slučaj poznate američke avioprevozničke kompanije, koja je uvođenje skladišta podataka upotrebila kao važnu meru za poboljšanje poslovanja i izlazak iz finansijskih problema [1].

Primer: Efekti praktične upotrebe skladišta podataka

Američki avioprevoznik, sedmi u svetu, koji je dnevno imao 2.300 letova na 227 destinacija, našao se u finansijskim problemima. Analiza je pokazala da je osnovni problem u organizaciji i zadovoljenju potreba putnika. Informacioni sistem je bio neefikasan, sastavljen od više različitih baza podataka i nasleđenih zastarelih sistema, sa sporim odgovorima na upite.

Izgrađeno je integrisano skladište podataka kompanije (EDW), koje je obezbeđivalo na jednom mestu u realnom vremenu informacije o putnicima, osoblju, bezbednosti i procesu letenja.

Osim redukcije troškova, eliminisane su i različite zloupotrebe, koje se nisu mogle otkriti bez integracije informacija iz različitih izvora, njihovog sumiranja i analitičke obrade.

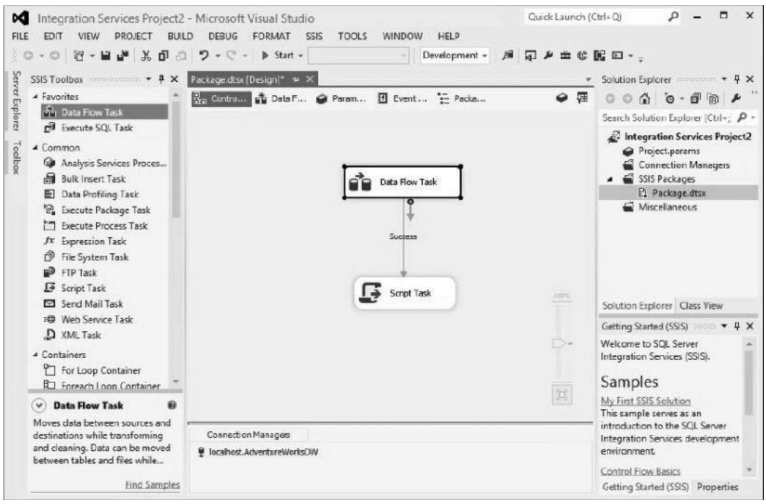
Direktni finansijski efekti investicije od 30 miliona dolara (u toku 6 godina) su povećanje prihoda u iznosu od 500 miliona dolara [1].

Razvoj skladišta se može realizovati kao kombinacija dva različita pristupa: centralizovani (EDW) i decentralizovani pristup (*Data Mart*).

Centralizovani pristup obezbeđuje jedinstven pogled na celu organizaciju, a za razvoj složene strukture skladišta koriste se metodi i alati za razvoj baza podataka (modeliranje podataka, inkrementalni razvoj).

Decentralizovani pristup se sastoji u planiranju celine i postupnoj izgradnji skladišta kroz više odvojenih tematskih skladišta koji se organizuju prema problematici pojedinih organizacionih celina. Često je decentralizovana izgradnja prvi korak ka razvoju centralizovanog skladišta.

Implementacija skladišta podataka je poseban projekt, jer postoji veliki broj različitih pristupa i niz faktora i kriterijuma koje treba zadovoljiti. Osnovni alati za integraciju podataka su deo softvera poznatih proizvođača DBMS/BI alata, kao što je npr. ETL alat Microsoft SSIS (*SQL Server Integration Services*) koji je deo Microsoft BI platforme, Sl. 49 [90].



Sl. 49. SSIS - SQL Server Integration Services

Sistem SSIS se koristi putem okruženja za istraživanje podataka *SQL Server Data Tools* (SSDT), ranije *Business Intelligence Development Studi*o (BIDS). Okruženje SSDT više nije deo softvera SQL Server 2014, već se instalira posebno, kao dodatak okruženja za razvoj softvera *Visual Studio* [91].

Skladištenje u realnom vremenu

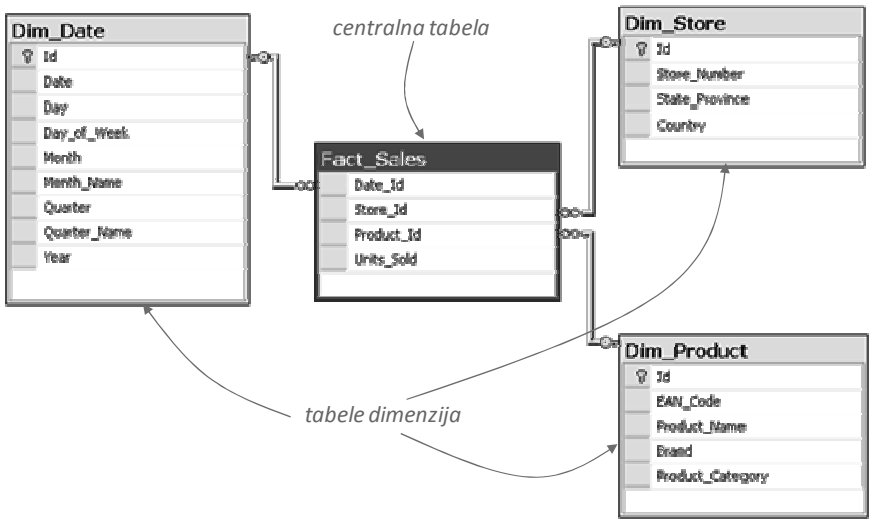
Performanse skladišta podataka opadaju porastom njegove veličine, koja kod savremenih skladišta podataka raste eksponencijalno, tako da u praksi već dostižu veličinu reda petabajta. *Skalabilnost* skladišta podrazumeva svojstvo da vreme pristupa linearno zavisi od veličine skladišta.

Period ažuriranja zavisi od potreba za ažurnim podacima u analizama. Tradicionalna skladišta podataka se ažuriraju periodično (npr. nedeljno). No, važni aspekti poslovanja mogu da zahtevaju donošenje odluka u realnom vremenu. Za potrebe korisnika koji direktno komuniciraju s kupcima i dobavljačima, EDW su evoluirali u *real time data warehousing* (RTDW) ili *active warehousing* (ADW).

Skladištenje u realnom vremenu omogućava najvišu ažurnost podataka i upotrebu skladišta i na najnižem, "taktičkom" nivou odlučivanja.

5.3.2 Struktura skladišta (zvezda)

Modeliranje dimenzija je sistem pristupa podacima velikog obima u kome se model operativne baze podataka transformiše u strukturu koja se može efikasno koristiti u analitičkim obradama. Osnovni način implementacije modela dimenzija je struktura zvezde (*star schema*), čiji primer je prikazan na Sl. 50.



Sl. 50. Pprimer strukture zvezde

Struktura zvezde se sastoji od:

- centralne tabele (*facts table*), u kojoj su posmatrani poslovni podaci i njihovi atributi, koji mogu biti analitički (mere performansi i metrike), deskriptivni (*query reporting*) ili ključevi (veze ka tabelama dimenzija);
- tabele dimenzija (*dimension table*), koji služe za klasifikaciju i agregaciju informacija o podacima u centralnoj tabeli.

Osim osnovne strukture zvezde, u praksi se koriste složeniji modeli dimenzija, kao što su višestruka zvezda (*multi star*) i struktura pahuljice (*snowflake*), gde tabele dimenzija imaju svoje tabele dimenzija.

Nivo detaljnosti skladištenja infomacija ili *granulacija* definiše mogućnosti buduće analitičke obrade. Optimizuje se da bi se postigle dobre performanse obrade uz razumnu veličinu skladišta podataka.

5.3.3 Softver za skladištenje

Softver za skladištenje je najčešće deo softvera za upravljanje bazama podataka. Poznati proizvođači namenskog softvera za skladištenje podataka su:

- Computer Associates (www.ca.com),
- Hyperion Solutions (www.hyperion.com),
- IBM (www.ibm.com),
- Informatica (www.informatica.com),
- Microsoft (www.microsoft.com),
- Oracle, PeopleSoft, Siebel (www.oracle.com),
- SAS Institute (www.sas.com),
- Sybase (www.sybase.com),
- Teradata (www.teradata.com).

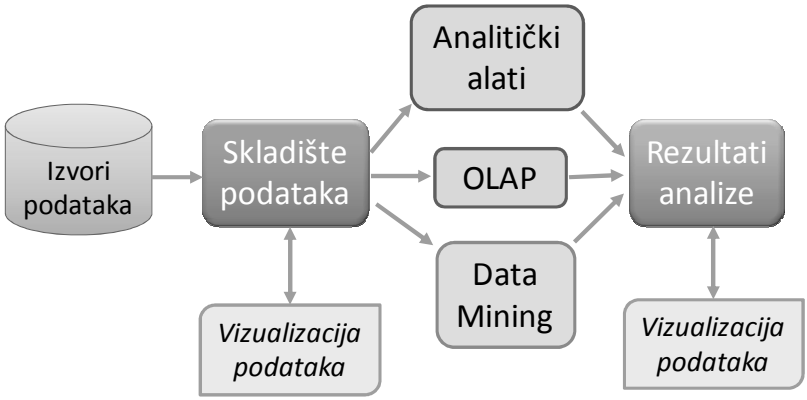
5.4 Istraživanje podataka

Istraživanje podataka (*Data Mining*) je označeno kao jedna od deset tehnologija koja će promeniti svet [24]. Postoje brojne definicije pojma istraživanja podataka, npr.

- ustanovljavanje zavisnosti u raspoloživim podacima [25];
- ekstrakcija implicitnih, prethodno nepoznatih i potencijalno korisnih informacija iz raspoloživih podataka [26];
- proces otkrivanja novih smislenih korelacija, obrazaca i trendova uvidom u veliki obim podataka smeštenih na računaru, korišćenjem tehnologije prepoznavanja uzoraka, kao i statističkih i metematičkih metoda (Gartner Group) [24];

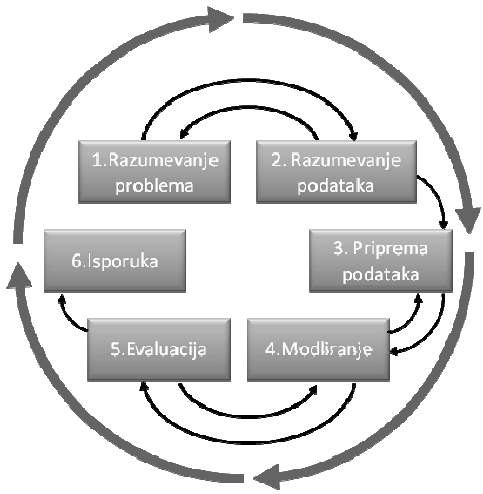
- proces otkrivanja novih obrazaca u velikim skupovima podataka metodima veštačke inteligencije, mašinskog učenja, statistike i baza podataka (ACM SIGKDD).

Istraživanje obimnih podataka je metod analize i dobijeni rezultati ne predstavljaju automatsko rešenje problema, već se dalje interpretiraju od strane čoveka i za njihovo korišćenje je potrebno specifično znanje o problemu koji se rešava, Sl. 51.



Sl. 51. Proces poslovne inteligencije

Sredinom devedestih godina predložen je standardni proces istraživanja podataka (*Cross-Industry Standard Process for Data Mining*, CRISP-DM) [24] koji predviđa šest iterativnih faza, Sl. 52.



Sl. 52. Faze standardnog procesa istraživanja podataka (CRISP-DM)

Tipični zadaci istraživanja podataka su deskripcija (*description*), estimacija (*estimation*), predikcija (*prediction*), klasifikacija (*classification*), klastering (*clustering*) i analiza asocijacija (*association*):

- *Deskripcija* predstavlja opisivanje obrazaca i trendova koji postoje u podacima. Za to se koriste pogodni metodi istraživanja podataka, koji daju opise razumljive ljudima, kao što su npr. stabla odlučivanja ili produkciona pravila.
- *Estimacija* je ustanovljavanje pravila za predviđanje numeričkih veličina, npr. prihoda prosečnog domaćinstva, kurseva valuta ili rezultata utakmica.
- *Predikcija* je predviđanje nenumeričkih i numeričkih veličina u budućnosti, npr. cene akcija kroz nekoliko meseci.
- *Klasifikacija* je ustanovljavanje pravila za predviđanje nenumeričkih veličina ili kategorija, npr. da li je neka transakcija ispravna ili sumnjiva, da li će prihodi biti niski, srednji ili visoki i slične klasifikacije u diskretne klase.
- *Klastering* je grupisanje sličnih opservacija ili slučajeva u grupe ili klastere, na takav način da se razlikuju od pripadnika drugih klastera. Obično se koristi kao početni korak istraživanja podataka, kada još nisu definisane ciljne promenljive.
- *Analiza asocijacija* se bavi grupisanjem sličnih atributa, npr. u analizi korpe za kupovinu, gde su atributi proizvodi, a traže se pravila za njihovo grupisanje koja omogućavaju prilagođavanje ponude ponašanju potrošača. Primer relacije je pravilo oblika IF {A and not B} THEN {C}, odnosno "98% ljudi koji kupuju poizvod A i ne kupuju proizvod B, kupuju proizvod C".

Metodi istraživanja podataka obuhvataju statističke i matematičke metode, kao što su multivariaciona analiza (*multivariate analysis*), analiza glavnih komponenti (*principal component analysis*), kao i metode mašinskog učenja u koje spadaju učenje klasifikacija i pravila regresije, učenje gupisanja (klastering), učenje višestrukih modela (*ensembles*), selekcija i estimacija atributa i analiza asocijacija [24], [25], [26].

Softverski alati za istraživanje podataka mogu biti:

- Samostalni integrisani alati istraživanja podataka, npr. alati otvorenog koda kao *Weka*, KNIME i *Rapid Miner*, odnosno komercijalni alati, kao što su *Knowledge Miner* i IBM *Intelligent Miner*;
- Samostalni programi mašinskog učenja određene namene, kao *CART* (stabla odlučivanja/regresije), *C4.5* (stabla odlučivanja), *C4.5Rules* (pravila) i *Ripper*;
- Delovi sistema poslovne inteligencije (BI) za učenje stabala odlučivanja, neuronskih mreža i modela nosećih vektora u sistemu otvorenog koda Pentaho i komercijalnim sistemima kompanija Oracle, IBM, Microsoft i dr.

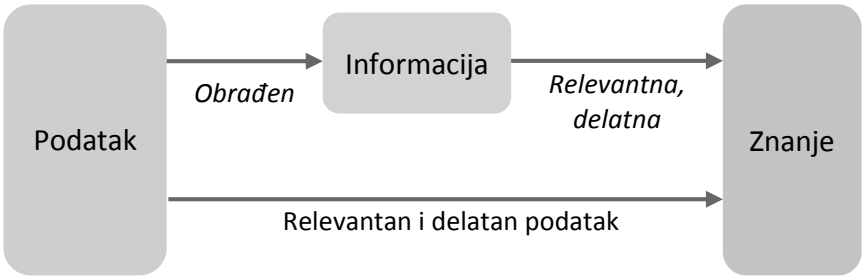
5.5 Upravljanje znanjem

Upravljanje znanjem predstavlja formu kolaborativnog računarstva čiji je cilj prikupljanje, smeštanje, održavanje i isporuka korisnog znanja svima u organizaciji kojima je potrebno, uvek i na svakom mestu. Upravljanje znanjem menja način grupnog rada i upotrebe računara.

Kao primer potrebe za upravljanjem znanjem može se uzeti kompanija *Siemens AG*, s tradicijom dužom od 150 godina, gotovo pola miliona zaposlenih širom sveta i velikim znanjem inženjera kompanije distribuiranim kroz vreme i prostor.

Znanje

Znanje se razlikuje od podatka i informacije, Sl. 53. Informacija je obrađeni podatak, a znanje se može razmatrati kao kontekstno zavisna, relevantna i delatna informacija ili podatak [1].



Sl. 53. Podatak, informacija i znanje

Npr. saobraćajna mapa se može posmatrati kao skup podataka, jer nema sve važne informacije relevantne za vožnju, izveštaj o trenutnom stanju na putevima kao informacija, dok se svest o postojanju alternativnih zaobilaznih pravaca može posmatrati kao znanje, jer omogućava delovanje.

Organizacije raspolažu podacima i informacijama, a znanje predstavlja viši nivo njihovog značenja. Specifična svojstva znanja kao imovine su:

- vrednost znanja se upotrebom ne smanjuje, već povećava;
- znanje se uvećava, grana i deli, pa ga treba stalno osvežavati;
- vrednost ulaganja u znanje je teško procenjivati, kao i vrednost deljenja znanja.

Upotreba znanja i upravljenje znanjem deo su globalne ekonomije, jer omogućavaju kompanijama bolje prilagođavanje brzim promenama poslovnog okruženja. Naziv *intelektualni kapital* se koristi kada se razmatra finansijska vrednost znanja.

Vrste znanja

Znanje se može posmatrati kroz dve osnovne forme, eksplicitno (*leaky knowledge*) i implicitno ili prećutno znanje (*tacit knowledge*). Eksplicitno znanje obuhvata racionalno, tehnički orijentisano znanje, koje uključuje politike, procedure, strategije, softver i dokumentaciju. Implicitno znanje je subjektivno i empirijsko, veoma se teško formalizuje. Tu spada iskustvo i individualno, ekspertsko znanje.

Informacioni sistemi tradicionalno prikupljaju eksplicitno znanje. Moderne organizacije uspostavljaju svoje sisteme za učenje iz sopstvenog iskustva, koje se zasniva na definisanju smislu, cilja učenja i merenju nivoa znanja. Sistemi za upravljanje znanjem (*Knowledge Management Systems*), koji predstavljaju više metodologiju nego tehnologiju, pomažu u kreiranju, prikupljanju, usavršavanju, pamćenju i jednostavnijoj upotrebi znanja. Oslanjaju se na komunikacione i kolaboracione tehnologije (GSS), kao i različite tehnologije smeštanja i pronalaženja informacija, uključujući baze podataka i inteligentne agente, koji pronalaze i izvlače znanje u različitim formama predstavljanja.

5.6 Primeri sistema poslovne inteligencije

Primeri sistema poslovne inteligencije će se obraditi u poglavlju posvećenom softverskim alatima mašinskog učenja i istraživanja podataka.

Pitanja za ponavljanje

1. Šta čini poslovnu inteligenciju?
2. Šta su skladišta podataka?
3. Koje su osnovne komponente procesa skladištenja podataka?
4. Koje arhitekture skladištenja podataka poznajete?
5. Šta je integracija podataka i koje su joj komponente?
6. Šta je "ETL proces"?
7. Koji su osnovni pritupi razvoju skladišta podataka?
8. Koja je osnovna struktura modela skladišta podataka?
9. Zbog čega se vrši skladištenje podataka u realnom vremenu?
10. Objasnite razliku između podatka, informacije i znanja.

6 Složeno odlučivanje: grupno donošenje odluka

1. Proces grupnog odlučivanja
2. Modeli grupnog odlučivanja
3. Sistemi za podršku saradnje i grupnog odlučivanja
4. Primeri sistema za grupni rad i odlučivanje

U ovom poglavlju se izlažu složeni modeli odlučivanja i neki sistemi za podršku grupnom odlučivanju.

6.1 Proces grupnog odlučivanja

Grupni rad je rad koji izvrši dvoje ili više ljudi zajedno. Pri tome, vreme i mesto izvršenja ne moraju biti isti, a mogu da učestvuju pripadnici više različitih organizacija. U grupi mogu postojati saradnja (sinergija), ali i konflikt. U zavisnosti od okolnosti, rezultati grupnog rada mogu biti bolji, ali i slabiji nego rad pojedinaca.

Rad grupa menadžera i analitičara je grupno odlučivanje, koje se najčešće realizuje na sastancima.

Ilustracija: Boeing Rocketdyne

Kompanija je postavila odeljenju za razvoj veoma složen projektni zahtev: da pronađe rešenje, koje će smanjiti cenu motora za 95%, a vreme razvoja i proizvodnje skratiti 10 puta [1]! Vreme za pronalaženje rešenja je takođe bilo veoma kratko, svega 10 meseci.

Odeljenje je uspelo, tako što je formiran tim vrhunskih stručnjaka iz više disciplina i menadžera, iz više kompanija, koji su sa 15% svog radnog vremena radili na postavljenom problemu.

Obezbeđena su tehnička sredstva za saradnju i razvijen poseban softver za grupni rad i saradnju, *InternetNotebook*.

Razvojni tim je održao 89 virtuelnih sastanaka (2-3 nedeljno) i pronašao rešenje u skladu s postavljenim zahtevima.

Sastanci (*meetings*) organizuju se radi stvaranja optimalnih uslova za donošenje grupne odluke. Neki od važnih uslova su:

- učesnici sastanka treba da imaju približno isti status;

- odluke zavise od znanja, mišljenja i odluka pojedinih učesnika i njihove podrške rešenju;
- odluke koje imaju podršku većine ili svih menadžera lakše se realizuju;
- različita mišljenja se obično prevazilaze pregovaranjem ili arbitražom;
- učesnici mogu biti na različitim lokacijama (*virtual team*);
- grupno odlučivanje može dati i pozitivne i negativne efekte.

Grupno odlučivanje je proces donošenja odluka od strane grupe menadžera. Predstavlja najsloženiji oblik odlučivanja.

Kao i individualno donošenje odluka, proces grupnog odlučivanja se može opisati nizom aktivnosti:

1. Definisanje problema
2. Određivanje zahteva
3. Određivanje ciljeva
4. Generisanje alternativa
5. Definisanje kriterijuma
6. Izbor metoda *grupnog* rešavanja
7. Ocena i izbor najbolje alternative
8. Ocena rešenja
9. Implementacija rešenja

6.2 Modeli grupnog odlučivanja

Na osnovu skupa alternativa i ciljeva jednog ili više donosilaca odluka, treba pronaći onu koja najbolje zadovoljava ciljeve ili rangirati alternative u skladu sa ciljevima. Poznati metodi grupnog izbora najboljeg rešenja su:

- autoritarni, gde odluku donosi lider;
- većinski, gde se odluka donosi glasanjem;
- negativni većinski;
- metod rangiranja alternativa;
- konsenzus.

Metod većinskog odlučivanja ili glasanja bira alternativu za koju se opredeli više od 50% (prosta većina) ili više od 66% (kvalifikovana većina) grupe. Ovaj metod je problematičan kada se bira između $n>2$ alternativa, jer se može dogoditi da odluku donese *manjina* od $1/(n+1)$ glasača;

Metod izbora rangiranjem, je pogodan kada se bira između više od $n>2$ alternativa. Poznati metodi izbora rangiranjem su, npr.

1) Bordin metod rangiranja (Borda[10]) se zasniva na rang listi alternativa koju glasači popunjavaju rednim brojevima. Na osnovu pojedinačnih rang lista formira se tabela u kojoj svaka od *n* alternativa dobija određen broj bodova na osnovu svakog glasa, tako što se prvo mesto na rang listi boduje sa *n*-1, a svaki sledeći s po jednim bodom manje, tako da poslednji ne dobija bodove.

   Metod je nastao u vreme francuske buržoaske revolucije, kao pokušaj da se izgradi dobar i pravedan izborni sistem, ali su u njemu mogući nerešeni rezultati, kada je potreban novi krug glasanja.

2) Kolektivno poređenje u parovima (Condorcet[11]) se zasniva na brojanju glasova koji međusobno porede sve parove alternativa.

3) Metod medijane (Kemeny, 1959) minimizuje sumu simetrične razlike relacija preferencije (lista).

Kao primer grupnog odlučivanja ilustrovaće se različiti metodi glasanja u velikim društvenim grupama.

Ilustracija: Metodi grupnog odlučivanja

Ilustrovaće se upotreba metoda grupnog odlučivanja za izbor jedne od ponuđenih alternativa na osnovu više pojedinačnih mišljenja metodom većinskog glasanja i dva metoda izbora rangiranjem.

Metodi će se ilustrovati na primeru odlučivanja grupe od 23 glasača, koji treba da izaberu jednu od tri alternative: *a*, *b* ili *c*.

Primer izbora većinskim glasanjem

1. Većinsko glasanje s rasporedom glasova po alternativama
   $a = 9$
   $b = 6$
   $c = 8$,

   Rezultati glasanja je da je izabrana alternativa ***a*** (9 glasova od 23 ili 39,1%).

Primer izbora metodom rangiranja (Borda)

2. Metod rangiranja svakoj alternativi za rang 1, 2, 3 na svakom listiću dodeljuje po 2, 1, 0 poena, odnosno:

rang 1.	*a*	*a*	*b*	*b*	*c*
rang 2.	*b*	*c*	*a*	*c*	*b*
rang 3.	*c*	*b*	*c*	*a*	*a*
br.glasača	5	4	2	4	8 (**23**)

[10] Borda - Jean-Charles, chevalier de Borda (1733-1799)
[11] Condorcet - Marie Jean Antoine Nicolas de Caritat, marquis de Condorcet (1743-1794)

Rezultati glasanja izraženi u poenima su:
a 2·(5+4)+1·2+0·(4+8)= 20
b 2·(2+4)+1·(5+8)+0·4= **25**
c 2·8 +1·(4+4)+0·(5+2)= 24
Izabrana je alternativa ***b***, iako se vidi da je *većina* glasala za to da je alternativa *c* bolja od alternative *b* (12 glasača, 8 da bude *c* ispred *a*, *b* i 4 za *c* ispred *b*).

Primer izbora poređenjem u parovima (Condorcet)

---

3. Grupno poređenje u parovima koristi iste rezultate glasanja:

rang 1.	*a*	*a*	*b*	*b*	*c*
rang 2.	*b*	*c*	*a*	*c*	*b*
rang 3.	*c*	*b*	*c*	*a*	*a*
br.glasača	5	4	2	4	8 (**23**)

Rangiranje alternativa se dobija brojanjem glasova *g* koji *međusobno porede* redosled parova elternativa. Funkcija poređenja parova alternativa *g*(*x*,*y*) broji glasove koji su za relativni poredak "*x* prethodi *y*", skraćeno "*x*>*y*".

U prikazanim rezultatima glasanja imamo tri alternative i 6 mogućih parova:
g(a,b)=9, *g(b,a)*=14 ⇒ *"b>a"*
g(a,c)=9, *g(c,a)*=12 ⇒ *"c>a"*
g(b,c)=11, *g(c,b)*=12 ⇒ *"c>b"*

Ove tri izvedene parcijalne relacije daju totalni poredak "c>b>a", ali u opštem slučaju i kod ovog metoda rezultat može da bude *nerešen*.

Postoje drugi metodi glasanja, npr. metod medijane i metod potencijala.

6.2.1 Model višekriterijumskog odlučivanja

Donošenje odluka obično podrazumeva primenu više kriterijuma za ocenu alternativa, koji mogu biti međusobno suprotstavljeni. Postoji veliki broj praktičnih situacija koje podrazumevaju uticaj više suprotstavljenih kriterijuma. Npr. visoki profiti su po pravilu povezani s rizičnim ulaganjima. Donosioci odluka moraju da pronađu kompromis između profita i rizika. Modeli višekriterijumskog odlučivanja su, npr.

- *Delphi tehnika*, interdisciplinarni konsenzus u kompleksnim problemima odlučivanja, bez neposredne komunikacije učesnika (1953, koristi upitnike za prikupljanje podataka);

- *nominalna grupna tehnika*, pojedinačno rangiranje alternativa i formiranje timskog konsenzusa.

Nauka o menadžmentu razmatra dve kategorije metoda računarske podrške višekriterijumskog odlučivanja [X]:

- *Višekriterijumsko ocenjivanje* (*Multiple Attribute Decision Making*, MADM), izbor najpovoljnije odluke iz ograničenog skupa mogućih alternativa na osnovu više atributa;
- *Višekriterijumska optimizacija* (*Multiple Objective Decision Making*, MODM), donošenje odluke na osnovu velikog/neograničenog broja alternativa, implicitno definisanih matematičkim uslovima.

Modeli višekriterijumskog odlučivanja vrše dekompoziciju složenog, opšteg problema odlučivanja na jednostavnije, manje složene probleme. Alternative ili opcije se dekomponuju u različite dimenzije, koje se se nazivaju atributi, promenljive, kriterijumi i sl. Svaka alternativa se zasebno ocenjuje, a rezultat se dobija procedurom agregacije ocena, odnosno funkcijom korisnosti (*utility function*), na osnovu koje se vrši izbor konkretne odluke ili rangiranje alternativa.

Model višekriterijumskog ocenjivanja je jednostavna procedura ocenjivanja ili rangiranja alternativa na osnovu svakog od kriterijuma. Višekriterijumsko ocenjivanje za kvantitativne atribute vrši se tako da se kriterijumima dodeli značaj (težina) $w_i \in [0..1]$, $\Sigma w_i = 1$, formira matrica odlučivanja, koja sadrži ocene svih alternativa kao proizvod svih definisanih kriterijuma i njihovih težina, a zatim se ocene alternativa agregiraju pomoću funkcije evaluacije. Ukupna ocena alternative *j* je težinska suma ovih ocena:

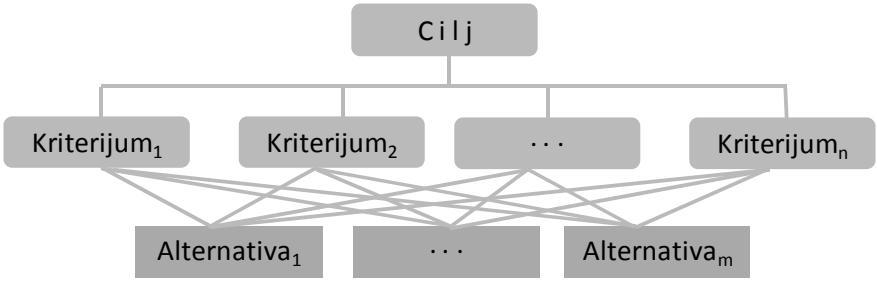
$$S_j = \sum_i w_i \, s_{ij}$$

gde je s_{ij} ocena (skor) alternative *j* po kriterijumu *i*, a w_i relativna važnost kriterijuma *i* za donosioca odluke.

Ponekad je u modelu višekriterijumskog ocenjivanja teško subjektivno određivati ocene i težine kriterijuma. Tada se može koristiti drugi model, npr. analitički hijerarhijski proces, koji obezbeđuje strukturiran pristup određivanju ocena i težina u modelu višekriterijumskog ocenjivanja. Metod je pogodan za grupno ocenjivanje i donošenje odluka.

Analitički hijerahijski proces (AHP) je praktični metod višekriterijumskog odlučivanja (Saaty, 1980), koji se zasniva na hijerarhijskom strukturiranju skupa kriterijuma. Osnovni koraci procesa su:

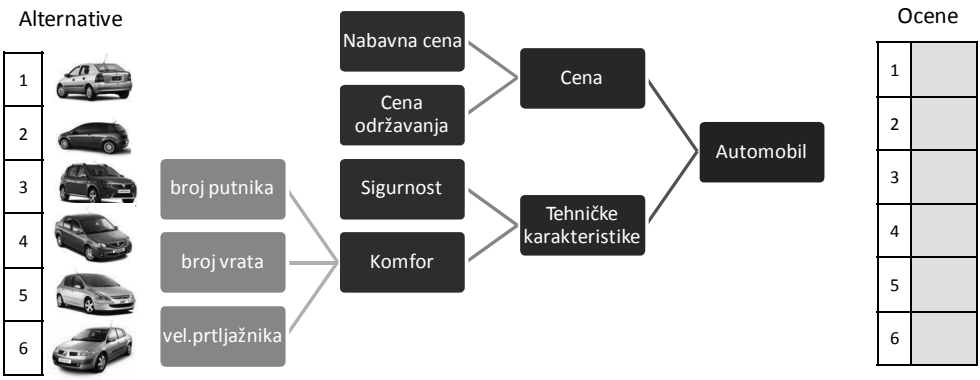
- *Dekompozicija*: problem se deli na manje delove, koji se hijerarhijski strukturiraju, tako da na jednom nivou ne bude suviše veliki broj kriterijuma, (npr. Milerov magični broj 7±2 elemenata):



- *Poređenje u parovima*: relativna važnost kriterijuma određuje se njihovim poređenjem u parovima. Značaj kriterijuma se dobija pomoću posebne skale (1..9, *Saaty's nine-point scale*);
- *Sinteza prioriteta*: Težine kriterijuma se izračunavaju iz matrice odlučivanja, a relativne težine se dobijaju njihovim dodavanjem u hijerarhiju izbora alternativa.

Primer dekompozicije problema izbora alternative na potprobleme i kreiranja hijerarhije atributa može se pokazati na primeru donošenja odluke o kupovini automobila. Model treba da oceni *alternative*, konkretne modele automobila [27], Sl. 54.

Model problema je stablo atributa, koji mogu biti kvalitativni i kvantitativni. Skup vrednosti atributa može biti uređen, parcijalno uređen ili neuređen. Računanje funkcije koristi (*utility function*) $f: X_1 \times X_2 \times ... \times X_n \rightarrow Y$ vrši se agregacijom vrednosti atributa od listova ka korenu stabla [27]:

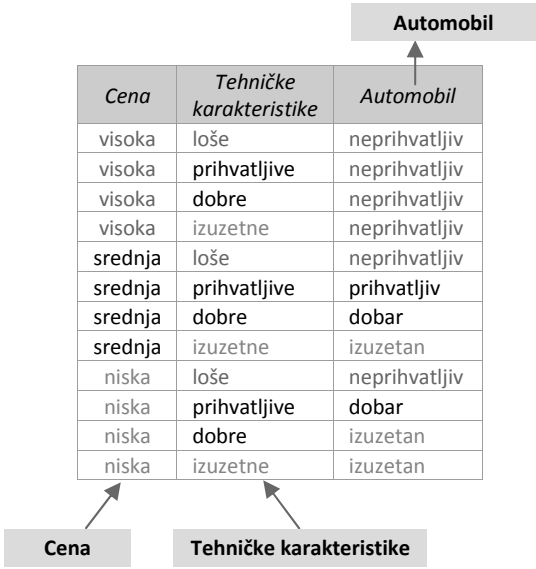


Sl. 54. Primer hijerarhijskog modela višekriterijumskog odlučivanja

Agregacija ocene modela sa Sl. 54 počinje računanjem funkcije koristi svih čvorova na prvom nivou stabla. Dobijaju se vrednosti atributa *Cena* i *Komfor*. Na sledećem nivou agregiraju se vrednosti atributa *Komfor* i *Sigurnost*, pa se dobija se vrednost atributa *Tehničke karakteristike*. Na kraju, na osnovu

vrednosti atributa *Cena* i *Tehničke karakteristike* računa se funkcija koristi *Automobil* u korenu stabla.

Funkcija agregacije ne mora biti numerička. Agregacija diskretnih vrednosti tada se vrši na osnovu definicije funkcije koristi, koja je zadana tabelarno. Vrednost funkcije dobija se iz tabele za odgovarajuću kombinaciju vrednosti argumenata. Primer je diskretna tabelarno zadana funkcija *Automobil*(*Cena*, *Tehničke karakteristike*), na Sl. 55 [27]. Za sve kombinacije diskretnih vrednosti argumenata *Cena*∈{*niska*, *srednja*, *visoka*} i *Tehničke karakteristike*∈{*loše*, *prihvatljive*, *dobre*, *izuzetne*} definisana je diskretna vrednost funkcije *Automobil* ∈{*neprihvatljiv*, *prihvatljiv*, *dobar*, *izuzetan*}.



Sl. 55. Primer diskretne tabelarno zadane funkcije

Model problema izbora automobila u primeru sa Sl. 54 može se realizovati tako da koristi samo kvalitativne atribute, tako što se izvrši diskretizacija kvantitativnih atributa, npr. broj vrata može da ima diskretne vrednosti koje predstavljaju broj vrata 1..5, ali bez numeričkog značenja. Funkcije agregacije (koristi) za sve atribute tada se zadaju tabelarno.

Diskretizacija kontinualnih atributa omogućava upotrebu sistema za podršku višekriterijumskom odlučivanju, kao što je sistem DEXi [27], koji koriste simbolički neparametarski metod i kvalitativno predstavljanje elemenata odlučivanja, koji najčešće odgovara ljudskom načinu razmišljanja.

6.3 Sistemi za podršku saradnji i grupnom odlučivanju

Softverski alati za grupni rad se često nazivaju *groupware*. Sistemi za podršku grupnom radu (*group support systems, GSS*) su računarski zasnovana klasa tehnologija i metoda, razvijenih za podršku grupnog rada i povećanje efikasnosti i efektivnosti sastanaka.

Sistem za podršku grupnom odlučivanju (*group decision support system,* GDSS) je kolekcija hardvera, softvera, ljudi i procedura, uređena na pogodan način, radi podrške grupnom odlučivanju.

Podrška može biti *indirektna*, kroz podršku saradnji i komunikaciji grupa i *direktna* podrška samom procesu donošenja odluke.

Sistemi za podršku grupnom odlučivanju su:
1. Sistemi za indirektnu podršku
2. Integrisani alati za podršku
3. Sistemi za direktnu podršku
4. Sistemi za podršku kreativnosti i stvaranju ideja

6.3.1 Sistemi za indirektnu podršku grupnom odlučivanju

Alati za grupni rad (*groupware*) mogu biti *sinhroni*, tako da se grupni rad obavlja istovremeno, i *asinhroni*, tako da se grupni rad učesnika obavlja u različito vreme.

1. Sinhroni alati za saradnju u realnom vremenu (*real-time collaboration tools*, RTC), među koje spadaju videokonferencijski sistemi (različite komunikacije) i Web konferencije (Internet/Web komunikacije).

2. Asinhroni alati za podršku za asinhronim komunikacijama, npr. e-mail sistemi, SMS sistemi, *online workspace* sistemi (kao Microsoft *SharePoint*), diskusione grupe i blogovi.

6.3.2 Integrisani alati za podršku grupnom odlučivanju

Integrisani alati za podršku grupnom odlučivanju su sistemi dobijeni integracijom više jednostavnijih softverskih alata u jedinstven sistem (*groupware suites*). Najpoznatiji sistemi su:

- IBM *Notes and Domino* (ranije Lotus *Notes/Domino*),
- Microsoft *NetMeeting*/*Live Meeting*/*SharePoint*,
- *Sococo*,
- Cisco *WebEx Meeting Center*.

6.3.3 Sistemi za direktnu podršku grupnom odlučivanju

Interaktivni računarski zasnovani sistemi koji omogućavaju grupno rešavanje poludelimično strukturiranih i nestrukturiranih problema su:

- sistemi za podršku grupnom odlučivanju (GDSS), npr. sobe za elektronske konferencije/odlučivanje;
- sistemi za podršku grupnom radu/saradnji:
  - grupni rad i odlučivanje (*collaborative Web meetings*): Facilitate Pro;
  - podrška grupnom razvoju softvera (*crowdsourcing*): TopCoder, Amazon Mechanical Turk;
  - odgovori članova grupe na pitanja (Q&A): Quora, Yahoo! Answers;
- sistemi za elektronske konferencije/odlučivanje (EMS).

6.3.4 Podrška kreativnosti i stvaranju ideja

Kreativnost je složen psihološki fenomen. Softver pomaže generisanje ideja, kroz računarsku podršku za *brainstorming*, neformalni grupni rad na otkrivanju novih pogleda na problem, oslobađanje i ohrabrivanje kreativnosti učesnika. Jedan vid podrške stvaranju i kreiranju ideja su i moderne aplikacije za upravljanje inovacijama, kao što je *Brightidea*, koje omogućavaju lako pokretanje, praćenje i poređenje inovativnih inicijativa.

Postoje eksperimentalna softverska rešenja koja demostriraju inteligentno ponašanje (savet i podrška u rešavanju složenih problema) i oponašaju ljudsku kreativnost (pisanje, komponovanje, slikanje).

6.3.5 Primeri

Važnije kategorije tehnologija i sistema za podršku grupnom radu i odlučivanju:

- VoIP www.telekom.rs
- Wiki-Wikilog www.sr.wikipedia.org
- kolaborativni B2B hub-ovi (e-trgovina) www.e-steel.com
- kolaborativne mreže (snabdevačke) www.logility.com
- korporativni portali www.ibm.com
- ostalo www.creativethink.com

Integrisani alati za podršku grupnom radu i odlučivanju (*groupware tools*) koji integrišu Web i *Office* tehnologije i interfejse u mrežnom radu su:

- Lotus Notes/Domino 9,
- Microsoft Share Point 2010/2013/2016.

6.4 Primeri sistema za grupni rad i odlučivanje

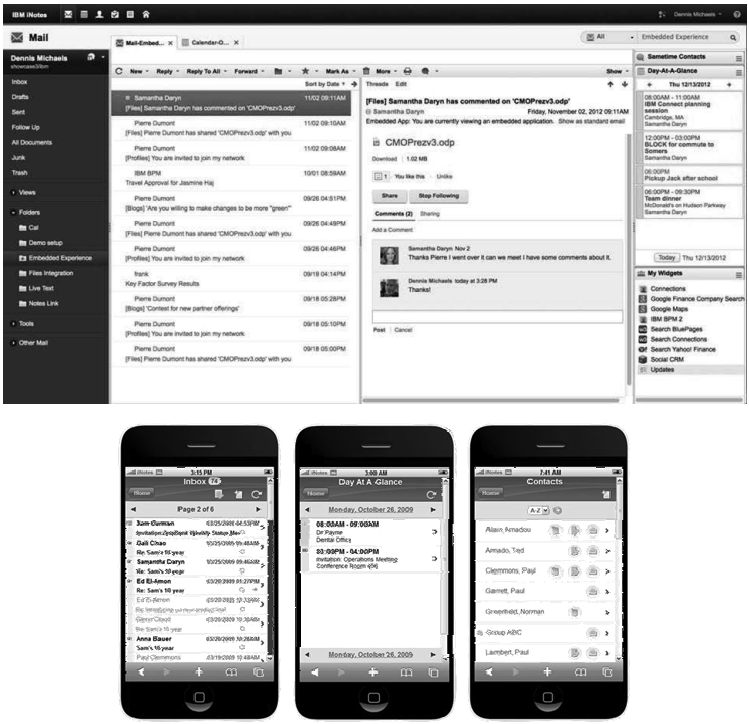
Kratko će se opisati integrisana okruženja za grupni rad (1) IBM *Notes Domino* i (2) Microsoft *SharePoint*, (3) sistem za direktnu podršku grupnom odlučivanju *FacilitatePro* i (4) primeri sistema za razvoj softvera i grupni rad na Internetu.

Primer 1: IBM Notes Domino 9 Social Edition/Live

IBM Notes and Domino (kao i Lotus Notes/Domino) integriše rad pomoću e-pošte, aplikacije za ćaskanje (*Instant Messenger*, IM), *Office* alata za kreiranje i uređivanje dokumenata, elektronskih tabela i prezentacija.

Sistem je uticao i na prilagođavanje grupnom radu ranijih poslovnih aplikacija, poput npr. *HelpDesk*-a, *CRM*-a, e-prodaje, foruma, blogova [72].

Grupni rad i saradnja oslanjaju se na e-poštu kao ključnu aplikaciju, koja se svakodnevno koristi, bilo da je virtuelni desktop na stolu, u kolima, u ruci ili u uhu.



Lotus Notes/Domino je uveden s namerom da bude "radni sto" budućnosti, jer objedinjuje sve svoje alatke za saradnju na jednom ekranu i predstavlja lični portal u Web 2.0 okruženju.

Primer 2: Microsoft SharePoint 2010/2013/2016

Sistem se sastoji od jednostavnih softverskih komponenti na bazi nekoliko poznatih Microsoft *Office* programa, koje funkcionišu zajedno u cilju obezbeđenja saradnje, poboljšanja distribucije informacija i podrške menadžmentu u okviru organizacije.

Interfejs sistema su Web stranice, a pristup funkcijama ostvaruje se putem korisničkog interfejsa, koji je realizovan po ugledu na Microsoft *Office*. Pošto se vrši ažuriranje Web sajta korišćenjem Web komponenti, sistem *SharePoint* je istovremeno i sistem za upravljanje Web sadržajem.



Efekti upotrebe sistema

Sistem omogućava saradnju i poboljšava efekte zajedničkog rada članova tima koji rade s informacijama tako što olakšava podelu rada, upravljanje informacijama i praćenje i upravljanje poslovnim procesima.

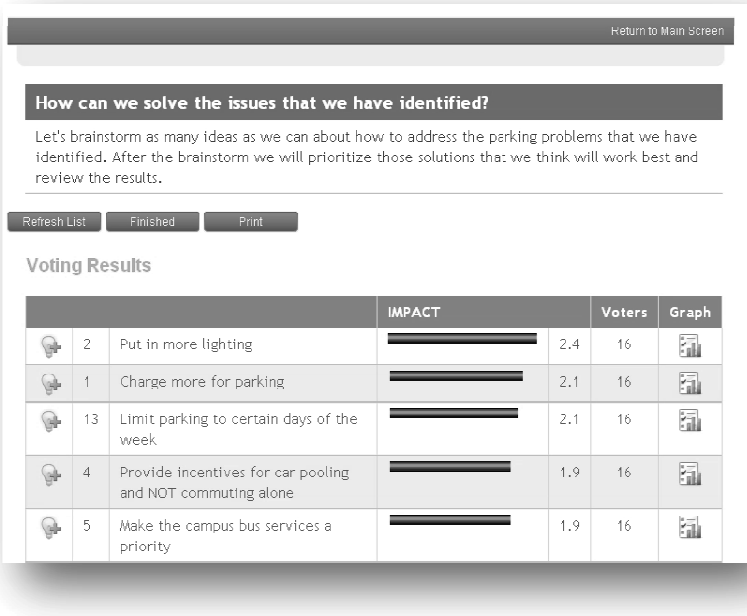
Potencijalne koristi korišćenja sistema uključuju:

- poboljšanu efektivnost tima,
- brz odziv na promene situacije,
- bolje upravljanje sadržajima,
- poboljšanu kontrolu nad zadacima,
- bolju distribuciju informacija,
- smanjenje troškova,
- brže postizanje ciljeva.

Primer 3: FacilitatePro

Sistem FacilitatePro [63] služi za direktnu podršku grupnom odlučivanju (*collaborative Web meetings*). Omogućava održavanje Web sastanaka i konferencija i kolaborativno učenje na virtuelnom radnom mestu (*interactive Webinars*).

Primer moguće uspešne upotrebe ovakvog sistema je kolaborativno rešavanje komunalnih problema. Na Sl. 57. prikazan je primer pregleda rezultata glasanja, sprovedenog nakon faze prikupljanja i klasifikacije predloga za rešavanje jednog od komunalnih problema.



Sl. 56. Pregled rezultata Web glasanja

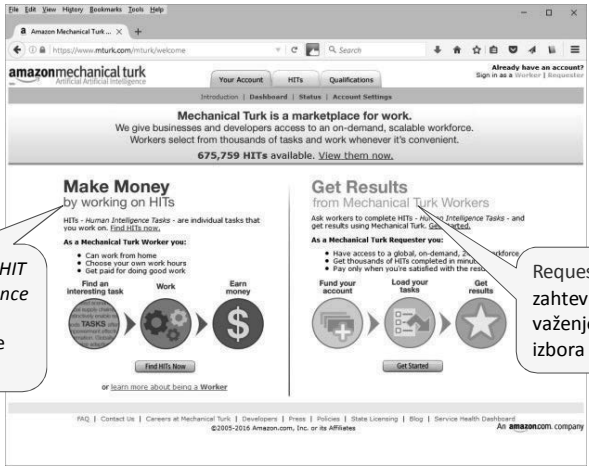
Primer 4: Sistemi za kolaborativni rad i razvoj softvera (*crowdsourcing*)

Termin *crowdsourcing* odnosi se na dobijanje usluga, ideja ili sadržaja prikupljanjem ponuda od velike grupe ljudi, obično s Interneta. Poznati su Web sajtovi za kolaborativni razvoj softvera, kao što je *TopCoder* i za različite računarske usluge, kao što je *Mechanical Turk* kompanije Amazon.

Na sajtu *TopCoder*, softver se razvija kroz niz kraćih takmičenja, u kojima zainteresovani izvršioci nude svoja rešenja algoritama, dizajna i razvoja softvera i otkrivanja greški u postojećem softveru. Najbolje rangirani takmičari dobijaju novčane nagrade. Sajt ima oko milion registrovanih

članova, čiji status se neprekidno prati. U proces razvoja softvera putem takmičenja uključile su se kompanije *Google* i *Yahoo*, a razvoj softvera na ovaj način sprovode i NASA, *Microsoft* i *Oracle*.

Na sajtu *Mechanical Turk* nude se različiti manji poslovi, kao što su pisanje, online istraživanja i prepoznavanje fotografija, koji se dobijaju aukcijski. Poslodavci (*Requestors*) nude poslove malog obima i vrednosti (do 20 dolara) s opisom posla, rokovima za prijavu i kriterijuma izbora izvršilaca, a zaintersovani izvršioci (*Workers*) se javljaju s ponudom rokova i cene. Konačnu odluku donosi poslodavac, koji nakon isporuke softvera uplaćuje dogovorenu naknadu na račun izvršioca.



Workers: rad na *HIT* (*Human Intelligence Tasks*), pregled poslova i davanje ponuda

Requesters: podnošenje zahteva (*HIT*) - rok, važenje, kriterijumi izbora radnika, cena

---

Pitanja za ponavljanje

1. Da li je saradnja preduslov grupnog rada (*groupwork*)?
2. Navedite zašto se grupno odlučivanje vrši na sastancima.
3. Objasnite ideju modela višekriterijumskog odlučivanja.
4. Navedite bar tri primera metoda izbora alternative na osnovu više mišljenja.
5. Koji je osnovni nedostatak izbora alternative metodom većinskog glasanja?
6. Koju alternativu od tri ponuđene je izabrala grupa od 20 učesnika *metodom rangiranja* (Borda), ako su rezultati rangiranja u priloženoj tabeli?

rang	komb 1	komb 2	komb 3	komb 4	komb 4
1	a	a	b	b	**c**
2	b	c	a	c	**b**
3	c	b	c	d	**a**
broj učesnika	4	4	2	3	7

7. Koji se problem rešava izborom alternative metodom poređenja u parovima (Condorcet) umesto metodom rangiranja?

8. Koje vrste sistema za indirektnu podršku grupnom odlučivanju poznajete?
9. Navedite bar tri *integrisana* alata za podršku grupnom odlučivanju.
10. Koja je osnovna karakteristika sistema za *direktnu* podršku grupnom odlučivanju?
11. Šta je *crowdsourcing*?

7 Inteligentni sistemi: podrška donošenju složenih odluka

1. Osnovni pojmovi veštačke inteligencije
2. Inteligentni sistemi
3. Istraživanje podataka i mašinsko učenje
4. Inteligentni interfejsi
5. Inteligentni agenti

U ovom poglavlju se izlažu metodi i alati za podršku složenom odlučivanju zasnovani na znanju, odnosno metodima i tehnikama veštačke inteligencije. Ukratko se izlažu osnovni pojmovi veštačke inteligencije, razmatra pojam inteligentnih sistema i neke tehnologije za njihovu realizaciju, kao što su metodi istraživanja podataka i mašinskog učenja, inteligentni interfejsi i inteligentni agenti.

7.1 Osnovni pojmovi veštačke inteligencije

U ovoj tački uvode se osnovni pojmovi neophodni za razumevanje funkcionisanja i upotrebe ekspertnih sistema: veštačka inteligencija, mašinsko učenje, znanje, predstavljanje znanja, zaključivanje i ekspertni sistem.

Veštačka inteligencija

Veštačka inteligencija je disciplina računarskih nauka koja se bavi ponašanjem računara koje se, kada ga izvršava ljudsko biće, naziva inteligentnim.

Primer: Igra šah



Šah je igra koja zahteva inteligenciju, iako igra ima ima konačan broj mogućih varijanti (partija). Njihov broj je suviše velik da bi se igra računarski razmatrala u celini, pošto je uporediv s brojem neutrona u poznatom svemiru (Shannon-ova procena broja varijanti igre je oko 10^{120}, dok je procenjeni broj neutrona do 10^{81} [61]). Uprkos tome, šahovski program *Deep Blue* kompanije IBM je 1997. godine pobedio svetskog šampiona Garija Kasparova u meču koji se igrao po pravilima za ljudske igrače.

Tjuringov test: može li mašina da misli?

Veliki matematičar i kriptoanalitičar Alan Tjuring[12] pokušao je da formalno definiše pojam inteligencije računara, odnosno apstraktne računske mašine. Osnovna ideja njegovog metoda je da se odgovor može dati samo na osnovu eksperimenta, zbog čega je metod kasnije dobio naziv Tjuringov test.

Eksperiment se sastoji od tri elementa: inteligentnog sudije, računara i čoveka, raspoređenih u dve odvojene prostorije: u jednoj je sudija (C), a u drugoj računar (A) i čovek (B).

Računar (računska mašina) je inteligentan ako inteligentni sudija (C) ne može da razlikuje ponašanje računara (A) i ljudskog bića (B) prilikom izvršavanja različitih zadataka.

Primer kineske sobe

Tjuringov pristup je naišao na brojne kasnije kritike, npr. Džon Serl je 1980. godine izneo kao suprotan argument primer tzv. kineske sobe (*Chinese room*).

U eksperimentu se u posebnoj sobi nalazi čovek koji ne zna kineski jezik, ali raspolaže odgovarajućom literaturom za davanje odgovora na kineskom jeziku, kao i uputstvom na svom jeziku za pronalaženje i davanje odgovora.

Zamišljena je situacija gde subjekt može da daje smislene odgovore, a da ne razume ni reč kineskog jezika, tako da ne zna ni koji problem se rešava niti razume odgovor koji daje. Primer kineske sobe je pokušaj da se pokaže kako je moguće položiti Turingov test bez razumevanja i rasuđivanja.

Prema [4], osnovna obeležja inteligentnog ponašanja su:

- učenje ili razumevanje na osnovu iskustva;
- shvatanje dvosmislenih ili kontradiktornih poruka;
- brzo i uspešno regovanje u novim situacijama;
- korišćenje zaključivanja u rešavanju problema;
- snalaženje u nepredvidivim situacijama;
- razumevanje i zaključivanje na običan, racionalan način;
- upotreba znanja za manipulisanje okruženjem;
- mišljenje i rezonovanje;
- prepoznavanje relativne važnosti različitih elemenata neke situacije.

[12] Alan Mathison Turing (1912-1954), engleski matematičar, logičar, kriptoanalitičar i jedan od osnivača računarskih nauka i veštačke inteligencije.

Mašinsko učenje

Prema [28], mašinsko učenje (*machine learning*) je skup procesa u koje spadaju: prikupljanje novog deklarativnog znanja, razvoj i usavršavanje motornih i saznajnih sposobnosti kroz praksu, strukturiranje postojećeg znanja i otkrivanje novih činjenica i teorija posmatranjem i aktivnim eksperimentisanjem.

Inženjerski orijentisana definicija mašinskog učenja je da je to proces u kome sistem poboljšava svoje performanse na datom zadatku bez dodatnog programiranja [29].

Učenje se može posmatrati kroz dve osnovne forme [28]:

- prikupljanje znanja (*knowledge acquisition*) ili učenje logičkih zakonitosti;
- uvežbavanje (*training*) ili poboljšavanje nekog stečenog znanja, kroz praktično ponavljanje i korekciju odstupanja od željenog ponašanja.

Smatra se da učenje kod čoveka predstavlja mešavinu obeju formi, pri čemu mentalne aktivnosti potenciraju prvu formu, a motorne aktivnosti drugu formu učenja.

Sistemi mašinskog učenja najčešće se dele prema odabranoj strategiji učenja, načinu predstavljanja znanja i području primene [28]. Podela prema strategiji učenja je u odnsu na potrebnu količinu zaključivanja sistema: učenje memorisanjem (*rote learning*), učenje na osnovu rečenog (*learning by being told*), učenje po analogiji (*learning by analogy*), učenje na osnovu primera (*learning by examples*) i učenje posmatranjem i samostalnim otkrivanjem (*learning by observation and discovery*) ili učenje bez učitelja.

Pojam *induktivno mašinsko učenje* obuhvata poslednje dve strategije učenja: učenje na osnovu primera i učenje posmatranjem i samostalnim otkrivanjem. Poseban oblik induktivnog učenja je *učenje koncepata*, u kome se uče logičke zakonitosti koje zadovoljavaju *princip razumljivosti* (*principle of comprehensibility*).

Predstavljanje znanja

Znanje se može razmatrati kao kontekstno zavisna, relevantna i delatna informacija.

Predstavljanje znanja u računarskim sistemima može se realizovati na različite načine, npr. pomoću logičkih izraza, produkcionih pravila (IF .. THEN ..), semantičkih mreža, frejmova (*frames*) i skupa primera instanci objekata (*cases*) i njihovih međusobnih veza.

Znanje može biti *eksplicitno* i *implicitno*. Eksplicitno znanje je objektivno, racionalno, tehničko i lako se formalizuje. Obično se predstavlja matematičkom

logikom, produkcionim pravilima, frejmovima i sl. Implicitno (distribuirano) znanje je subjektivno, empirijsko i teško se formalizuje. Za njegovo predstavljanje se koriste npr. veštačke neuronske mreže i modeli nosećih vektora.

Znanje se može predstaviti *deklarativno* (opisno), *proceduralno* (u obliku računarskog programa) ili kao *metaznanje*.

Stepen razumljivosti predstavljenog znanja može biti:

1. *najniži* stepen razumljivosti, kada se znanje može koristiti samo uz pomoć računara,
2. *viši* stepen razumljivosti, kada se znanje može koristiti samo uz pomoć računara, ali postoji mehanizam za davanje objašnjenja,
3. *najviši* stepen razumljivosti, koji omogućava korišćenje naučenog znanja i bez pomoći računara.

Zaključivanje

Zaključivanje (*inference*) je proces izvođenja logičkih zaključaka na osnovu premisa za koje se zna ili pretpostavlja da su istinite. Zaključivanje može biti:

- Induktivno (od pojedinačnog ka opštem),
- Deduktivno (od opšteg ka pojedinačnom),
- Abduktivno (od konkretnog ka konkretnom).

Primer abdukcije:

if a & b	if pijan(x) & not hoda-pravo(x)
b	not hoda-pravo(*Petar*)
a	pijan(*Petar*)

Zaključivanje u uslovima neizvesnosti proučava se u okviru matematičkih disciplina verovatnoće i statistike. Postoje različite forme zaključivanja (logike), a osnovne su logika sudova i logika predikata.

Bajesova formula jedan je od pristupa zaključivanju na osnovu verovatnoća. Bajesova formule koristi se u kontekstu sistema za podršku odlučivanju, konkretno u dijagnostičkim ekspertnim sistemima, za računanje verovatnoća neke dijagnoze d na osnovu simptoma s. Pri tome se vrši aproksimacija verovatnoća relativnim frekvencijama. Oboljenja mogu da imaju više simptoma, pa se koristi izraz:

$$P(d|s_1\& \dots \&s_n) = P(d) \cdot \frac{P(s_1\& \dots \&s_n|d)}{P(s_1\& \dots \&s_n)}$$

Pri tome se pretpostavlja međusobna nezavisnost simptoma, odnosno da važi:

$$P(s_i|s_j) = P(s_i)$$
$$P(s_i|s_j\&d) = P(s_i|d)$$

Verovatnoća dijagnoze *d* (odluke) se tada može izračunati kao:

$$P(d|s_1\& \dots \&s_n) = P(d) \cdot \frac{P(s_1|d) \cdot \ \dots \ \cdot P(s_n|d)}{P(s_1\& \dots \&s_n)}$$

Ekspertni sistem

Ekspert je osoba koja poseduje specijalističko znanje iz neke uske oblasti, iskustvo i sposobnost rasuđivanja. *Ekspertiza* je specifično znanje eksperata, stečeno na osnovu učenja, obuke i prakse.

Ekspertni sistem je računarski zasnovan sistem, koji je namenjen rešavanju problema za koje je potrebno ekspertsko znanje, odnosno netrukturiranih ili delimično strukturiranih problema.

7.2 Inteligentni sistemi

Inteligencija se definiše opisno, pomoću pojma inteligentnog ponašanja, koje obuhvata:
- rasuđivanje (*Reasoning*),
- učenje (*Learning*),
- adaptaciju (*Adaptivity*).

Inteligentni sistem se samostalno adaptira promenama u problemima, odnosno pokazuje sposobnost automatizovanog učenja.

Mašinska inteligencija (*machine intelligence*) je disiplina koja se bavi izradom softvera koji se ponaša na sličan način kao i čovek na sličnim zadacima.

Inteligentni sistemi su softverski sistemi koji pokazuju mnoge od sposobnosti čoveka na poslovima koji zahtevaju inteligenciju.

Inteligentni sistemi u poslovanju obuhvataju različite softverske alate i sisteme, koji se često koriste u podršci odlučivanju. Očekivani poslovni efekti njhove primene su povećanje produktivnosti i postizanje kompetitivne prednosti.

Područja primene inteligentnih sistema u poslovanju obuhvataju npr. otkrivanje uzoraka ponašanja potrošača, marketinških trendova i povećanje efikasnosti.

Primeri inteligentnih sistema su sistemi za:
- podršku korisnicima (*Customer Relations Modelling*),
- planiranje,
- predviđanja na finasijskim tržištima (*Financial market prediction*).

Veštačka inteligencija i mašinsko učenje

Veštačka inteligencija (*Artificial Intelligence*, AI) je disciplina računarskih nauka čiji je osnovni cilj razvoj softvera koji omogućava računaru da rešava probleme metodima rezonovanja koje koristi čovek. Nastala je 1956. godine, na jednoj poznatoj računarskoj konferenciji.

Iako se pre svega bavi izgradnjom sistema zasnovanih na znanju i metodima rešavanja problema koje koristi čovek, obuhvata i proučavanje funkcionisanja mozga radi razumevanja njegove strukture i funkcije.

Poznate uspešne primene metoda veštačke inteligencije su npr.

- ekspertni sistemi (*Expert Systems*),
- sistemi zaključivanja na osnovu rešenih slučajeva (*Case Based Reasoning Systems*, CBR).

Metodi realizacije inteligentnih sistema

Inteligentni sistemi mogu istovremeno da koriste jedan ili više metoda realizacije, u koje spadaju:

1. Ekspertni sistemi (*Expert Systems*),
2. Veštačke neuronske mreže (*Artificial Neural Networks*),
3. Genetički algoritmi (*Genetic Algorithms*, GA),
4. Fazi sistemi (*Fuzzy Systems*),
5. Zaključivanje na osnovu slučajeva (*Case Based Reasoning*, CBR),
6. Istraživanje podataka/otkrivanje znanja (*Data Mining/Knowledge Data Discovery*),
7. Inteligentni agenti (*Intelligent Software Agents*),
8. Komunikacija u prirodnom jeziku (*Language Technology*).

Inteligentni agenti u e-trgovini

Primena u maloprodaji ili B2C (*Business-to-Consumer*) sistemima za:

- posredovanje proizvoda (*Product Brokering*) - agent predlaže kupcu proizvode (sistem *Jango*);
- trgovačko posredovanje (*Merchant Brokering*) - agent predlaže prodavca određenog proizvoda (sistem *BergainFinder*);
- pregovaranje - agent određuje cene ili druge elemente transakcija (aukcije i ugovori, T@T).

Primena u veleprodaji ili B2B (*Business-to-Business*) sistemima obuhvata npr. lance snabdevanja (*Supply Chain Management*).

Ostale novije primene su:

- mobilni agenti,
- evolucioni agenti (*Evolutionary Agents*),
- *Data Mining* agenti, itd.

7.3 Istraživanje podataka i mašinsko učenje

Istraživanje podataka može se definisati na različite načine, npr. kao proces otkrivanja novih obrazaca u velikim skupovima podataka metodima veštačke inteligencije, mašinskog učenja, statistike i baza podataka (ACM SIGKDD) [24].

Mašinsko učenje se može posmatrati kao proces u kome softverski sistemi poboljšavaju svoje performanse na datom zadatku bez dodatnog programiranja, ali i kao ustanovljavanje zakonitosti u raspoloživim podacima [25].

Vidi se da je zajednički cilj istraživanja podataka i mašinskog učenja ustanovljavanje zavisnosti ili zakonitosti u podacima [24], [25]. Istraživanje podataka i mašinsko učenje koriste veliki broj istih metoda i softverskih alata, jer je istraživanje podataka primenjeno interdisciplinarno područje, koje koristi statističke metode, metode mašinskog učenja, baza podataka i tehnike vizualizacije podataka.

Statistički metodi su npr. regresiona, korelaciona i klaster analiza i Bajesove mreže. Statistički modeli se kreiraju na osnovu podataka, a zaključke na osnovu tih modela izvode ljudi.

Metodi mašinskog učenja takođe grade modele na osnovu podataka, ali su podaci najčešće predstavljeni ne samo numeričkim dimenzijama, već atributima različitih tipova bližih originalnim kognitivnim modelima koje koriste ljudi kada interpretiraju iste podatke. Zbog toga metodi mašinskog učenja koriste posebne heuristike za izgradnju modela kao što su stabla odlučivanja, skupovi produkcionih pravila i logički izrazi. U istraživanju podataka se najčešće koriste metodi učenja razumljivog znanja, kao što su induktivno učenje stabla odlučivanja, produkcionih pravila i konceptualni klastering.

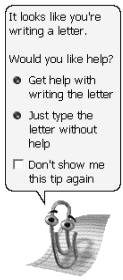
Metodi baza podataka ne traže najbolje modele, već ispituju svojstva podataka metodima modeliranja i posebnim heuristikama, kao što je analiza asicijacija, pretraživanje baza podataka radi pronalaženja čestih ponavljanja određenih kombinacija (*frequent itemsets*), na osnovu kojih se kreiraju pravila asocijacije.

Tehnike vizualizacije podataka transformišu podatke u vizuelne predstave, kao što su tačke, linije i površine, kreirajući dvodimenzionalne ili trodimenzionalne prikaze, kao što su dijagrami i grafovi. Transformacija se vrši tako da se očuvaju i istaknu određene relacije, koje postoje u podacima. Interaktivno ispitivanje ovih prikaza pomaže u pronalaženju interesantnih pogleda na podatke i identifikaciji tih relacija.

Deo područja istraživanja podataka, koji se bavi predviđanjem trendova i obrascima ponašanja, naziva se *prediktivna analitika*.

Novi termin za interdisciplinarno područje, koje se bavi procesima i sistemima za izdvajanje znanja iz (obično velikog obima) podataka ili njihovo razumevanje je *nauka o podacima* (*Data Science*). Predstavlja nastavak rada na nekim metodima analize podataka iz oblasti statistike, mašinskog učenja, istraživanja podataka i prediktivne analitike, a po nekim autorima u pitanju je samo noviji termin za napredne metode statistike [96].

7.4 Inteligentni interfejsi



Inteligentni korisnički interfejs (*Intelligent User Interface*, ili ponekad interfejs agent) je korisnički interfejs koji uključuje u svoje funkcionisanje neki aspekt veštačke inteligencije [61]. Poznatiji primeri su Microsoft *Office Assistant* (čija je prepoznatljiva vizuelna predstava bila spajalica, *Clippy*), koji nije uspeo da se dopadne korisnicima i Microsoft *Agent*, koji se kao animirani interfejs s podrškom govornoj komunikaciji (ActiveX kontrola) koristio zaključno s verzijom operativnog sistema Windows 7.

Inteligentni interfejs podrazumeva da računar ima znanje o području primene ili neki model korisnika, što mu omogućava da bolje razume potrebe korisnika i pomaže mu u interakciji s računarom.

Multimedijski princip daje prednost istovremenoj upotrebi više vrsta komunikacije čoveka i računara, čime se povećava sposobnost čoveka da primi nove informacije i time povećava obim, kvalitet i brzina učenja.

Razvijena je značajna računarska podrška razumevanju govora, prirodnog jezika (pre svega engleskog) i prepoznavanje rukom pisanog teksta, fizionomija, scena i pokreta, koji omogućavaju multimedijsku komunikaciju čoveka i računara.

Primer programa za inteligentnu tekstualnu komunikaciju[13] na engleskom jeziku je *Alice*, program za ćaskanje (*chatbot*), koji prikazuje animiranog virtuelnog sagovornika i pogledom prati kretanje miša na ekranu, Sl. 64 [67].

[13] Prvi sistem za komunikacija u prirodnom jeziku nastao je kao eksperiment iz veštačke inteligencije na MIT (1966). Nazvan je ELIZA, po liku iz poznatog komada Bernarda Šoa



Sl. 57. Program za inteligentnu tekstualnu komunikaciju Alice

Inteligentni multimedijski interfejsi omogućili su razvoj inteligentnih sistema za podršku, kao što su mobilni asistenti poznatih proizvođača: Apple *Siri*, Google *Now*, Microsoft *Cortana* i Samsung *S Voice*.

7.5 Inteligentni agenti

Preveliki obim informacija koje treba razmotriti prilikom donošenja odluke smanjuje sposobnost odlučivanja za 50% (Gartner) [1]. Sposobnost donošenja odluka je u današnje vreme povezana s automatizacijom procesa pretraživanja i obrade obilja informacija u računarskim sistemima, što je Alvin Tofler nazvao poplavom informacija (*Future Shock*, 1970).

Veliki broj aktivnosti u procesu odlučivanja podrazumeva samostalno delovanje učesnika. U svakodnevnom životu *agent* je osoba koja deluje samostalno i sarađuje s drugim licima radi obavljanja određenih poslova [1].

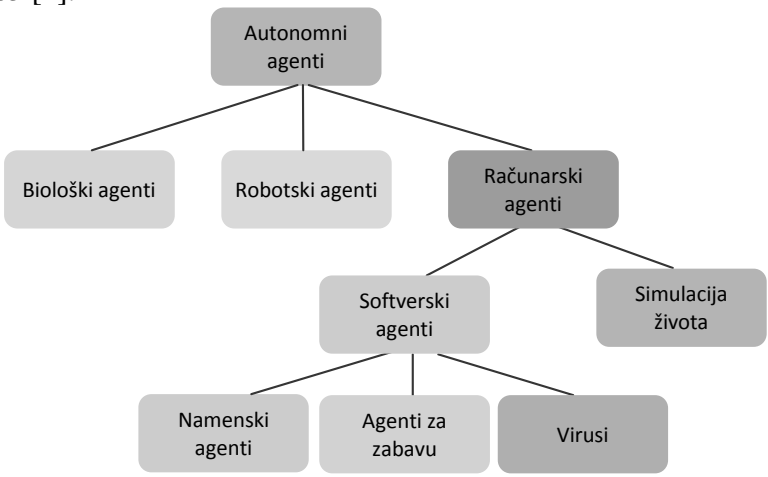
Računarski ili *softverski agent* je računarski program koji funkcioniše samostalno, opaža svoje okruženje, postoji tokom dužeg vremenskog perioda, prilagođava se promenama, kreira i postiže ciljeve. *Racionalni agent* [4] je

Pigmalion (Pigmalion je legendarni kiparski vajar, koji se zaljubio u savršenu skulpturu žene koju je izradio i koju je boginja Afrodita na njegovu molbu oživela).

agent koji deluje tako da ostvari najbolji očekivani ishod u uslovima neizvesnosti.

Inteligentni agent je računarski program koji pomaže korisniku u obavljaju rutinskih poslova na osnovu utvrđenih pravila i baze znanja [1].

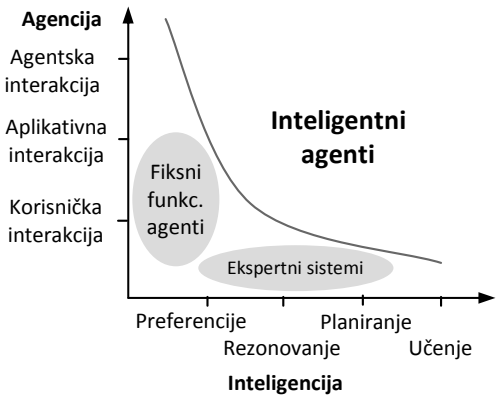
Softverski agent (*software daemon*, *software agent*) se još naziva čarobnjak (*wizzard*) ili softverski robot (*softbot*, *chatterbot*, *shopbot*, *knowbot*, *mailbot*, *spambot*). Klasifikacija autonomnih agenata prema njihovoj nameni prikazana je na Sl. 65 [1].



Sl. 58. Klasifikacija agenata prema nameni

Najvažnija svojstva ili dimenzije agenata su, Sl. 66:

1. *Agencija* - stepen autonomije (asinhronost);
2. *Inteligencija* - stepen zaključivanja i sposobnost učenja;
3. *Mobilnost* - sposobnost premeštanja u mrežnom okruženju.



Sl. 59. Svojstva agenata

Agencija je najvažnije svojstvo, jer agenti moraju da učestvuju u procesima i u vreme kad korisnik nije na mreži, kao npr. u višednevnim aukcijama.

U odnosu na područje primene, najpoznatije vrste agenata su npr.

- Internet agenti,
- e-mail agenti (*mailbot*),
- Web browsing asistenti,
- inteligentni pretraživači, koji za indeksiranje informacija koriste autonomne module ili robote (*robots*, *spiders*, *wanderers*),
- softboti, programi za pronalaženje informacija, npr. na sajtovima google.com, hotjobs.yahoo.com,
- agenti za upravljanje i nadzor,
- e-commerce agenti, kao što su bestwebbuys.com, ebay.com,
- inteligentni interfejsi,
- agenti za učenje, preporučivanje, izradu profila klijenata,
- agenti za Web istraživanje podataka (*Web mining*), itd.

Primeri agenata

Primeri agenata mogu se naći na razvojnim sajtovima velikih proizvođača softvera, školskih institucija i nekim namenskim sajtovima:

- Oracle www.oracle.com
- Microsoft www.microsoft.com
- IBM www.ibm.com
- Hyperion www.hyperion.com
- Harvard Bussines School Case Collection www.harvardbussinesonline.hbsp.harvard.edu
- Primeri inteligentnih agenata i interfejsa www.botspot.com

Pitanja za ponavljanje

1. Šta je predmet izučavanja veštačke inteligencije?
2. Navedite bar tri obeležja inteligentnog ponašanja.
3. Definišite ekspertizu i ekspertni sistem.
4. Koje vrste znanja i načine predstavljanja znanja razlikujemo?
5. Navedite osnovne metode istraživanja podataka (*Data Mining*).
6. Šta je nauka o podacima (*Data Science*)?
7. Šta je ELIZA?
8. Šta su inteligentni agenti?

9. Koja su osnovna svojstva (dimenzije) inteligentnog agenta?
10. Navedite bar pet primera (vrsta) Internet agenata.

8 Ekspertni sistemi

1. Nastanak ekspertnih sistema
2. Struktura ekspertnih sistema
3. Vrste ekspertnih sistema
4. Alati za razvoj ekspertnih sistema
5. Primeri ekspertnih sistema za podršku odlučivanju

U ovom poglavlju objašnjava se nastanak, struktura i komponente ekspertnih sistema, kao i metodi, tehnike i alati za njihov razvoj. Na kraju se daju ilustratuvni primeri realizovanih ekspertnih sistema

8.1 Nastanak ekspertnih sistema

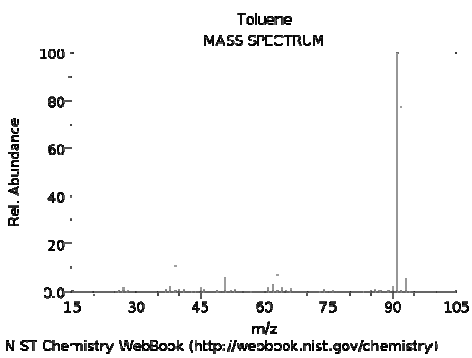
Ekspertni sistemi su računarski zasnovani sistemi namenjeni rešavanju nestrukturiranih problema za koje je potrebno ekspertsko znanje.

Sedamdesetih godina prošlog veka realizovana je ideja da se *nestrukturirani*, intelektualni problemi mogu uspešno rešavati uz pomoć računara, tako što se koristi (1) opšta procedura rezonovanja i (2) znanje, koje je specifično za svaki pojedinačni problem. Prvi uspešni ekspertni sistemi nastali su šezdesetih i sedmdesetih godina dvadesetog veka na univerzitetu Stanford.

Sistem DENDRAL (*Heuristic Dendral*, Stanford, 1965), prvi uspešan sistem zasnovan na znanju [18], namenjen je rešavanju nestrukturiranog problema identifikacije molekula u organskoj hemiji na osnovu masenih spektrograma. Razvijen je u jeziku LISP i prethodnik je ekspertnih sistema MYCIN, MACSYMA, PROSPECTOR, XCON. Jedan od proizvoda razvoja je i *Meta-Dendral*, program mašinskog učenja za induktivno generisanje pravila na osnovu primera.

Ilustracija: Identifikacija molekula u organskoj hemiji

U organskoj hemiji je čest problem identifikacija jedinjenja na osnovu masenog spektrograma, dijagrama raspodele masa sastavnih elemenata. Na slici je primer masenog spektrograma toluena (C_7H_8), sastojka nafte i ulja, koji se, između ostalog, koristi se kao dodatak gorivu formule 1 (*octan booster*), za hlađenje nuklearnih reaktora i u proizvodnji sirupa Koka-kole.



Proces identifikacije se zasniva se na specijalističkom znanju iz organske hemije, koje je u sistemu predstavljeno produkcionim pravilima oblika:

```
IF   spektrogram ima dva vrha za mase xl i x2, tako da važi
     xl+x2 = M+28            AND
     xl-28 je "visoki" vrh   AND
     x2-28 je "visoki" vrh   AND
     najmanje jedan od xl ili x2 je "visoki" vrh
THEN Molekul sadrži ketonsku grupu.
```

Sistem je pravila koristio za upravljajnje procesom pretraživanja mogućih rešenja. Pošto je dostigao uspešnost iskusnih hemičara, korišćen je kao anlitički alat organske hemije u industriji i nauci.

Sistem MYCIN (Stanford, 1970) je eksperimentalni ekspertni sistem za dijagnostiku bakterijskih infekcija i propisivanje antibiotskih terapija. Imao je oko 600 pravila i propisivao oko 69% ispravnih terapija u najtežim kliničkim slučajevima, čime je u testovima prevazišao uspešnost lekara. Nije ušao u kliničku upotrebu, ali su kao rezultati projekta nastali ljuska ekspertnih sistema Empty MYCIN i poseban metod zaključivanja u uslovima neizvesnosti, tzv. faktore pouzdanosti (*Certanity Factors*, CF). Prerastao u ljusku ekspertnih sistema KEE i (kasnije) ekspertni sistem CADUCEUS.

Znanje je teorijsko ili praktično razumevanje nekog pitanja ili određene oblasti. Predstavlja skup onoga što je trenutno poznato. Ljudi koji poseduju to znanje su eksperti. Ekspert je svako ko poseduje detaljno, specijalističko znanje i praktično iskustvo u rešavanju problema u nekoj oblasti, koje mu omogućava da rešava probleme koje ne može da reši većina drugih ljudi.

Ekspert koristi mentalne procese koji se ne mogu eksplicitno opisati u obliku algoritma, ali se važni aspekti primene ekspertskog znanja mogu opisati uz pomoć pravila za rešavanje problema.

Jednostavni primer saobraćajnog eksperta, koji daje pravila za bezbedan prelazak ulice u skladu sa signalizacijom semafora. Osnovna pravila su da se ulica prelazi kad je uključeno zeleno svetlo, a čeka kad svetli crveno. Ovo znanje se formalno može predstaviti pomoću produkcionih pravila:

```
IF svetlo semafora je zeleno
THEN akcija je pređi ulicu
IF svetlo semafora je crveno
THEN akcija je stoj
```

Produkciona pravila su najčešći oblik predstavljanja znanja u ekspertnim sistemima, jer se lako formulišu, razumljiva su i lako ih je menjati, pošto su

relativno nezavisna. Svako pravilo se sastoji iz dva dela, *uslova* (anrecedente ili premise) i *zaključka* (posledice, konsekvente ili akcije).

Osnovni oblik ili sintaksa pravila je:

IF <uslov>
THEN <zaključak>

U opštem slučaju uslov pravila je složen izraz, koji se sastoji od više pojedinačnih uslova povezanih logičkim konektivima AND (konjukcija) i OR (disjunkcija). Dobra praksa je da se izbegavaju kombinacije različitih konektiva u istom pravilu, jer su složeni konjuktivno-disjunktivni izrazi manje razumljivi.

IF <uslov 1>
AND <uslov 2>
…
AND <uslov n>
THEN <zaključak>

IF <uslov 1>
OR <uslov 2>
…
…
OR <uslov n>
THEN <zaključak>

Zaključci, posledice ili akcije pravila takođe mogu biti složeni od više elementarnih zaključaka:

IF <uslov>
THEN < zaključak 1>
< zaključak 2>
…
< zaključak *m*>

Uslov pravila se sastoji od elementarnih uslovnih izraza, koji se sastoje od (lingvističkog) objekta, relacije ili operatora i vrednosti. Npr. objekt 'svetlo semafora', relacija 'je' i neneumerička vrednost 'crveno', odnosno svojstvo 'težina pacijenta', relacija '>' i numerička vrednost 50. Primer pravila je:

```
IF    starost klijenta < 18
AND   podizanje gotovine > 1000
THEN  potpis roditelja = 'obavezan'
```

Zaključak takođe sadrži izraz sličnog oblika, s tim da se vrednost tog izraza dodeljuje lingvističkom objektu. Npr. kada je svetlo semafora zeleno, zaključak

ili akcija može biti 'akcija je preći ulicu'. Zaključak može biti i numerički izraz, npr.

```
IF    oporezivi prihod > 16.283
THEN  doprinos za zdravstvenu zaštitu = oporezivi prihod * 1.5/100
```

Pravila u bazi znanja mogu da opisuju relacije, preporuke, uputstva, strategije i heuristike, npr.

Relacija

```
IF    rezervoar za gorivo je prazan
THEN  automobil je nepokretan
```

Preporuka

```
IF    godišnje doba je jesen
AND   nebo je oblačno
AND   prognoza je slaba kiša
THEN  preporuka je poneti kišobran
```

Uputstvo

```
IF    automobil je nepokretan
AND   rezervoar za gorivo je prazan
THEN  akcija je napuni gorivo
```

Strategija

```
IF    automobil je nepokretan
THEN  akcija je proveri rezervoar za gorivo'; korak1 je završen
IF    korak1 je završen
AND   rezervoar za gorivo je pun
THEN  akcija je proveri akumulator; korak2 je završen
```

Heuristika

```
IF    materija je tečna
AND   pH materije < 6
AND   miris materije je sirće
THEN  sastav materije je sirćetna kiselina
```

8.2 Struktura ekspertnih sistema

Ekspertni sistemi imaju strukturu koja omogućava predstavljanje ekspertskog znanja i njegovu upotrebu u rešavanju nestrukturiranih problema. Komponente ekspertnih sistema su [1], [10], [11], [18]:

1. Baza znanja (*knowledge base*),
2. Mehanizam zaključivanja (*inference engine*),
3. Korisnički interfejs,
4. Pomoćni podsistemi:
   - podsistem za prikupljanje znanja (*knowlege acquisition*),
   - podsistem za davanje objašnjenja,

- podsistem za poboljšanje baze znanja,
- radni prostor (*workplace*), odnosno memorija.

Osnovni podsistemi svakog ekspertnog sistema su baza znanja, mehanizam zaključivanja i korisnički interfejs. Pomoćni podsistemi služe za bolje funkcionisanje, razvoj i održavanje ekspertnih sistema.

8.2.1 Baza znanja (Knowledga Base)

Najčešći način predstavljanja znanja u bazi znanja su produkciona pravila, logički izrazi i semantičke mreže.

Produkciona pravila (*production rules*) su izrazi oblika

IF *uslov* THEN *zaključak*

Produkciona pravila se često koriste u realizaciji ekspertnih sistema, jer su razumljiva ljudima. Njihova struktura odgovara strukturi logičkog mišljenja i konstrukcijama prirodnog jezika (ako... onda...). Omogućavaju jednostavno predstavljanje *neizvesnosti* i realizaciju probabilističkog zaključivanja. Dodavanje novih pravila u bazu znanja je jednostavno, zbog njihove relativne nezavisnosti.

Logički izrazi se takođe često koriste za predstavljanje znanja u ekspertnim sistemima, a njihov oblik u jeziku logičkog programiranja Prolog [61] je:

$H :- B_1, ..., B_n$

gde je H zaključak, a $B_1,..., B_n$ su elementi složenog uslova tvrdnje. Zarez označava konjukciju elemenata uslova, a oznaka ":-" implikaciju s desna u levo. U ovakvim sistemima se zaključivanje posmatra kao dokazivanje teorema.

Primer: Baza znanja u jeziku Prolog

Jezik *ProLog* je neproceduralni programski jezik. Naredbe jezika su logičke tvrdnje. Činjenice i pravila su su istinite tvrdnje. Upiti su tvrdnje čija se istinitost proverava u odnosu na stanje činjenica i pravila u trenutku postavljanja upita. Na slici je primer kratkog programa, koji ilustruje način interaktivne upotrebe jezika Prolog.

```
% Činjenice (facts)
lokacija(tf,kumodraska261).
lokacija(pfb,danijelova32).
studijski_program(finansije, pfb).
studijski_program(osiguranje, pfb).
studijski_program(er, tf).
status(tf, akreditovan).
predaje_na(petrovic, finansije).
predaje_na(jovanovic, osiguranje).
predaje_na(markovic, er).
```

```
% Pravila (Rules)
predaje_na(X,Y) :- studijski_program(Z,Y), predaje_na(X,Z).

%Upit: "Ko predaje na pfb?"
? predaje_na(X, pfb).
X= jovanovic;
X= petrovic
```

Činjenice i pravila su zadani, a korisnik postavlja upite, tako što kao prefiks neke tvrdnje unosi znak pitanja. Na osnovu činjenica i pravila zaključivanja, odgovor na upit je deduktivni zaključak koji nigde eksplicitno nije naveden. Sistem je odgovor, spisak profesora na PFB, dobio na zaključivši da su to profesori koji predaju na studijskim programima PFB.

Semantičke mreže su jedan od složenijih i manje formalnih načina predstavljanja znanja u ekspertnim sistemima. Koriste se za predstavljanje složenih grafovskih relacija, kao što su jezičke pojmovne strukture. Savremeni način predstavljanja znanja pomoću semantičkih mreža koristi XML.

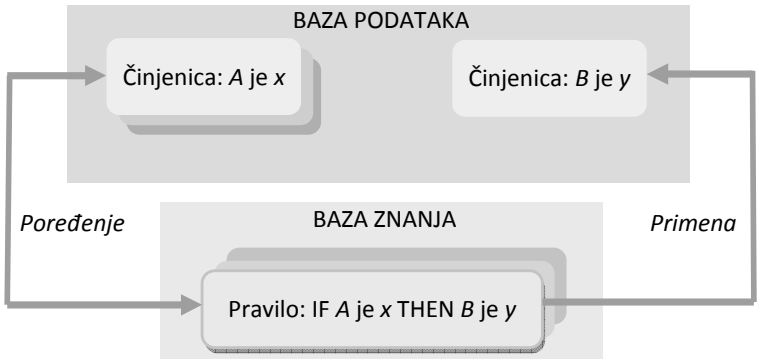
8.2.2 Mehanizam zaključivanja (Inference Engine)

Mehanizam zaključivanja je komponenta ekspertnog sistema koja stvarno realizuje operacije izvođenja zaključaka na osnovu trenutnog stanja baze podataka, odnosno radnog prostora ekspertnog sistema.

Metod zaključivanja zavisi od načina predstavljanja znanja, npr.

Predstavljanje znanja	Metod zaključivanja
Logička pravila	princip rezolucije (*resolution principle*)
Produkciona pravila	unapred (*forward*) i unazad (*backward*)
Semantičke mreže	nasleđivanje i posebni metodi
Pamćenje primera (slučajeva)	analogno, koristi se koncept sličnosti

U ekspertnim sistemima zasnovanim na pravilima, znanje se predstavlja skupom produkcionih pravila i skupom činjenica koji se odnose na trenutnu situaciju. Mehanizam zaključivanja poredi uslove pravila baze znanja (IF) s činjenicama i aktivira pravila za koje su uslovi zadovoljeni (THEN), odnosno izvršava predviđenu akciju, koja se može sastojati u promeni činjenica, Sl. 58.



Sl. 60. Osnovni ciklus mehanizma zaključivanja

Podudaranje uslova pravila s činjenicama proizvodi nizove zaključaka. Niz zaključaka pokazuje način na koji je ekspertni sistem došao do konačnog zaključka i predstavlja osnovu za njegovo obrazlaganje.

Na Sl. 61 je elementarni primer male baze znanja koja treba da pomogne u donošenju odluke o kupovini prenosnog računara (*notebook*) [1].

```
Rule 1:
 IF      namena=obrada teksta
 AND     način upotrebe=na putovanju
 THEN    težina=lagan

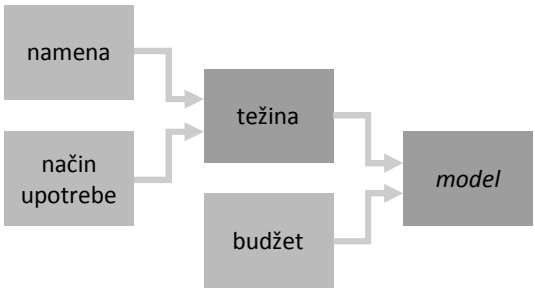
Rule 2:
 IF      namena=obrada teksta
 AND     način upotrebe=u kancelariji
 THEN    težina=nije bitna

Rule 3:
 IF      budžet<=2000
 AND     budžet>1000
 AND     težina=lagan
 THEN    model=Najnoviji model

Rule 4:
 IF      budžet<1000
 AND     težina=nije bitna
 THEN    model=Stariji model
```

Sl. 61. Primer male baze znanja za pomoć u kupovini prenosnih računara

Niz zaključaka, kojim se ustanovljava vrednost ciljne promenljive *model*, prikazan je na Sl. 62.



Sl. 62. Primer lanca zaključivanja

Dva osnovna načina izbora pravila koje treba primeniti su metod zaključivanja unapred (*forward chaining*) i zaključivanje unazad (*backward chaining*).

Zaključivanje unapred

U ovom pristupu se traži uslov pravila (*uslov*) koji odgovara činjenicama iz baze podataka i aktivira odgovararajuće pravilo. Metod zaključivanja unapred je tehnika koja prikuplja informacije i izvodi sve zaključke koji se iz njih mogu izvesti. Pošto mnogi od tih zaključaka nisu povezani s ciljem, kad je cilj ustanovljavanje neke konkretne činjenice u ekspertnim sistemima s velikim brojem pravila, zaključivanje unapred je neefikasno, pa se koristi zaključivanje unazad.

Zaključivanje unazad

Zaključivanje unazad usmereno je ka dostizanju cilja. Pretpostavi se zaključak, a mehanizam zaključivanja pokušava da prikupi činjenice koje ga potvrđuju. Mehanizam zaključivanja pretražuje zaključke pravila, u kojima traži pretpostavljenu činjenicu. Ako postoji i ako je primenjivo u odnosu na činjenice iz baze podataka, pravilo se primeni i ciljna tvrdnja je dokazana. Inače se pravilo smešta u stek i postavlja se novi (pod)cilj, da se dokaže istinitost uslova ovog pravila. Ovaj podcilj se dokazuje na isti način, pretraživanjem zaključaka pravila koji mogu da ga dokažu. Proces postavljanja u stek i pretraživanja se ponavlja sve dok ima pravila.

Baza znanja u obliku produkcionih pravila sa Sl. 61 može se koristiti za zaključivanje unapred Sl. 63 (A), kao i za zaključivanje unazad, Sl. 63 (B).

(A) Kod zaključivanja *unapred*, pravila se primenjuju redom, npr. Rule1, Rule2, Rule3, Rule4. Vrednosti promenljivih *osnovna namena*, *način upotrebe* i *budžet* unapred su zadane u radnoj memoriji sistema. Na osnovu tih vrednosti pravila Rule1 i Rule2 donose zaključak o vrednosti promenljive

težina. To je dovoljno pravilima Rule3 i Rule4 da donesu zaključak o modelu računara koji se preporučuje.

(B) Kod zaključivanja *unazad*, vrednosti promenljivih nisu poznate, pa sistem pronalazi neko pravilo koje ustanovljava traženi zaključak, a zatim postavlja korisniku pitanja o vrednostima promenljivih koje se koriste u uslovu pravila. Ako u primeru sa Sl. 61 sistem počne od pravila Rule3, koje u uslovu ima promenljive *težina* i *budžet*, postavlja pravilo u stek i zatim pita korisnika za vrednosti promenljivih iz uslova pravila. Zavisno od njihovih vrednosti donosi zaključak pravila Rule3 ili prelazi na sledeće pravilo. Za odgovore korisnika da traži *lagan* računar i da je budžet između 1.000 i 2.000, pravilo Rule3 se primeni i ustanovi zaključak: predlog kupovine najnovijeg modela. Pošto u steku nema drugih pravila, to je i konačni zaključak.

Primena oba metoda zaključivanja na iste činjenice koje sistem ima na raspolaganju daje isti zaključak, u ovom primeru preporuku za kupovinu istog modela prenosnog računara.

Ekspertni sistem može da prikaže tok zaključivanja pomoću podsistema za davanje objašnjenja.

Pravilo	(A) Zaključivanje unapred
	• vrednosti promenljivih zadane su unapred • pravila se aktiviraju redom, 1-2-3-4
1	**Osnovna namena?** a. obrada teksta b. komunikacije c. multimedija Odgovor: **a**
2	**Način upotrebe?** a. u kancelariji b. na putovanju Odgovor: **b**
3	**Koliki je budžet?** a. <1000 b. 1000..2000 c. >2000 Odgovor: **b**
	*Sistem preporučuje:**Najnoviji model***
	Obrazloženje: *lagan, pogodan za obradu teksta i putovanja, zadovoljava budžet*

Pravilo	(B) Zaključivanje unazad
	• vrednosti promenljivih nisu unapred poznate • prvo se aktivira neko pravilo koje zaključuje o modelu, npr. 3 (ili 4)
3	**Koliki je budžet?** a. <1000 b. 1000..2000 c. >2000 Odgovor: **b**
1	**Osnovna namena?** a. obrada teksta b. komunikacije c. multimedija Odgovor: **a**
2	**Način upotrebe?** a. u kancelariji b. na putovanju Odgovor: **b**
	*Sistem preporučuje:**Najnoviji model***
	Obrazloženje: *lagan, pogodan za obradu teksta i putovanja, zadovoljava budžet*

Sl. 63. Primeri zaključivanja unapred i unazad

Oba metoda zaključivanja odgovaraju metodima zaključivanja koje eksperti koriste za rešavanje različitih problema. Kad ekspert treba da zaključuje o prethodno prikupljenim informacijama, koristi zaključivanje unapred. Kad postavlja hipoteze i traži činjenice koje bi ih potvrdile ili opovrgle, koristi zaključivanje unazad. Ekspertni sistem DENDRAL je za određivanje molekularne strukture na osnovu masenih spektrograma (raspoloživih informacija) koristio zaključivanje unapred. Dijagnostički ekspertni sistemi, kao što je MYCIN, koriste zaključivanje unazad.

Mnoge ljuske ekspertnog sistema koriste kombinaciju zaključivanja unapred i unazad. Osnovni mehanizam je obično zaključivanje unazad, a zaključivanje unapred se aktivira kad se pojavi nova činjenica, da bi se najbolje iskoristio novi podatak.

8.2.3 Algoritam RETE

Zaključak pravila se dobija nakon evaluacije uslova jednog pravila ili većeg broja zavisnih pravila. U najgorem slučaju je potrebno ispitati *sve* činjenice iz radne memorije ekspertnog sistema (n_1) u odnosu na *sve* premise (n_2) *svih* pravila iz baze znanja (n_3), odnosno izvršiti $n_1 \cdot n_2 \cdot n_3$ računanja.

Zbog toga su sistemi sa velikim brojem pravila spori i naivna implementacija mehanizma zaključivanja, odnosno evaluacije pravila nije dovoljno efikasna.

Rešenje je efikasniji algoritam evaluacije RETE (lat. mreža), koji pronalazi zajedničke delove pravila iz baze znanja i konstruiše mrežu evaluacije pravila, na osnovu koje se vrše samo neophodna računanja, Sl. 63.

Rad algoritma će se prikazati na primeru male baze znanja sa 4 pravila:

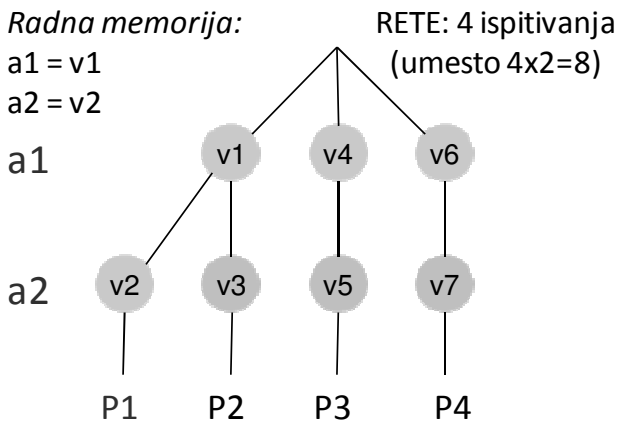
P1: **if** a1=v1 **and** a2=v2 **then** zaključak 1

P2: **if** a1=v1 **and** a2=v3 **then** zaključak 2

P3: **if** a1=v4 **and** a2=v5 **then** zaključak 3

P4: **if** a1=v6 **and** a2=v7 **then** zaključak 4

Algoritam RETE će za izbor pravila i ustanovljavanje vrednosti zaključka pravila izvršiti samo 4 evaluacije od maksimalnih 8 (4 pravila sa po dva ispitivanja), pošto na osnovu stanja memorije a1=v1 odustaje od evaluacije 2, P3 i P4, Sl. 63.



Sl. 64. Primer funkcionisanja algoritma RETE

8.2.4 Primena Bajesovog pravila

Bajesovo pravilo se može razmatrati kao mehanizam zaključivanja koji računa verovatnoće tvrdnji prema iterativnom algoritmu [14]:

```
Bayesian(S) {
 // S = {s1, .. , sn} simptomi
 for each di Pcurr(di)= P(di);
 repeat
   sj = s[1]; s = s\{s[1]};
   for each di   Pcurr(di)= Pcurr(di)*I(di|sj);
 until sj={};
 List top N diseases by Pcurr
}
```

$$P(d_i \mid s_1,...,s_m) = P(d_i) \cdot \frac{\prod_{j=1}^{m} P(s_j \mid d_i)}{P(s_1,...,s_m)}$$

Ovaj jednostavni algoritam podrazumeva da su s_i simptomi, koji se zadaju na ulazu, a d_i moguće dijagnoze. Verovatnoća dijagnoze d se, prema Bajesovoj formuli, računa množenjem uticaja na dijagnozu jednog po jednog simptoma s.

Računski problemi, koji se javljaju prilikom množenja malih decimalnih vrednosti, u praksi se rešavaju tako da se verovatnoće interno predstavljaju samo preko eksponenata baze 2 i koriste se samo za prikaz konačnog rezultata. Umesto klasičnog množenja verovatnoće $P(s|d)/P(s)$ u svakom prolazu koristi se celobrojna aritmetika eksponenata, odnosno faktor $I(d|s)$.

8.3 Vrste ekspertnih sistema

Prema načinu predstavljanja znanja i tehnologiji realizacije, ekspertni sistemi se mogu podeliti u više kategorija:

- Ekspertni sistemi *zasnovani na pravilima* (*rule-based*), koji za predstavljanje znanja koriste produkciona pravila;
- Ekspertni sistemi *zasnovani na okvirima* (*frames*), koji koriste relacione strukture za predstavljanje složenih koncepata;
- *Hibridni* ekspertni sistemi, koji istovremeno koriste više različitih načina predstavljanja znanja;
- Ekspertni sistemi *zasnovani na modelima* (*model-based*), koji za predstavljanje znanja koriste različite *modele*, npr. za simulaciju strukture i funkcionisanje sistema;
- Ekspertni sistemi *opšte namene* (*off-the-shelf*), koji su namenjeni korišćenju u široj oblasti primene;
- Ekspertni sistemi *posebne namene* (*custom-made*), koji su izrađeni prema posebnim zahtevima krajnjih korisnika;
- Ekspertni sistemi *u realnom vremenu* (*real-time*), koji su izgrađeni tako da svojim performansama zadovoljavaju stroga ograničenja vremena odziva sistema na promene u njegovom okruženju.

8.4 Alati za razvoj ekspertnih sistema

Softverski alati za razvoj ekspertnih sistema su:

- programski jezici opšte namene, kao što su C/C++, C#, F#, Java, Python, R, Prolog i LISP;
- ljuske ekspertnih sistema, npr. Exsys CORVID, CLIPS, Jess i Experise2Go;
- gotova rešenja za određene šire oblasti primene, kao što je oblast osiguranja, medicina ili planiranje (*Haley*, *ILOG*, *LPA VisiRule*).

Programski jezici opšte namene su algoritamski jezici kao C/C++, Java, C#, Python i R, funkcionalni jezici kao LISP, Erlang ili F#, kao i jezici za logičko programiranje, kao što je Prolog (detaljnije u poglavlju 11).

Ljuske ekspertnih sistema (*Expert Systems Shells*) su u stvari ekspertni sistemi sa praznom bazom znanja, ali je definisana njena struktura, odnosno način predstavljanja i upotrebe znanja. Baza znanja se razvija za svako specifično područje primene, pri čemu se koriste tehnike prikupljanja ili izvlačenja znanja (*knowledge acquisition*, *knowledge elicitation*) u koje spadaju i sistemi za istraživanje podataka, koji se često nazivaju i sistemi za otkrivanje znanja (*Data Mining/Knowledge Discovery*).

Primeri ljuski ekspertnih sistema su:

- EMYCIN (*empty* MYCIN) / TMYCIN,
- Exsys CORVID,
- CLIPS (*C Language Integrated Production System*),
- JESS (*Java Expert Systems Shell*),
- IBM ILOG (*Business Rule Management System*, C/C++, .Net, Java),
- Expertise2Go,
- JLog/JScriptLog (interpreter jezika Prolog u jezicima Java/JavaScript).

8.5 Primeri ekspertnih sistema za podršku odlučivanju

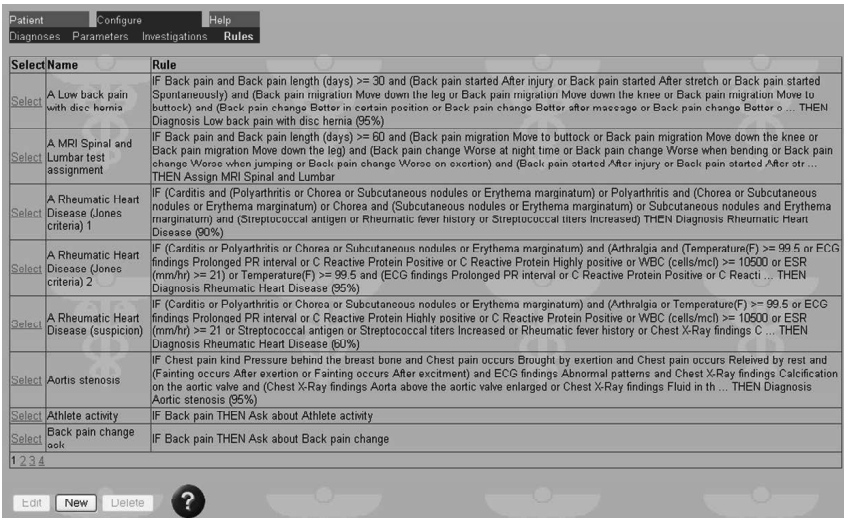
Ekspertni sistemi se koriste u brojnim oblastima za podršku donošenju odluka, odnosno rešavanju različitih vrsta problema, kao što su:

- *Interpretacija* - opis nastale situacije;
- *Predviđanje* - predviđanje posledica situacije;
- *Dijagnostika* - na osnovu posmatranja stanja;
- *Projektovanje* - uz poštovanje ograničenja;

- *Planiranje* - usklađenost s ciljevima;
- *Nadzor* - otkrivanje odstupanja/neispravnosti;
- *Opravke* - upravljanje otklanjanjem neispravnosti;
- *Učenje* - pomoć studentima;
- *Upravljanje* - funkcionisanjem sistema.

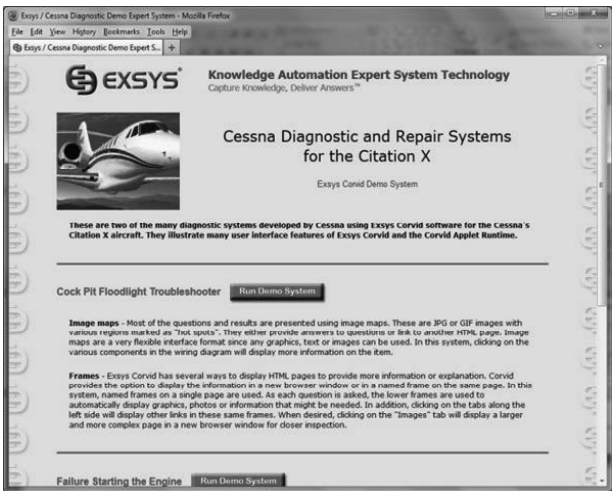
Ilustracija primene ekspertnih sistema za podršku odlučivanju u nekoliko različitih oblasti su:

- *Dijagnostika u medicini*, sistem Medical Expert www.med-expert.net;



- *Nadzor i sprečavanje zloupotreba na berzi*, sistem SONAR (*Securities Observation, News Analysis, and Regulation*);

- *Dijagnostika složenih tehničkih sistema*, Cessna Diagnostic and Repair Systems www.exsys.com/cessnatop.html.



Ekspertni sistemi se koriste za funkcionisanje Weba, a primeri ekspertnih sistema se mogu naći na:

- Exsys www.exsys.com (inteligentni Web baner, *banner with brains*);
- e2go www.expertise2go.com (više malih demonstracija).

Pitanja za ponavljanje

1. Objasnite osnovnu ideju ekspertnih sistema.
2. Navedite primere prvih ekspertni sistema
3. Koje vrste znanja i načine predstavljanja znanja razlikujemo?
4. Koje su osnovne forme logičkog zaključivanja?
5. Objasnite primenu Bajesove formule u ekspertnim sistemima.
6. Koje su osnovne komponente (podsistemi) ekspertnog sistema?
7. Koje su pomoćne komponente (podsistemi) ekspertnog sistema?
8. Zbog čega su produkciona pravila najčešći način predstavljanja znanja u ekspertnim sistemima?
9. Koje vrste mehanizama zaključivanja koriste ekspertni sistemi?
10. Kada se koristi zaključivanje unapred?
11. Koja je namena algoritma RETE?
12. Koji su osnovni alati za razvoj ekspertnih sistema?
13. Navedite bar tri konkretna primera alata za razvoj ekspertnih sistema.
14. Šta su ljuske ekspertnih sistema (*expert system shells*)?
15. Navedite bar tri konkretna primera ljuski ekspertnih sistema.

9 Metodi i tehnike mašinskog učenja

1. Pojam mašinskog učenja
2. Produkciona pravila i stabla odlučivanja
3. Induktivno učenje i procena tačnosti naučenog znanja
4. Veštačke neuronske mreže (ANN)
5. Metod nosećih vektora (SVM)
6. Zaključivanje po analogiji (CBR)
7. Programski alati mašinskog učenja i istraživanja podataka

U ovom poglavlju se objašnjava pojam mašinskog učenja, važniji metodi i tehnike mašinskog učenja i njihova primena u izgradnji inteligentnih sistema. Daje se pregled programskih alata istraživanja podataka i poslovne analitike, koji se uglavnom zasnivaju na primeni metoda mašinskog učenja.

9.1 Pojam mašinskog učenja

Učenje je jedno od obeležja inteligentnog ponašanja. Mašinsko ili automatizovano učenje je disciplina računarskih nauka koja se bavi metodima koji omogućavaju softverskim sistemima da samostalno uče na osnovu informacija koje imaju na raspolaganju.

Mašinsko ili automatizovano učenje (*machine learning*) može se definisati kao:

- Proces u kome sistem poboljšava svoje performanse na datom zadatku bez dodatnog programiranja [29];
- Skup procesa, gde spadaju: prikupljanje novog deklarativnog znanja, razvoj i usavršavanje motornih i saznajnih sposobnosti kroz praksu, strukturiranje postojećeg znanja i otkrivanje novih činjenica i teorija posmatranjem i aktivnim eksperimentisanjem [28];
- Ustanovljavanje zavisnosti u raspoloživim podacima [25].

Osnovne forme mašinskog učenja

Mašinsko učenje se može posmatrati kroz dve osnovne forme kao [28]:

- *prikupljanje znanja* (*knowledge acquisition*), koje predstavlja učenje nove, simboličke informacije, tako da se ona može efektivno primeniti. Tako čovek uči teorijska znanja, npr. fiziku.
- *uvežbavanje* (*training*), koje obuhvata poboljšavanje nekog stečenog znanja, mentalne ili motorne koordinacije, kroz praktično ponavljanje i korekciju odstupanja od željenog ponašanja. Tako čovek uči *veštine*, npr. vožnju bicikla ili sviranje klavira, pri čemu je *prikupljanje znanja* prva faza učenja.

Smatra se da je učenje kod čoveka mešavina obeju formi, s tim da mentalne aktivnosti potenciraju prvu, a motorne aktivnosti u većoj meri drugu formu učenja [28].

Strategije mašinskog učenja

Mašinsko učenje može se podeliti prema strategiji učenja, odnosno vrsti i obimu zaključivanja koje metodi učenja koriste, na [28]:

1. Učenje memorisanjem (*rote learning*)

   Nema zaključivanja ni transformacije znanja (programiranje, baze podataka).

2. Učenje na osnovu rečenog (*learning by being told*)

   Vrši se samo transformacija u internu formu predstavljanja i integracija sa postojećim znanjem, koje sistem upotrebljava bez dodatnog programiranja.

3. Učenje po analogiji (*learning by analogy*)

   Potreban je veći obim zaključivanja, pošto se novo znanje stiče transformacijom i proširivanjem postojećeg znanja u takav oblik da se može upotrebiti za rešavanje novih problema, koji su u određenoj meri slični već rešenim problemima.

4. Učenje na osnovu primera (*learning by examples*) ili učenje s učiteljem

   Zahteva *induktivno zaključivanje*, odnosno generalizaciju rešenih primera i kontraprimera neke klase pojava (pojma, koncepta).

5. Učenje posmatranjem i samostalnim otkrivanjem (*learning by observation and discovery*) ili učenje bez učitelja

   Zahteva najviše zaključivanja, jer sistem mora samostalno da otkriva nove značajne klase objekata (pojmove, koncepte), postavlja i proverava hipoteze, odnosno stvara teorije.

Induktivno mašinsko učenje obuhvata poslednje dve strategije učenja, odnosno:

- učenje na osnovu primera (učenje s učiteljem, *supervised learning*) i
- učenje posmatranjem i samostalnim otkrivanjem (učenje bez učitelja, *unsupervised learning*).

Učenje na osnovu primera je klasa metoda induktivnog učenja, koja ima brojne praktične primene i veliki broj različitih metoda. U tom pristupu se na osnovu analize i generalizacije već *rešenih* primera i kontraprimera neke klase pojava (pojma, koncepta), dolazi do pravila, teorije ili opisa koji objašnjava (gotovo) sve primere i (gotovo) nijedan kontraprimer.

Obučavajući primeri se obično predstavljaju vektorima vrednosti atributa, koji mogu biti numerički i nenumerički. Skup vrednosti pojedinačnog atributa može biti potpuno uređen, parcijalno uređen ili neuređen. Npr. skup numeričkih vrednosti potpuno je uređen, skup imena ili oznaka je neuređen.

Pri tome znanje o problemu, kao i naučeno znanje, može biti egzaktno ili približno (aproksimativno), a može se predstaviti u obliku stabala odlučivanja, produkcionih pravila, veštačkih neuronskih mreža, logičkih pravila i drugih načina predstavljanja.

Mašinsko učenje klasifikacija

Induktivno učenje se može koristiti za učenje pravila predviđanja numeričkih (*regresija*) ili nenumeričkih vrednosti, što predstavlja učenje pravila razvrstavanja objekata u unapred definisane klase (*klasifikacija*).

Sami objekti mogu biti problemi odlučivanja, npr. podaci o ranijim ili novim pacijentima koje treba razvrstati u neke klase, npr. na zdrave i obolele. Klase se mogu posmatrati kao *odluke*, npr. dijagnoze oboljenja, izabrani proizvodi ili kupovina/prodaja akcija.

Induktivno učenje koncepata

Induktivno učenje koncepata je indukcija logičkih zakonitosti koje su razumljive ljudima, odnosno koje zadovoljavaju princip razumljivosti (*principle of comprehensibility*). Pri tome razumljivost novog, induktivno generisanog znanja ne zavisi samo od načina njegovog predstavljanja (stabla, pravila, mreže), već i od drugih faktora, kao što su složenost znanja i saglasnost novog znanja s postojećim znanjem.

Mašinsko učenje razumljivog znanja

Učenje razumljivog znanja odnosi se na učenje logičkih zakonitosti, koje treba da budu razumljive čoveku. Pri tome se razlikuju tri stepena razumljivosti mašinski naučenog znanja [Michalski]:

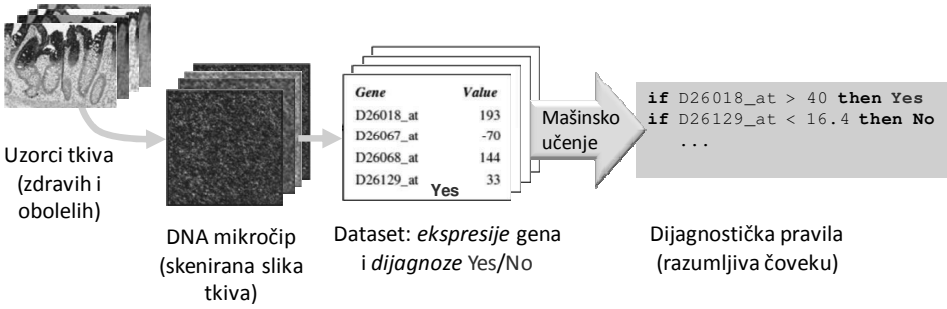
1. znanje može koristiti samo uz pomoć računara (najniži stepen razumljivosti),
2. znanje se može koristiti uz pomoć računara, koji može da obrazlaže odluke (daje objašnjenja), npr. kao u ekspertnim sistemima,
3. naučeno znanje se može koristiti bez pomoći računara, npr. u komunikaciji između ljudi, objavljivanjem naučnog rezultata i za donošenje odluka.

Modeli kao što su veštačke neuronske mreže imaju prvi, najniži stepen razumljivosti, jer se znanje može koristiti samo uz pomoć računara.

Modeli kao što su produkciona pravila i stabla odlučivanja imaju treći, *najviši* stepen razumljivosti, pošto ga ljudi mogu koristiti za razmenu informacija i donošenje odluka bez pomoći računara.

Ilustracija: Mašinsko učenje razumljivog znanja na osnovu primera

Savremeni primer primene mašinskog učenja je učenj dijagnostičkih pravila na osnovu *genskih ekspresija*. Genske ekspresije su intenziteti osvetljenosti skeniranih uzoraka tkiva nanesenih na posebne DNA mikročipove, Sl. 67. Vrednost intenziteta odgovara aktivnosti pojedinih gena, što se koristi za otkrivanje njihove veze sa pojedinim oboljenjima.



Sl. 65. Primer induktivnog učenja razumljivog znanja

Najpoznatiji metodi predstavljanja znanja razumljivog čoveku su produkciona pravila i stabla odlučivanja. Koriste se u sistemima zasnovanim na znanju (*knowledege based systems, expert systems*) jer omogućavaju razumljivu komunikaciju sa ljudima, posebno kod donošenja i obrazlaganja odluka, kao što su dijagnoze oboljenja ili kvarovi u tehničkim sistemima [25], [56], [57].

9.2 Produkciona pravila i stabla odlučivanja

Produkciona pravila i stabla odlučivanja su metodi učenja znanja koje je načelno razumljivo čoveku i može se direktno prevesti u prirodni jezik. Osim najvišeg stepena razumljivosti, metodi učenja stabala i pravila omogućavaju veliku brzinu učenja, prihvatljivu tačnost i poseduju ugrađenu sposobnost selekcije relevatnih atributa.

9.2.1 Metodi induktivnog učenje stabala odlučivanja

Osnovna ideja je preuzeta od sistema CLS (*Concept Learning System*, 1966). U psihološkim eksperimntima ustanovljeno je da ljudi stvaraju nove (diskretne) koncepte na osnovu nestrukturiranih podataka tako što razbijaju podatke i koncepte u podskupove sličnih karakteristika, sve dok se ne dobiju podskupovi

primera istih svojstava. Ovakva rekurzivna podela na podskupove (*recursive partitioning*) daje hijerarhiju (stablo).

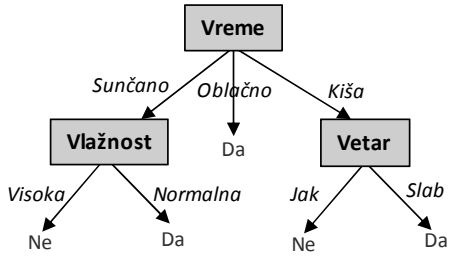
Na slici je primer skupa podataka s primerima odluka (Da/Ne) zasnovanih na vremenskim prilikama, opisanih vektorima vrednosti atributa <Vreme, Temperatura, Vlažnost, Vetar>:

Dan	Vreme	Temperatura	Vlažnost	Vetar	Odluka
1	Sunčano	Toplo	Visoka	Slab	Ne
2	Sunčano	Toplo	Visoka	Jak	Ne
3	Oblačno	Toplo	Visoka	Slab	Da
4	Kiša	Umereno	Visoka	Slab	Da
5	Kiša	Hladno	Normalna	Slab	Da
6	Kiša	Hladno	Normalna	Jak	Ne
7	Oblačno	Hladno	Normalna	Jak	Da
8	Sunčano	Umereno	Visoka	Slab	Ne
9	Sunčano	Hladno	Normalna	Slab	Da
10	Kiša	Umereno	Normalna	Slab	Da
11	Sunčano	Umereno	Normalna	Jak	Da
12	Oblačno	Umereno	Visoka	Jak	Da
13	Oblačno	Toplo	Normalna	Slab	Da
14	Kiša	Umereno	Visoka	Jak	Ne

Algoritam rekurzivnog particioniranja CLS ima oblik:

1. *Heuristički* se izabere kriterijum (atribut) najpogodniji za razbijanje skupa primera koncepta na podskupove;
2. Skup primera se razbija na podskupove prema vrednostima izabranog kriterijuma (atributa);
3. Ponavlja se postupak za svaki od podskupova, sve dok se ne dobiju poskupovi primera iste klase.

Algoritam za prikazani skup od 14 primera daje stablo odlučivanja:



Atribut po kome se vrši razbijanje dobija se izborom na osnovu heuristika kao što su *Information gain*, *Gini ideks* i sl [26], [31]. Algoritam CLS na svakom koraku bira atribut koji prilikom upotrebe najviše smanjuje neizvesnost klasifikacije, odnosno za koji je najveća razlika u količini informacija pre i nakon izvršene particije:

$$I(C|V) = H(C) - H(C|V)$$

gde su

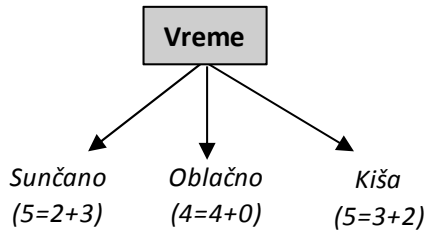
$$H(C|V) = \sum_{j=1}^{V} P(V_j) \log_2 P(C|V_j)$$

$$H(C) = -\sum_{i=1}^{C} P(C_i) \log_2 P(C_i)$$

Na prvom koraku razlika u količini informacija za svaki atribut je, prema slici:

I(*Vreme*) = 0,247 bita
I(*Temperatura*) = 0,029 bita
I(*Vlažnost*) = 0,152 bita
I(*Vetar*) = 0,048 bita.

Za koren stablja izabran je atribut *Vreme*, kao najinformativniji.



Razvijeni su brojni algoritmi učenja stabala odlučivanja koji koriste metod rekurzivne particije primera, npr.

- ID3 (*Iterative Dichotomiser 3*, Quinlan, 1979)

  Dozvoljava upotrebu samo nenumeričkih atributa, a ciljni atribut je takođe nenumerički.

- CART (*Classification and Regression Trees*, Breiman, Friedman, Olshen, Stone, 1982) [30]

  Sistem je razvila grupa profesora statistike univerziteta Berkeley i Stanford. Dozvoljava upotrebu nenumeričkih i numeričkih atributa i za ciljni atribut, pa se koristi za učenje klasifikacionih i regresionih stabala. U sistemu *Weka* implementirana verzija za učenje klasifikacija pod nazivom *SimpleCart*.

- Asisstant-86 (Cestnik, Kononenko, Bratko, 1984)

  Sistem za učenje stabala odlučivanja razvijen na Institutu Jože Štefan u Ljubljani, Jugoslavija. Dozvoljava numeričke i nenumeričke atribute.

- C45 (Quinlan, 1993) [26], [31]

  Algoritam za učenje stabala odlučivanja C4.5 koristi nenumeričke i numeričke atribute, dok je ciljni atribut nenumerički. U sistemu Weka implementirana je verzija 4.8, pod nazivom *J48*.

9.2.2 Metodi induktivnog učenja pravila odlučivanja

Algoritmi učenja pravila (*rule learning*) najčešće koriste strategiju pokrivanja skupa (*set covering*), popularno "odvoj pa savladaj" (*separate-and-conquer*). Naziv je nastao zbog toga što se opis klase obučavajućih primera ili koncepta dobija sukcesivnom izgradnjom pravila koja pokrivaju veliki broj pozitivnih primera jedne klase, a malo ili nijedan negativan primer koji pripada nekoj od ostalih klasa, nakon čega se obuhvaćeni primeri uklanjaju iz razmatranja/odvajaju (*separating-out*). Pokrivanje i odvajanje se ponavlja za preostale primere razmatrane klase, sve dok se ne opišu svi primeri. Postupak pokrivanja i odvajanja zatim se ponavlja za ostale klase obučavajućih primera.

Poznati algoritmi učenja pravila metodom pokrivanja skupa su:

- AQ (Michalski, 1969)
- CN2 (Clark-Niblett, 1989) [32]
- Ripper (Cohen, 1995) [33]

U sistemu *Weka* implementirana je verzija sistema Ripper u jeziku Java pod nazivom *JRip*.

Poseban učenja pravila je transformacija stabala odlučivanja u pravila, koji koristi metod *C45Rules* (Quinlan,1993), u sistemu Weka realizovan pod nazivom *PART* (*PARtial Tree*).

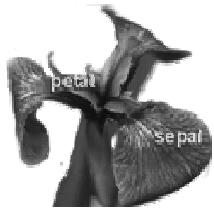
Postoje i metodi učenja pravila, koji se zasnivaju na prethodnom učenju drugih modela predstavljanja znanja, kao što su modeli neuronskih mreža i nosećih vektora, za koje se zatim generiše razumljiv opis u obliku logičkih pravila.

Primer: Poređenje učenja podukcionih pravila i stabala odlučivanja

Stabla odlučivanja su specijalni slučaj produkcionih pravila. Na Sl. 66 prikazani su rezultati učenja stabala odlučivanja i pravila metodima PART i J48 iz sistema *Weka* na primerima klasifikacije cveta perunike (problem je objavio poznati statističar Fišer još 1938. godine).

Sistemu su zadane dimenzije dveju vrsta cvetnih latica za 150 primera cveta perunike, koji pripadaju jednoj od tri vrste perunike, lat. Iris Setosa, Iris Virginica i Iris Versicolor. Naučeno stablo ili produkciona pravila treba da omoguće klasifikaciju cveta samo na osnovu mera njegovih delova, latica (*petal*) i čašičnih listića (*sepal*).

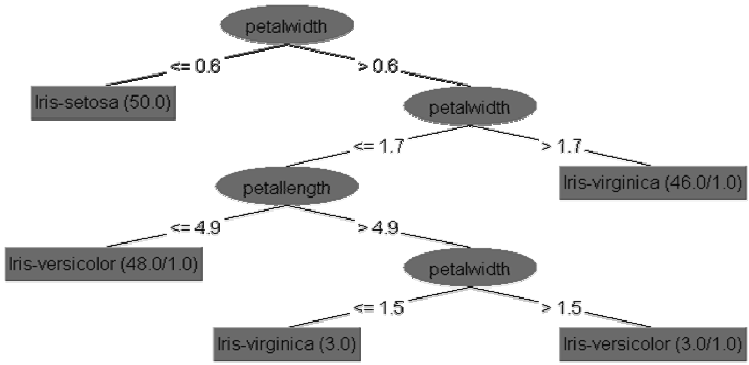
No.	sepallength Numeric	sepalwidth Numeric	petallength Numeric	petalwidth Numeric	class Nominal
1	5.1	3.5	1.4	0.2	Iris-setosa
2	4.9	3.0	1.4	0.2	Iris-setosa
3	4.7	3.2	1.3	0.2	Iris-setosa
4	4.6	3.1	1.5	0.2	Iris-setosa
5	5.0	3.6	1.4	0.2	Iris-setosa
6	5.4	3.9	1.7	0.4	Iris-setosa
7	4.6	3.4	1.4	0.3	Iris-setosa
8	5.0	3.4	1.5	0.2	Iris-setosa



Produkciona pravila za primer Iris

```
(petallength <= 1.9) => class=Iris-setosa (50.0/0.0)
(petalwidth >= 1.7) => class=Iris-virginica (48.0/2.0)
(petallength >= 5) => class=Iris-virginica (5.0/1.0)
 => class=Iris-versicolor (47.0/0.0)
```

Ekvivalentno stablo odlučivanja za primer Iris

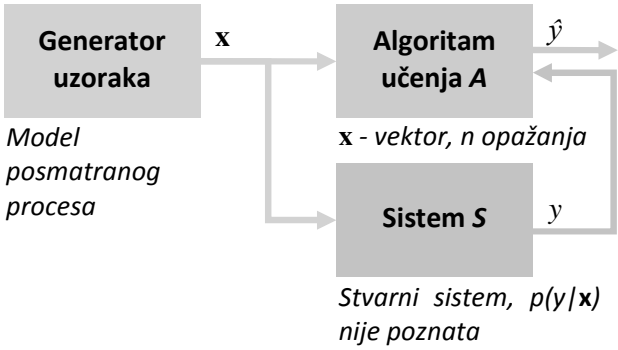


Sl. 66. Primer IRIS - induktivno učenje pravila i stabala odlučivanja

9.3 Induktivno učenje i procena tačnosti naučenog znanja

Da bi bilo korisno, naučeno znanje mora biti tačno i razumljivo. Tačnost predviđanja (*predictive accuracy*) predstavlja procenat uspešnosti klasifikacije novih, nerazmatranih primera korišćenjem naučenih pravila. Proces induktivnog učenja na osnovu primera može se predstaviti pomoću stohastičkog modela, Sl. 67 [25]. U prikazanom modelu, generator uzoraka modelira proces posmatranja neke pojave: uzorkuje nepoznatu distribuciju $p(\mathbf{x})$ i generiše slučajne vektore $\mathbf{x} \in \Re^d$ (opservacije, obučavajuće primere) za algoritam učenja *A*, dok sistem *S* za svaki slučajni vektor **x** na osnovu fiksne nepoznate uslovne funkcije gustine distribucije $p(y|\mathbf{x})$ generiše izlaznu vrednost *y*.

Algoritam učenja koristi konačan broj od *n* opservacija sistema *S* za formiranje što tačnije aproksimacije njegovih izlaza. Skup funkcija koje se mogu naučiti i način ocene najbolje aproksimacije definisani su unapred, na osnovu znanja o sistemu *S* i konstrukcije algoritma učenja.



Sl. 67. Model induktivnog učenja na osnovu primera

Tačnost aproksimacije *A* (*accuracy*) funkcije može se izračunati iz rezultata statističkog eksperimenta (učenja i testiranja) kao količnik broja ispravno i pogrešno klasifikovanih primera:

$$A = \frac{TP + TN}{TP + FP + TN + FN}$$

gde su:

TP (*True Positive*) - broj ispravno klasifikovanih pozitivnih primera,
TN (*True Negative*) - broj ispravno klasifikovanih negativnih primera,
FP (*False Positive*) - broj pogrešno klasifikovanih pozitivnih primera,
FN (*False Negative*) - broj pogrešno klasifikovanih negativnih primera.

Procena (estimacija) tačnosti predviđanja naučene funkcije vrši se statističkim eksperimentom. Neka je *V* skup neklasifikovanih primera, *Y* skup

oznaka klasa, a $X=V\times Y$ prostor klasifikovanih primera. Neka je $D=\{x_1,...,x_n\}$ skup od n klasifikovanih primera $x_i=<v_i,y_i>$, $v_i\in V$, $y_i\in Y$, a $C(D,v)$ oznaka klase koju primeru v dodeljuje skup pravila C induktivno naučen na skupu D. Tada se tačnost skupa pravila definiše kao verovatnoća ispravne klasifikacije slučajno izabranog primera $x=<v,y>$, odnosno $A=P(C(v)=y)$, pri čemu se podrazumeva ista distribucija primera kao kod obučavajućeg skupa [45].

Najčešći metodi estimacije tačnosti predviđanja su: metod testnog uzorka (*holdout method*), metod unakrsne validacije (*cross-validation*, CV) i metod izostavljanja jednog primera (*leave-one-out*, LOO):

1. *Metod testnog uzorka* deli primere na dva međusobno nezavisna podskupa - obučavajući skup i testni ili *holdout* skup primera. Obično se 2/3 primera koristi za učenje, a 1/3 za ocenu tačnosti predviđanja.

   Nakon učenja, skup pravila se koristi za klasifikaciju primera iz testnog skupa. Ako je D skup primera, D_H testni, a $D_T=D\backslash D_H$ obučavajući skup, koji imaju h i t primera, tačnost predviđanja metodom testnog uzorka P_H je:

   $$P_H = \frac{1}{h}\sum_{\langle v_i,y_i\rangle\in D_H}\begin{cases}1, & C(D_T,v_i)=y_i\\ 0, & C(D_T,v_i)\neq y_i\end{cases}$$

   Estimacija tačnosti predviđanja je slučajni broj, koji zavisi od podele primera na obučavajući i testni skup. Postupak se može ponaviti k puta (*random subsampling*) i tačnost se može oceniti kao srednja vrednost.

   Pošto za učenje nisu upotrebljeni svi primeri, ovaj metod daje pesimističku ocenu tačnosti predviđanja i neefikasno koristi raspoloživi skup primera.

2. *Metod unakrsne validacije* ili rotaciona estimacija slučajno deli skup primera D na k međusobno isključivih podskupova primera $D_1,D_2,...,D_k$ približno iste veličine. Postupak učenja i ocenjivanja se ponavlja k puta, svaki put koristeći po jedan podskup D_i kao testni skup. Ukupna tačnost računa se kao odnos broja ispravnih klasifikacija i ukupnog broja primera u skupu D:

   $$P_{CV} = \frac{1}{n}\sum_{\langle v_i,y_i\rangle\in D}\begin{cases}1, & C(D\backslash D_i,v_i)=y_i\\ 0, & C(D\backslash D_i,v_i)\neq y_i\end{cases}$$

   Estimacija tačnosti predviđanja unakrsnim ocenjivanjem je slučajni broj koji zavisi od podele primera na podskupove. Kompletna unakrsna validacija je srednja vrednost svih $\binom{m}{m/k}$ mogućnosti izbora m/k od ukupno m primera.

   Za stabilne algoritme induktivnog učenja, koji generišu pravila s istim predviđanjima nezavisno od redosleda obučavajućih primera, estimacija tačnosti predviđanja unakrsnom validacijom je *nepristrasna*. U praksi se najčešće koristi vrednost k=10, odnosno desetostruka krosvalidacija.

3. *Metod izostavljanja jednog primera* granični je slučaj unakrsne validacije, koji se dobija za *k*=1. Učenje se vrši *n* puta, tako što se izostavi jedan slučajno izabran primer i na njemu testira naučeno znanje. Postupak je skoro objektivan, ali računski zahtevan.

Primer: Mašinsko učenje pravila klasifikacije

Problem predviđanja stečaja kompanije opisan je modelom sa 5 atributa (količnika) [1]:

1. *Obrtni kapital/ukupna aktiva* (WC/TA)
2. *Zadržana dobit/ukupna aktiva* (RE/TA)
3. *Dobit pre kamata i poreza/ukupna aktiva* (EBIT/TA)
4. *Tržišna vrednost kapitala/ukupni dug* (MVE/TD)
5. *Prodaja/ukupna aktiva* (S/TA)

Klasifikacija BR/NB vrši se u 2 klase (0 - stečaj, 1 - neće ići u stečaj). Obučavajući skup se sastoji od podataka o 20 firmi, za koje su izračunati pokazatelji i uneti podaci o uspešnosti, odnosno stečaju.

Firm	WC/TA	RE/TA	EBIT/TA	MVE/TD	S/TA	BR/NB
1	0.165	0.1192	0.2035	0.813	1.6702	1
2	0.1415	0.3868	0.0681	0.5755	1.0579	1
3	0.5804	0.3331	0.081	1.1964	1.3572	1
4	0.2304	0.296	0.1225	0.4102	3.0809	1
5	0.3684	0.3913	0.0524	0.1658	1.1533	1
6	0.1527	0.3344	0.0783	0.7736	1.5046	1
7	0.1126	0.3071	0.0839	134	1.5736	1
8	0.0141	0.2366	0.0905	0.5863	1.4651	1
9	0.222	0.1797	0.1526	0.3459	1.7237	1
10	0.2776	0.2567	0.1642	0.2968	1.8904	1
11	0.2689	0.1729	0.0287	0.1224	0.9277	0
12	0.2039	-0.0476	0.1263	0.8965	1.0457	0
13	0.5056	-0.1951	0.2026	0.538	1.9514	0
14	0.1759	0.1343	0.0946	0.1955	1.9218	0
15	0.3579	0.1515	0.0812	0.1991	1.4582	0
16	0.2845	0.2038	0.0171	0.3357	1.3258	0
17	0.1209	0.2823	-0.0113	0.3157	2.2319	0
18	0.1254	0.1956	0.0079	0.2073	1.489	0
19	0.1777	0.0891	0.0695	0.1924	1.6871	0
20	0.2409	0.166	0.0746	0.2516	1.8524	0

U primeru će se automatizovano naučiti pravila predviđanja stečaja (A) sa obrazloženjem i (B) bez obrazloženja.

(A) Predviđanje stečaja sa obrazloženjem, metod C45 Rules (PART)

Tačnost naučenog skupa pravila procenjena je na 80% metodom unakrsne validacije (*crossvalidation*, CV). Naučeni skup pravila je:

```
PART decision list
------------------
RE/TA <= 0.2038: 0 (11.0/2.0)

: 1 (9.0/1.0)

Number of Rules  :   2
```

Obrazloženje ili opis se može dati jednostavnim prevođenjem naučenih pravila u prirodni jezik:

"Ako je odnos '*Zadržana dobit / Ukupna aktiva'* (RE/TA) veća od 0,2038 firma neće ići u stečaj;

Inače firma ide u stečaj."

(B) Predviđanje stečaja bez obrazloženja, metod ANN (Multilayer Perceptron)

Tačnost naučene veštačke neuronske mreže je procenjena na 100%.

```
Sigmoid Node 0
    Inputs    Weights
    Threshold    0.2371217113969034
    Node 2    -1.1628334833591436
    Node 3    5.604749511353181
    Node 4    -4.881596560417678
Sigmoid Node 1
    Inputs    Weights
    Threshold    -0.22668980982630868
    Node 2    1.1444939668327085
    Node 3    -5.608403146810656
    Node 4    4.882711122017258
Sigmoid Node 2
    Inputs    Weights
    Threshold    -0.1724610782503284
    Attrib WC/TA    -0.1794107913471712
    Attrib RE/TA    1.3806454084725033
    Attrib MVE/TD    0.379066064551664
    Attrib S/TA    -0.16378542159685397
   ....
Class 0
    Input
    Node 0
Class 1
```

```
Input
Node 1
```

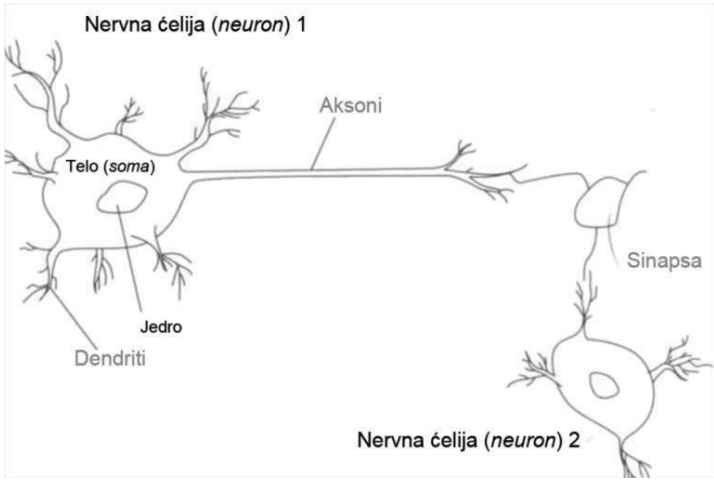
Neuronska mreža nema razumljivo obrazloženje, ali je tačnost predviđanja veća nego kod razmotrenih razumljivih metoda, u ovom primeru čak 100%.

Ovakvi metodi su pogodni kada nije potreban visok stepen razumljivosti. To je slučaj kod obučavanja, učenja znanja koje ljudi inače ne razumeju, kao što su znanja vezana za percepciju ili motorne veštine, npr. učenje prepoznavanja otisaka prstiju, fizionomija ljudi ili kretanje robota.

9.4 Veštačke neuronske mreže (ANN)

Ideja veštačkih neuronskih mreža (*Artificial Neural Networks*, ANN) nastala je proučavanjem funkcionisanja nervnog sistema, pre svega kore velikog mozga kod čoveka.

Uprošćena struktura nervne ćelije mozga prikazana je na Sl. 68.



Sl. 68. Struktura nervne ćelije

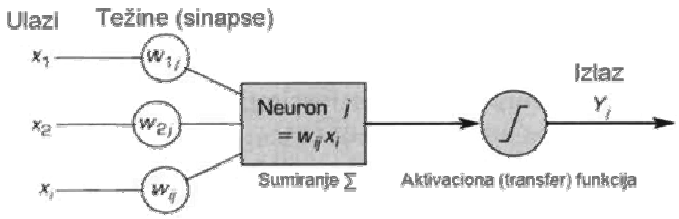
Nervne ćelije mozga, *neuroni*, povezani su nervnim vlaknima, koja omogućavaju razmenu elektro i hemijskih signala. Sastoje se od *dendrita*, koji predstavljaju ulaze za prikupljanje signala iz drugih neurona, *aksona*, koji predstavljaju aktivacione izlaze za druge neurone i *sinapsi*, koje predstavljaju međusobne veze neurona u kojima se memorišu informacije.

Model veštačkog neurona, koji bi funkcionalno odgovarao ćelijama ljudskog mozga, predložen je još 1943. godine (McCulloch, McCulloch & Pitts). Računarski model mreže veštačkih neurona nazvan *perceptron* (Rosenblatt) realizovan je 1957. godine [1], [4]. Veštačke neuronske mreže počinju se

intenzivno izučavati sredinom osamdamdesetih godina prošlog veka, što je dovelo do njihove uspešne praktične primene na brojnim područjima, uključujući i podršku odlučivanju. Treba napomenuti da veštačke neuronske mreže ne predstavljaju model funkcionisanja mozga.

Veštački neuron i neuronska mreža

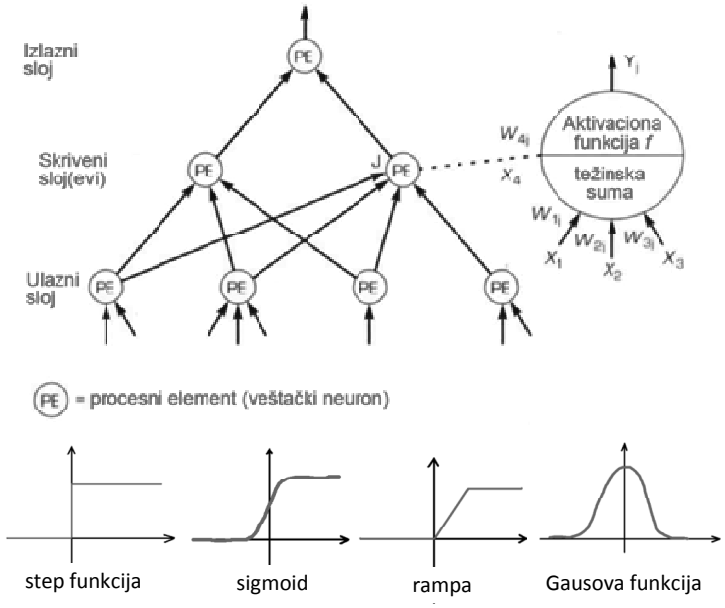
Veštački neuron, Sl. 69, predstavlja osnovni procesni element (PE) veštačke neuronske mreže, Sl. 70.



Sl. 69. Model veštačkog neurona

Veštački neuron ima *n* ulaza, čije vrednosti se koriste sa zadanim težinama w_i, i jedan izlaz, čija se vrednost dobija pomoću neke operacije kombinovanja ili sumiranja ulaznih signala, korišćenjem njihovih težina. Konačni izlaz neurona dobija se nakon primene *aktivacione funkcije*, koja normalizuje dobijeni rezultat.

Aktivaciona funkcija je najčešće neka nelinearna funkcija $f(I)$, koja ograničava izlaz $I=\sum_i w_i x_i$, npr. step funkcija, sigmoid, rampa ili Gausova funkcija, Sl. 70.



Sl. 70. Mreža veštačkih neurona

Neuronska mreža može biti *jednoslojna*, kod koje su ulazni neuroni direktno povezani s izlazima (perceptron), ili *višeslojna*. Višeslojna veštačka neuronska mreža se sastoji od ulaznog (*input layer*), izlaznog (*output layer*) i jednog ili više skrivenih slojeva neurona (*hidden layers*).

Učenje veštačkih neuronskih mreža (ANN)

Učenje ili obučavanje neuronske mreže sastoji se u određivanju optimalnih težina elemenata veštačke neuronske mreže, koji omogućavaju što tačniju klasifikaciju budućih primera, odnosno donošenje ispravnih odluka.

Veštačke neuronske mreže koriste se za induktivno mašinsko učenje na osnovu primera, i to učenje s učiteljem (*supervised learning*), učenje bez učitelja ili samoobučavanje (*unsupervised learning*) i učenje s podsticajem (*reinforcement learning*).

Proces učenja s učiteljem ili učenja pravila klasifikacije je iterativni postupak, koji se može opisati sledećim koracima:

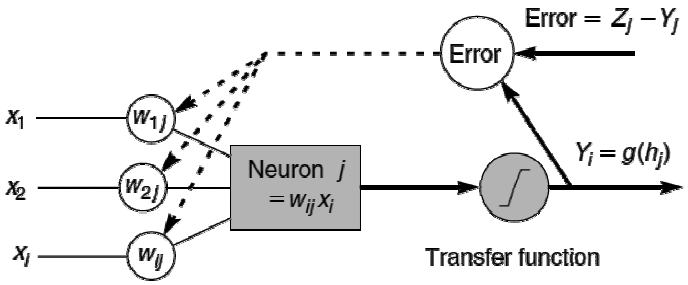
1. Izračunati privremene izlaze.
2. Uporediti izlaze s traženim izlazima iz obučavajućih primera i izračunati razliku, koja predstavlja grešku Δ.
3. Prilagoditi težine i ponoviti ceo proces.

Postupak okončava kad se postigne zadovoljavajući nivo performansi, koji se meri greškom klasifikacije, odnosno tačnošću predviđanja.

Algoritam propagacije greške

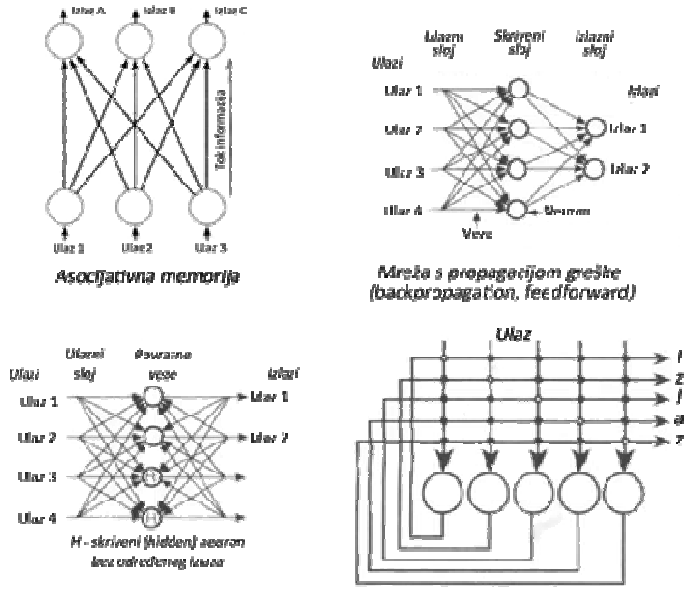
Poseban značaj u učenju veštačkih neuronskih mreža ima algoritam propagacije greške (*backpropagation*, *back-error propagation*), Sl. 71 [1], koji omogućava prilagođavanje težina propagacijom unazad razlike između očekivanih izlaza obučavajućih primera i trenutne klasifikacije mreže:

1. Inicijalizuj težine slučajnim vrednostima.
2. Učitaj ulazni vektor i njegov zadani izlaz.
3. Izračunaj trenutni izlaz mreže, od ulaza ka izlazu.
4. Izračunaj grešku klasifikacije na obučavajućim primerima (Δ).
5. Menjaj težine *unazad*, od izlaznog sloja preko skrivenih slojeva ka ulazu.



Sl. 71. Algoritam propagacije greške

Veštačke neuronske mreže mogu imati različite arhitekture, npr. arhitekturu asocijativne memorije, mreže s propagacijom greške (*backpropagation*, *feedforward*) i rekurentne mreže, Sl. 72 [1]. Postoje i brojna specifična rešenja modela neuronskih mreža, kao što su npr. radijalne funkcije (*Radial Basis Functions*, RBF), probabilističke neuronske mreže, Hopfildove i samoorganizujuće mreže.



Sl. 72. Tipične arhitekture veštačkih neuronskih mreža

Primena veštačkih neuronskih mreža

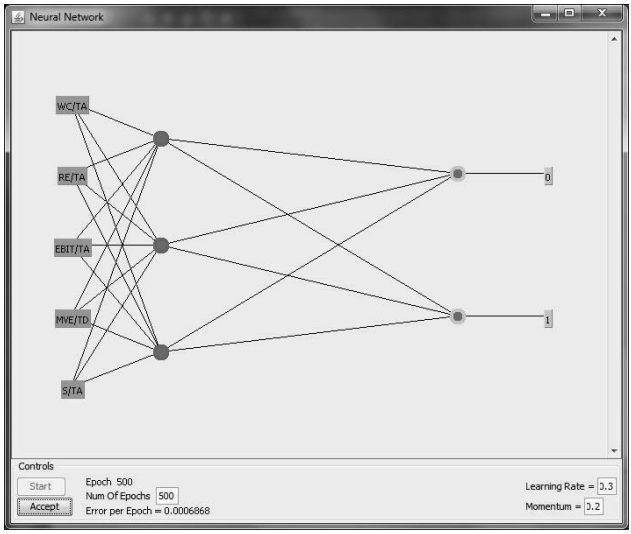
Razvoj veštačkih neuronskih mreža je složen i računski veoma zahtevan postupak, dok je njihova primena veoma efikasna. Obično se prvo izabere arhitektura i definiše konkretna struktura mreže. Obučavanje se vrši dok se ne dobiju zadovoljavajuće performanse na testnom skupu ili dok ponavljanje postupka više ne daje poboljšanja.

Obučavanje mreže koja ima fiksnu, unapred zadanu strukturu svodi se na određivanje težina veza. Kompletan proces razvoja modela veštačke neuronske mreže za rešavanje nekog problema može se ukratko opisati kao:

- prikupljanje podataka i formiranje obučavajućeg skupa;
- izbor pogodne strukture mreže, npr. s jednim skrivenim slojem;
- izbor algoritma učenja, npr. *backprop*;
- *obučavanje*, koje se vrši u iteracijama, s tim da se u svakoj iteraciji vrši testiranje modela na testnom skupu radi merenja performansi;
- implementacija mreže, koja se može izvršiti hardverski ili softverski, korišćenjem nekog programskog jezika ili namenskog softvera.

Primer: Učenje pravila klasifikacije pomoću veštačke neuronske mreže

Predviđanje stečaja kompanije je problem opisan u tački 8.2. Učenje pravila klasifikacije pomoću veštačke neuronske mreže može se realizovati u okviru programa Weka (metod *Multilayer Perceptron*), koji poseduje i grafički editor strukture mreže:



Rezultati učenja vide se iz izveštaja:

```
Scheme:       weka.classifiers.functions.MultilayerPerceptron -L 0.3 -
M 0.2 -N 500 -V 0 -S 0 -E 20 -H a
Relation:     Bankruptcy
Instances:    20
Attributes:   6
              WC/TA
              RE/TA
              EBIT/TA
              MVE/TD
              S/TA
              BR/NB
Test mode:    10-fold cross-validation
...
Correctly Classified Instances          20               100      %
Incorrectly Classified Instances         0                 0      %
```

Vidi se da procenjena tačnost naučene veštačke neuronske mreže metodom unakrsne validacije iznosi 100%. Konfiguracija naučene neuronske mreže daje strukturu i vrednosti elemenata mreže, težina pojedinih ulaza i pragova aktivacione sigmoid funkcije:

```
Sigmoid Node 0
    Inputs    Weights
    Threshold    0.2371217113969034
    Node 2    -1.1628334833591436
    Node 3    5.604749511353181
    Node 4    -4.881596560417678
Sigmoid Node 1
    Inputs    Weights
    Threshold    -0.22668980982630868
    Node 2    1.1444939668327085
    Node 3    -5.608403146810656
    Node 4    4.882711122017258
Sigmoid Node 2
    Inputs    Weights
    Threshold    -0.1724610782503284
    Attrib WC/TA    -0.1794107913471712
    Attrib RE/TA    1.3806454084725033
    Attrib EBIT/TA    0.9229764807764474
    Attrib MVE/TD    0.379066064551664
    Attrib S/TA    -0.16378542159685397
Sigmoid Node 3
    Inputs    Weights
    Threshold    0.8174695707991583
    Attrib WC/TA    0.7099360820546824
    Attrib RE/TA    -5.514066749665638
    Attrib EBIT/TA    -3.6571182734280567
    Attrib MVE/TD    -1.4494847826617454
    Attrib S/TA    0.8912329990963338
Sigmoid Node 4
     Inputs    Weights
    Threshold    -0.7487772170032302
    Attrib WC/TA    -0.6335761781507784
    Attrib RE/TA    5.030403436548837
    Attrib EBIT/TA    3.35334973876284
    Attrib MVE/TD    1.3058638713501698
    Attrib S/TA    -0.7975513809774544
```

```
Class 0
    Input
    Node 0
Class 1
    Input
    Node 1
```

Primeri primene VNM u podršci odlučivanju

Prikazaće se dva primera upotrebe metoda mašinskog učenja VNM u podršci odlučivanju u praktičnim nestrukturiranim problemima [1]. Prvi primer je upotreba veštačke neuronske mreže u praćenju kvaliteta i ukusa piva u poznatoj pivari.

Primer: Praćenje kvaliteta i ukusa piva

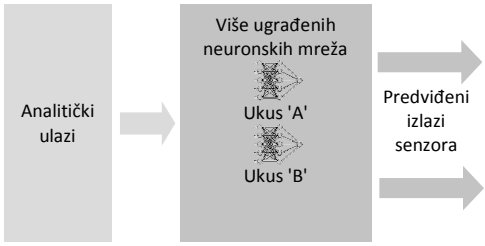
Poznata britanska pivara *Coors Brewers Ltd*. želela je da unapredi proces proizvodnje i obezbedi konstantni kvalitet proizvedenog piva [1].

Presudan faktor proizvodnje piva je njegov ukus, koji zavisi od hemijskog sastava. Veza rezultata hemijskih analiza i ukusa piva nije u potpunosti poznata. Postoji veliki broj analitičkih podataka (hemijske analize) i podataka senzora (test panel) prikupljanih godinama.

Kompanija je odlučila da podatke upotrebi za kreiranje sistema koji bi automatizovao proces upravljanja kvalitetom proizvedenog piva. Obučena je veštačka neuronska mreža (VNM) za *predviđanje ukusa* piva na osnovu njegovog hemijskog sastava.

Analitički podaci: na ulazu	Podaci senzora: na izlazu
alkohol	alkohol
boja	kiselost
slad	slad
etil-acetat	zrnatost
izobutil-acetat	izgorelost
etil butirat	aroma hmelja
izoamil-acetat	karamel
etil-heksanoat	slatkoća

Obučavanje neuronske mreže realizovano je u više faza, korišćenjem različitih arhitektura i metoda obučavanja, dok se u praksi nisu postigle zadovoljavajuće performanse.



Drugi primer je istovremena upotreba više metoda mašinskog učenja za kreiranje sistema koji otkriva i sprečava prevare u mobilnim telekomunikacijama.

Primer: Prevare u telekomunikacijama

Najveća južnoafrička telekomunikaciona kompanija imala je godišnje gubitke zbog zloupotreba veće od 37 milliona dolara [1]. Najveći gubici su nastajali zbog zlopupotreba pretplate, npr. neplaćanjem *postpaid* računa, kao i zbog manipulisanja telefonskim karticama, falsifikovanja i kloniranja.

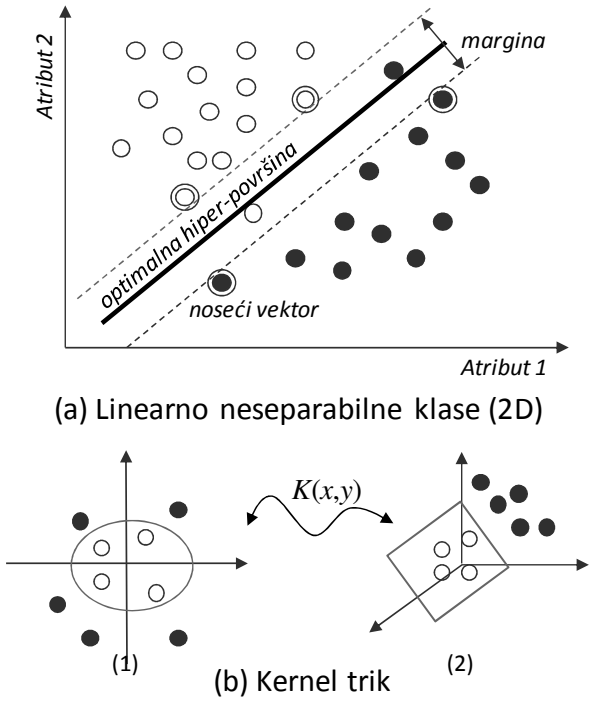
Razvijen je sistem za prevenciju zloupotreba *Minotaur*, u čiju realizaciju su uključeni metodi učenja na osnovu primera, tehnike neuronskih mreža, pravila i zaključivanja na osnovu slučajeva (CBR). Za učenje su korišćeni podaci o pozivima, mreži i informacije o pretplatnicima.

U prva tri meseca upotrebe sistema za prevenciju zloupotreba ostvareni su značajna poslovni rezultati i poboljšana efikasnost otkrivanja zloupotreba:

- prosečni gubici zbog zloupotreba po slučaju smanjeni su za 40%,
- prosečno vreme otkrivanja zloupotreba smanjeno je za 83%,
- prosečno vreme analize sumnjivih slučajeva smanjeno je za 75%,
- prosečna stopa otkrivanja poboljšana je za 74%.

9.5 Metod nosećih vektora (SVM)

Metod nosećih vektora (*Support Vector Machines*, SVM) metod je mašinskog učenja koji pronalazi *optimalnu hiperpovršinu*, koja razdvaja obučavajuće primere s maksimalnom marginom i služi za klasifikaciju novih primera, Sl. 73 (a). Noseći vektor je naziv za *granične primere*, koji su najbliži hiperpovršini maksimalne margine [58], [59].



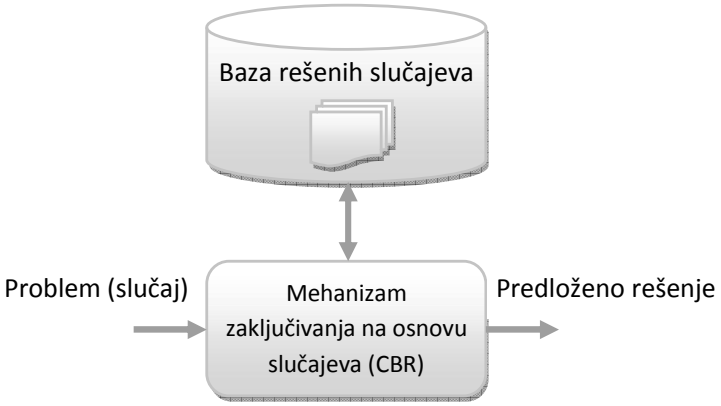
Sl. 73. Princip rada metoda nosećih vektora

Primere nije uvek moguće linearno razdvojiti u originalnom prostoru atributa, pa se koristi tehnika preslikavanja u prostor s većim brojem dimenzija, tzv. *kernel trik*, Sl. 73 (b). Veći broj dimenzija pomaže da se nelinearni problem (1) transformiše u linearni (2) u prostoru u kome se optimalna hiperpovršina lakše konstruiše. Nelinearna hiperpovršina dobija se inverznim preslikavanjem u originalni prostor. Simetrična funkcija preslikavanja *K(x,y)* naziva se *kernel funkcija*.

Metod je veoma tačan i računski efikasan, tako da se koristi kao alternativa za veštačke neuronske mreže, čije je obučavanje računski zahtevno [60].

9.6 Zaključivanje po analogiji (CBR)

Zaključivanje direktno na osnovu svih poznatih primera (*Case Based Reasoning*, CBR) forma je zaključivanja po *analogiji*. Metod vrši implicitnu generalizaciju, tako što ustanovljava sličnost primera s primerima ranije donesenih odluka i predlaže najsličniju odluku kao rešenje, Sl. 74 [46].



Sl. 74. Primena metoda zaključivanja na osnovu slučajeva

Primeri rešenih problema odlučivanja zadani su vrednostima *n* atributa i klasom primera, koja predstavlja konkretnu odluku. Načelno se pamte svi prethodno rešeni, odnosno uspešno klasifikovani primeri ili slučajevi.

Sličnost dva primera odluka **x** i **y** definiše se kao inverzno rastojanje ili distanca $d(\mathbf{x},\mathbf{y})$ dva vektora u *n*-dimenzionalnom prostoru, definisanom skupom atributa, kao na Sl. 75.

Neke od najčešćih mera rastojanja primera u višedimenzionalnom prostoru su:

- Euklidska distanca (*Euclid*)

$$d(\mathbf{x},\mathbf{y}) = \sqrt{\sum_{i=1}^{d}(x_i - y_i)^2}$$

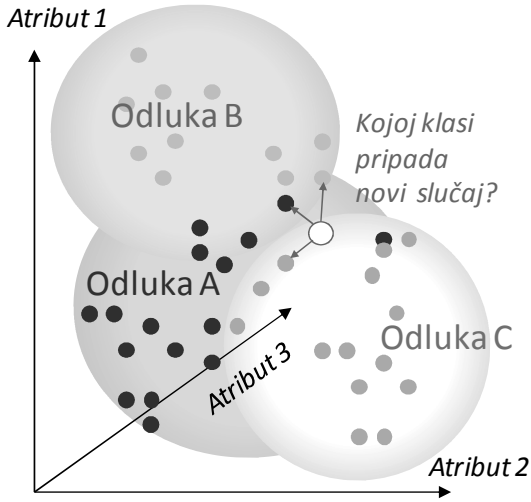
- Menhetn (*Manhattan*, *city-block*) distanca

$$d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{d} |x_i - y_i|$$

- Hamingova (*Hamming*) distanca

$$d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{d} \begin{cases} 0 & x_i = y_i \\ 1 & x_i \neq y_i \end{cases}$$

Odlučivanje se sastoji u klasifikaciji novog slučaja ili primera u jednu od unapred definisanih klasa ili odluka, koja se vrši na osnovu vrednosti funkcije sličnosti sa najbližim susedom. Da se izbegne preveliki uticaj izuzetaka, često se koristi grupna klasifikacija *k* najbližih suseda, $k>1$ [46].



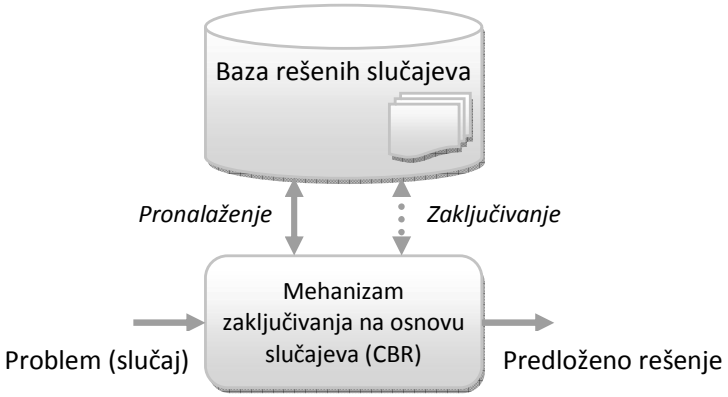
Sl. 75. Klasifikacije novih primera metoda zaključivanja na osnovu slučajeva

Postupak zaključivanja

Postupak odlučivanja može se ukratko opisati kao sekvenca:

1. Pronađi (*retrieve*) - pronađi najsličniji primer u bazi;
2. Ponovo upotrebi (*reuse*) - ponovo upotrebi informacije i znanje o tom primeru za rešenje problema;
3. Izmeni (*revise*) - izmeni predloženo rešenje za rešavanje novog problema;
4. Sačuvaj (*retain*) - zadrži delove novog rešenja, koji će verovatno biti korisni za buduće rešavanje problema.

Na Sl. 76 je primer računanja mere sličnosti dva primera pomoću Hamingove distance, koja se može primeniti na primere opisane nenumeričkim (nominalnim, kategoričkim) atributima.



novi slučaj x:	4	Kiša	Umereno	Visoka	Slab
slučaj iz baze y:	3	Oblačno	Toplo	Visoka	Slab

Hamingova distanca $d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{5} \begin{cases} 0 & x_i = y_i \\ 1 & x_i \neq y_i \end{cases} = 1+1+1+0+0 = 3$

minimalna distanca ⇔ najveća sličnost slučajeva

Sl. 76. Postupak zaključivanja na osnovu slučajeva

Alati za razvoj sistema u tehnologiji CBR

Razvojni alati za CBR tehnologiju su samostalni (*stand-alone*) sistemi, kao delovi okruženja mašinskog učenja i istraživanja podataka (*Data Mining*), odnosno sistema poslovne inteligencije (*Business Intelligence Suits*).

Primeri primene CBR u podršci odlučivanju

Primena metoda zaključivanja na osnovu slučajeva (CBR) u rešavanju značajnih prektičnih problema menadžmenta ilustrovaće se na dva primera iz literature [2]: unapređenju održavanja avionskih motora kompanije *Snecma* i upravljanja infrastrukturom grada Verdena (*Quebec*, *Canada*).

Primer: Unapređenje održavanja avionskih motora



Avionski motor je veoma složen tehnički sistem. Francuski proizvođač *Snecma* je četvrti proizvođač avionskih motora u svetu [1], čiji motori su ugrađeni u najpoznatije modele aviona, kao što su Boing 737 i Airbus 340.

Kompanija je tražila način da smanji troškove održavanja avionskih motora. Pošto na

dijagnostiku motora otpada 50% vremena održavanja, pokrenut je projekt CASIOPEE, čiji je cilj bio smanjenje cene održavanja avionskih motora pomoću naprednog softvera za dijagnostiku, koji koristi tehnologiju zaključivanja na osnovu slučajeva (CBR).

U bazu sistema je ugrađeno 16.000 slučajeva, opisanih pomoću 80 atributa (oko 40 po slučaju). Generisano je stablo odlučivanja za klasifikaciju slučajeva i upravljanje defektacijom.

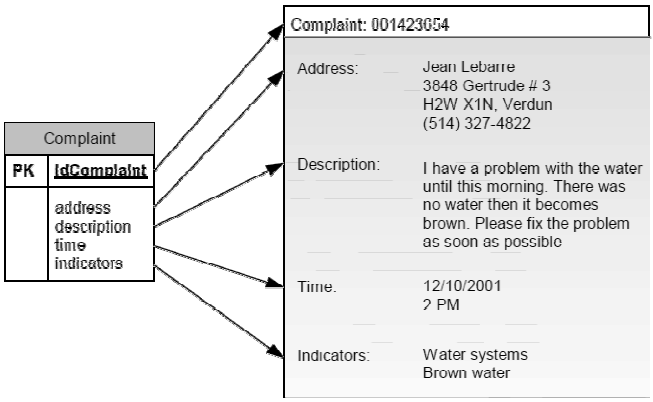
Efekti upotrebe sistema su:

- smanjenje vremena otklanjanja kvarova,
- minimizacija troškova dijagnostike,
- smanjenje broja pogrešnih dijagnoza i
- razvoj sistema održavanja i izrada dokumentacije o ekspertizi (*know-how*) od strane najobučenijih specijalista.

Drugi primer je problem upravljanja infrastrukturom grada kada su resursi za razvoj i održavanje infrastrukture ograničeni, a postoji veliki broj konkretnih problema, koje nije moguće precizno predvideti i koji često zahtevaju brzo i adekvatno reagovanje.

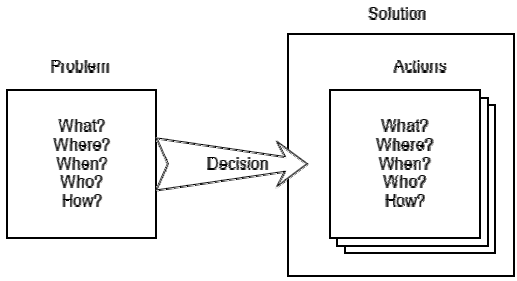
Primer: Unapređenje upravljanja infrastrukturom grada Verdena

Problem gradske uprave grada Verdena (*Verdun, Quebec*, *Canada*) bio je nedostatak resursa za sanaciju gradske infrastrukture [44]. Grad je raspolagao informacionim sistemom koji je prikupljao informacije o zahtevima i žalbama, kao i stanju infrastrukture, sa velikim brojem senzorskih ulaza, npr. o nivoima vode u rezervoarima gradskog vodovoda. Obrada zahteva i realizacija rešenja nisu bili dovoljno efikasni.



Rešenje problema je pronađeno u *inteligentnom sistemu za podršku odlučivanju (IDSS)* zasnovanom na modelima.

Sistem za pristup znanju o istorijatu problema, preduzetim akcijama i rešenjima koristi tehnologiju zaključivanja na osnovu slučajeva (CBR):



Efekti upotrebe novog sistema za podršku odlučivanju su poboljšana realizacija, bolje performanse i koordinacija gradskih službi, kao i brže rešavanje žalbi građana, uz manji utrošak resursa.

9.7 Programski alati mašinskog učenja i istraživanja podataka

Programski alati za mašinsko učenje i istraživanje podataka su brojni i mogu se pojaviti u različitim konfiguracijama: kao samostalni integrisani sistemi, samostalni programi mašinskog učenja, delovi komercijalnih sistema poslovne inteligencije (*Business Intelligence Suites*, BI) i specifični Web alati.

Samostalni integrisani alati

Poznati i popularni alati za istraživanje podataka otvorenog koda (*Open-source Data Mining Tools*) su [64]:

- Weka,
- KNIME,
- Rapid Miner.

Postoji veliki broj besplatnih alata različitih mogućnosti. Integrisani sistem za istraživanje podataka metodima induktivnog učenja autora ove knjige pod nazivom *Empiric* prikazan je u Prilogu E.

Poznatiji komercijalni alati za istraživanje podataka (*Data Mining Tools*) su npr. [64]:

- SAS Institute *Enterprise Miner*;
- IBM *Intelligent Miner* (deo sistema za upravljanje bazama podataka DB2);

- IBM *Cognos*;
- IBM *SPSS Modeler* (ranije poznat pod nazivom *Clementine*);
- Oracle, Oracle BI 11g;
- Microsoft SQL Server 2005/2008 *Business Intelligence Development Studio* (BIDS), odnosno najnoviji SQL Server 2012 *Data Modeler*;
- Angoss *KnowledgeSTUDIO*;
- *KXEN Analytic Framework*;
- Megaputer *PolyAnalyst*.

Samostalni programi mašinskog učenja

Postoje brojni besplatni i komercijalni sistemi mašinskog učenja, npr. za induktivno učenje stabala odlučivanja (CART [30], C45[31]), produkcionih pravila (CN2 [32], RIPPER [33]), neuronskih mreža (NeuroTools [65]) i metoda nosećih vektora (LIBSVM [34]).

Imaju različit interfejs, format ulaznih podataka i naučenog znanja, kao i brojne parametre za čije razumevanje je potrebno određeno predznanje.

Delovi komercijalnih sistema poslovne inteligencije

Komercijalni sistemi poslovne inteligencije za analize koriste različite metode mašinskog učenja stabala odlučivanja, pravila, veštačkih neuronskih mreža, nosećih vektora i sl.

Postoje sistemi poslovne inteligencije otvorenog koda, koji su uglavnom razvijeni na osnovu metoda mašinskog učenja, npr.

- Pentaho - sistem koji se zasniva na Java platformi i sistemu za istraživanje podataka otvorenog koda *Weka*;
- KNIME (*Konstanz Information Miner*) - sistem za integraciju podataka, analize i istraživanje podataka;
- Rapid Miner (ranije Yale) - sistem za istraživanje podataka i predviđanja (*Data Mining and Predictive Analytics*).

Komercijalni sistemi poslovne inteligencije se najčešće nude kao delovi ili dodaci sistemima za upravljanje bazama podataka, npr.

- Oracle BI 11g,
- IBM BI (DB2),
- Microsoft BI (SQL Server 2005/2008, SharePoint, Office).

Algoritmi mašinskog učenja u sistemu SQL Server 2008

Sistem za upravljanje bazama podataka Microsoft *SQL Server* sadrži više metoda mašinskog učenja, koji omogućavaju npr.

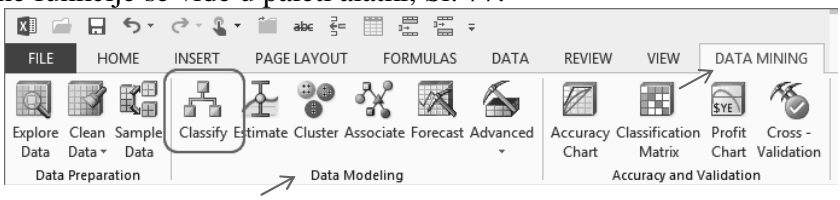
- učenje klasifikacija, recimo u analizi lojalnosti klijenata (*churn*) i analizi rizika (*risk analysis*);
- učenje pravila regresije, recimo za predviđanje profita ili prihoda;
- učenje asocijacija, recimo za analizu na osnovu više prediktivnih promenljivih.

Dodatak za istraživanje podataka Microsoft Office (Data Mining Add-In)

Podržava kompletan proces istraživanja podataka u sistemu Microsoft *Excel*, koji obuhvata:

- *Priprema podataka* - istraživanje, čišćenje i i priprema podataka za istraživanje znanja;
- *Modeliranje podataka* - izgradnja uzoraka i trendova na osnovu podataka radi predviđanja;
- *Tačnost i validacija* - testiranje i validacija modela;
- *Rad s modelima istraživanja podataka* - pregled, modifikacija i upravljanje postojećim modelima;
- Dokumentovanje - beleženje aktivnosti u obliku naredbi, Data Mining Extensions (DMX) ili Analysis Services Scripting Language (ASSL).

Osnovne funkcije se vide u paleti alatki, Sl. 77.



Sl. 77. Paleta alatki Microsoft Excel Data Mining dodatka

Sistem za istraživanje podataka Palisade DecisionTools Suite (Excel Add-In)

Palisade DecisionTools Suite je komercijalni integrisani skup programa za analizu rizika i podršku odlučivanju u uslovima neizvesnosti.

Softver se koristi kao dodatak za Microsoft *Excel*. Ima module mašinskog učenja *PrecisionTree* za stabla odlučivanja, *NeuralTools* za učenje/predviđanje pomoću neuronskih mreža (ANN), uz ostale elemente namenjene podršci odlučivanju: *StatTools* za statističke analize i predikciju (*forecasting*), *@RISK*

za Monte Karlo simulacije, *TopRank* za *what-if* analizu osetljivosti, te *Evolver* i *RISKOptimizer* za različite optimizacije.

Objektne programske biblioteke i jezici

Postoje brojne programske biblioteke i dodaci za razvoj i implementaciju sistema koji koriste metode mašinskog učenja. Primer veoma često korišćenih sistema su programske biblioteke vezane za programski jezik *R* i sistem *Matlab*:

- *R* - sistem i interpretativni programski jezik otvorenog koda za jednostavnu manipulacija podacima i sa dobrom grafičkom podrškom; veoma popularan statistički softver; standard analize i istraživanja podataka s velikim brojem biblioteka/paketa za mašinsko učenje i istraživanje podataka koje se mogu pronaći na Web sajtu CRAN (*Comprehensive R Archive Network*). Detaljnije je opisan u Prilogu F.
- *Python* - programski jezik opšte namene, namenjen produktivnom razvoju desktop, mrežnih i Web aplikacija. Ima relativno mali broj osnovnih naredbi. Raspolaže bogatom standardnom bibliotekom s interfejsom za veliki broj funkcija operativnog sistema. Omogućava objektno-orijentisani, imperativni i funkcionalni pristup programiranju, veliki broj tipova podataka i efikasno upravljanje dinamičkom memorijom. Postoji veći broj biblioteka metoda mašinskog učenja i istraživanja podataka razvijenih za jezik Python.
- *Matlab* - komercijalni sistem i interpretativni jezik za numeričke primene, manipulisanje matricama sa dobrom grafičkom podrškom i velikim brojem dodataka (*toolboxes*/*libraries*).

Web alati i mrežno računarstvo

Alati za mašinsko učenje na masivnim mrežnim serverima (*cloud-based machine learning tools*) koriste se u analizi podataka radi dodavanja novih mogućnosti različitim Web aplikacijama.

- Google *Prediction* API omogućava programerima pristup naprednim algoritmima mašinskog učenja kompanije Google, pomoću kojih mogu da analiziraju širok spektar empirijskih podataka velikog obima i daju predviđanja ishoda koji su od interesa.

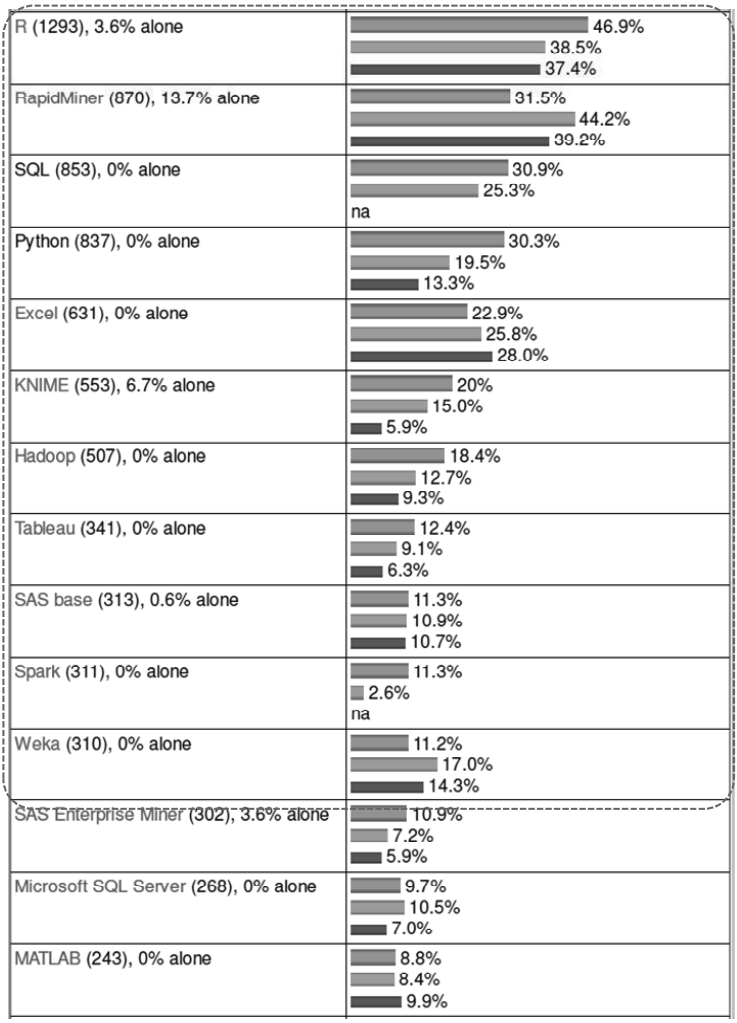
  Primeri primene su različiti inteligentni sistemi - automatizovani *spam* i bezbednosnu filteri, sistemi preporučivanja zasnovani na demografiji klijenata ili sistemi koji pronalaze skrivene obrasce u finansijskim podacima.
- Apache *Mahout* je programska biblioteka otvorenog koda, koja se u praksi najviše koristi za implementaciju sistema za preporučivanje (*recommender*

engines), ali poseduje metode učenja klasifikacija i grupisanja (*clustering*). Osnovna namena sistema je obrada velikih skupova podataka, prevelikih za obradu na jednom računarskom sistemu.

Sistemi za preporučivanje su najbrže prihvaćena primena mašinskog učenja u praksi. Danas veliki broj sajtova e-trgovine svoju ponudu zasniva na prepoznavanju ponašanja klijenata u prošlosti i pogađanju njegovog ukusa i interesovanja, koji se vrši na osnovu automatizovano naučenih pravila.

Istraživanje KDNuggets.com (2013-2015)

Poznati portal za stručnjake na području istraživanja podataka KDNuggets.com svake godine sprovodi ankete, između ostalog s pitanjem: *Koji alat za istraživanje znanja ste koristili u poslednjih 12 meseci u stvarnim projektima*? Rezultati ankete za 2015. i prethodne dve godine prikazani su na Sl. 78.



Sl. 78. Popularnost alata za istraživanje podataka prema KDNuggets.com

Posetioci sajta KDnuggets u 2015. godini najviše su koristili alate otvorenog koda, među kojima su vodeći bili *R*, *RapidMiner*, *Weka* i *KNIME*.

Od komercijalnih alata, kao vodeći su označeni *Excel*, *SAS*, Microsoft *SQL Server*, *Tableau*, *MATLAB*, IBM *SPSS Statistics*, SAS *Enterprise Miner* i IBM *SPSS Modeler*.

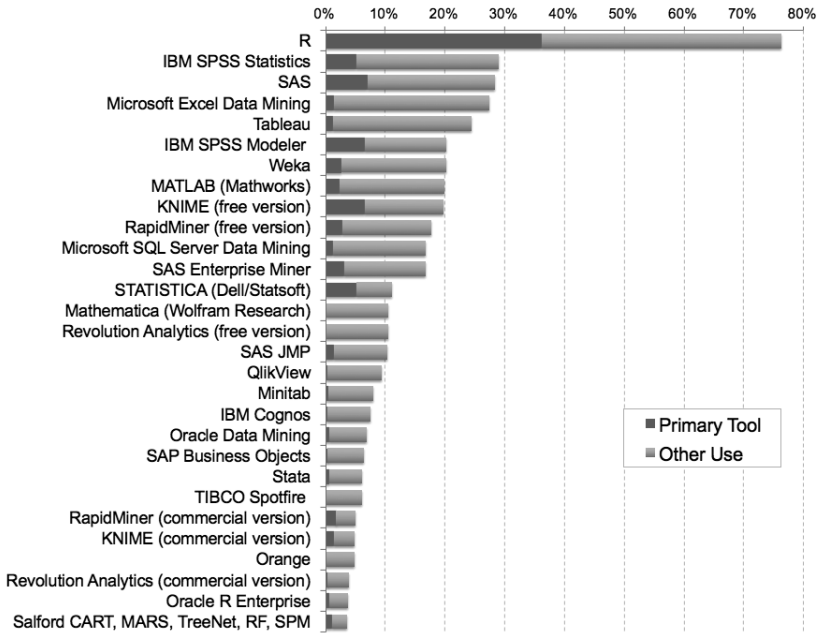
Analiza prakse istraživanja podataka Rexer Analytics (2015)

U istraživanju o praksi istraživanja podataka, ovaj put pod novim nazivom "nauka o podacima" (*Data Science*), anketirano je 1.220 profesionalnih istraživača/analitičara podataka iz 72 zemlje. Algoritmi koji su se najviše koristili su metodi učenja modela regresije, stabala odlučivanja (*decison trees*) i klasteringa (*cluster analysis*).

Softver otvorenog koda za istraživanje podataka *R* prevazišao je sve ostale alate i od 2010 je alat kog koristi najviše stručnjaka (76%), Sl. 79.

Ispitanici se se napohvalnije izrazili o sistemima KNIME i IBM SPSS Modeler, nakon čega po stepenu zadovoljstva korisnika slede R, STATISTICA, Weka, te ostali softverski alati za istraživanje podataka.

Važan zaključak je da i dalje jedan alat za istraživanje podataka nije dovoljan, pošto stručnjaci za rad koriste u proseku 5 različitih alata. Sistem *R* je alat koji najveći broj stručnjaka navodi kao osnovni (36%), Sl. 79.



Sl. 79. Popularnost alata za istraživanje podataka prema Rexer Analytics

Pitanja za ponavljanje

1. Šta je mašinsko učenje?
2. Koje su osnovne forme mašinskog učenja?
3. Koje strategije mašinskog učenja spadaju u induktivno mašinsko učenje?
4. Koja je osnovna razlika između zadatka klasifikacije i regresije u mašinskom učenju?
5. Kako se definiše razumljivo znanje?
6. Koji modeli predstavljanja znanja imaju najviši stepen razumljivosti?
7. Koji su osnovni metodi induktivnog učenja pravila?
8. Navedite osnovne metode estimacije tačnosti predviđanja naučenog znanja.
9. Koji je osnovni procesni element (PE) veštačke neuronske mreže (ANN)?
10. Koji interni algoritam je od posebnog značaja za proces obučavanja veštačke neuronske mreže (ANN)?
11. Koji metod mašinskog učenja je uporediv po tačnosti predviđanja s veštačkim neuronskim mrežama (ANN)?
12. Koji metod mašinskog učenja po analogiji poznajete?
13. Koji matematički koncept se koristi u za računanje sličnosti slučajeva (primera) u CBR (*case-based reasoning*) sistemima?
14. Navedite osnovne korake procedure zaključivanja po analogiji u CBR (*case-based reasoning*) sistemima.

10 Inteligentna podrška elektronskoj trgovini

1. Elektronska trgovina
2. Inteligentni sistemi u e-trgovini
3. Inteligentni interfejsi u e-trgovini
4. Primeri inteligentne podrške e-trgovini

Inteligencija je sposobnost učenja i razumevanja, rešavanja problema i donošenja odluka. Veštačka inteligencija je nauka čiji cilj je izrada sistema (mašina) koje mogu da izvršavaju zadatke koji bi zahtevali inteligenciju ako ih izvršavaju ljudi. Mašina (računarski program) se smatra inteligentnom ako može postići ljudski nivo performansi u nekim kognitivnim zadacima.

Mašinsko učenje je proces u kome sistem poboljšava svoje performanse na datom zadatku bez dodatnog programiranja.

Znanje je kontekstno zavisna, relevantna i delatna informacija (eksplicitna ili implicitna)

10.1 Elektronska trgovina

Elektronska trgovina (*e-commerce*) obuhvata komercijalne transakcije između organizacija i pojedinaca zasnovane na digitalnoj tehnologiji [17], [35]. Transakcije e-trgovine se odnose na razmenu vrednosti preko organizacionih ili individualnih granica u zamenu za proizvode i usluge.

Elektronsko poslovanje (*e-business*) obuhvata transakcije i procese u okviru jedne organizacije zasnovane na digitalnoj tehnologiji, uz pomoć i pod kontrolom njenog informacionog sistema. Elektronsko poslovanje ne obuhvata komercijalne transakcije izvan organizacionih granica kojima se vrši razmena vrednosti.

Osnovni oblici (tipovi) elektronske trgovine su, prema odnosu učesnika na tržištu, između kompanija i pojedinaca (*Business-to-Consumers*, B2C), između kompanija (*Business-to-Business*, B2B) i između pojedinaca (*Consumer-to-Consumer*, C2C). Prema tehnologiji, osim klasičnih serverskih rešenja, izdvajaju se još *Peer-to-Peer* (P2P) i mobilna (*M-commerce*) rešenja.

Razvoj Web sajta e-trgovine predstavlja izgradnju specifičnog softverskog proizvoda, transakcionog informacionog sistema u Web tehnologiji. Standardne funkcije Web sajta e-trgovine su [35]

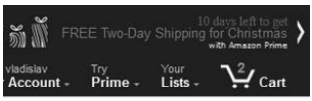
1. Online katalog proizvoda/usluga;

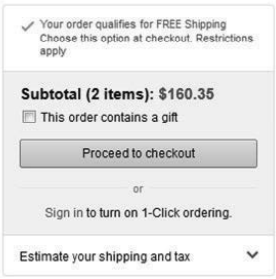
2. Baze podataka proizvoda i prodaje;
3. Korpa za kupovinu (*shopping cart*) i sistemi plaćanja;
4. Baza podataka korisnika (uključujući one koji su registrovani *online*);
5. Praćenje rada korisnika (*site log*, *pogodan za data mining*);
6. Server za reklame (*Ad server*) i e-poštu; upravljanje prikazom banera i marketinškim *e-mail* kampanjama;
7. Praćenje i analizu rada sajta (*site tracking and reporting*);
8. Upravljanje zalihama (*inventory management*).

Online katalog proizvoda/usluga

Sastoji se od liste proizvoda i usluga, detaljnih informacija o proizvodima i njihovim cenama. Može da sadrži multimedijske elemente, fotografije, animacije, zvuk, illustracije i video demonstracije. Neki sajtovi omogućavaju uspostavljanje direktne tekstualne (*chat*) i videokonferencijske komunikacije sa predstavnicima kompanije.

Korpa za kupovinu





Korpa za kupovinu (*shopping cart*) je popularan naziv za programski interfejs sajta e-trgovine, koji je posebno dizajniran za lakše pronalaženje i izbor proizvoda, odnosno udobno kreiranje narudžbe.

Interfejs je implementiran tako da *online* kupovanje što više podseća kupovinu ustvarnoj prodavnici. Izbor proizvoda vrši se jednostavnim aktiviranjem tastera, npr. *Add to Cart*, kao i kupovina, tasterom kao što je *Proceed to checkout*, koji pokreće funkciju *online* plaćanja. Softver korpe za kupovinu predstavlja deo serverskog softvera i smešten je na Web serveru.

Funkcije servera elektronske trgovine

Server e-trgovine (*e-Commerce Merchant Server*) je softver koji omogućava realizaciju osnovnih funkcija neophodnih za online prodaju.

Osnovne funkcije serverauključuju:

- pregled *online* kataloga proizvoda/usluga;

- preuzimanje narudžbine *online*, tako da kupci mogu da odvajaju na stranu proizvode koje žele da kupe, pregledaju šta su naručili i realizuju narudžbinu aktiviranjem tastera na Web stranici;
- obradu kreditnih kartica *online*; verifikacija kartice i bezbedan prenos navedenog iznosa na račun e-prodavca.

Softverski paketi elektronske trgovine

Softverski paketi e-trgovine (*e-Commerce Suites*) pružaju većinu ili sve funkcije i resurse potrebne za razvoj složenih Web sajtova. Prema mogućnostima i performansama, postoji osnovna, srednja i visoka klasa ovih softverskih rešenja.

Paket e-trgovine	***Kategorije Web sajtova e-trgovine***					
	U nastajanju	Mali	Srednji	Veliki	Veoma veliki	Najveći
IBM WebSphere	●	●	●	●	●	●
Oracle	•	•	•	●	●	•
Hybris	•	•	•	●	·	•
Intershop	•	•	•	•	•	·
Magento	●	●	●	●	•	
Drupal	•	•	•	•		
Prestashop	●	●	●	·		
Opencart	●	●	●	●	•	
Zencart	•	•	•	•		

Slika 80 – Relativna zastupljenost Web sajtova (po kategorijama) realizovanih poznatijim paketima e-trgovine

Na tržištu postoji veliki izbor besplatnih i komercijalnih paketa e-trgovine osnovne (niže) klase, kao što su:

- Web sajtovi hosting kompanija s interaktivnim generatorom sajtova e-trgovine, koji (besplatno) obezbeđuju najjednostavniji način kreiranja sajta s korpom za kupovinu, *online* plaćanjem i analitičkim funkcijama, dok se hosting obično naplaćuje u skladu s veličinom prodavnice. Primeri su *Yahoo Merchant Solutions* i *Webs.com*.
- Samostalni programski paketi korpi za kupovinu [8], kao što su *3DCart*, *Volusion* i *OSCommerce*, koji perdstavljaju veoma popularne alate za razvoj malih prodajnih sajtova, kod kojih se funkcije *online* plaćanja obično realizuju korišćenjem interfejsa servisa *PayPal*.

- Moduli sistema za upravljanje Web sadržajem (CMS), koji sadrže korpu za kupovinu, online plaćanje i analitičke alate. Primeri su Wordpress moduli *WP e-Commerce* i *WooCommerce*, odnosno *Drupal* modul *Commerce*.

Primeri alata iz srednje klase su *IBM WebSphere Commerce* (Express Edition) i *Sitecore Commerce Server* (ranije Microsoft) [17].

Softver za razvoj sajtova visokih performansi obuhvata veliki broj softverskih paketa različitih mogućnosti i cena, npr. *IBM WebSphere Commerce on Cloud* (Professional i Enterprise), *Oracle ATG Web Commerce*, *GSI Commerce*, *Demandware* i *Magento* [17].

Na Sl. 80 prikazana je relativna zastupljenost Web sajtova realizovanih pomoću najpoznatijih paketa e-trgovine među 10.000 najposećenijih [85].

Model ponašanja online potrošača

Ponašanje potrošača na mreži ima i neke specifičnosti u odnosu na tradicionalne potrošače. Opšti model ponašanja (demografski faktori, tržište, odluke potrošeča) treba proširiti novim faktorima:

- osobinama korisnika,
- svojstvima proizvoda i
- mogućnostima Web sajta, u koje spadaju sadržaj, dizajn i njegova funkcionalnost, koju određuju brzina odziva, navigacija i poverenje korisnika u bezbednost podataka.

Odluka o *online* kupovini je složen proces, koji se može podeliti u pet faza:
1. svest o potrebi,
2. traganje za više informacija,
3. procena alternativa,
4. konkretna odluka o kupovini,
5. kontakt sa firmom nakon kupovine.

Ponašanje potrošača u prodavnici u klasičnom modelu zamenjeno je ponašanjem potrošača prilikom *online* kupovine (*clickstream behaviour*). Dnevnik transakcija (*transaction log*) prati ponašanje potrošača od pretraživanja do kupovine.

Uloga marketinga

Marketing čine strategije i aktivnosti firme, koje se preduzimaju radi uspostavljanja odnosa sa potrošačima i podsticanja kupovine svojih proizvoda ili usluga. Marketing se koristi i za sprečavanje da proizvod postane roba. Roba su takva dobra ili usluge za koje postoji veliki broj prodavaca i svi proizvodi suštinski su identični, npr. pšenica ili čelik. Izbor kupca se zasniva isključivo na ceni i uslovima isporuke. Marketing nastoji da izbegne isključivo konkurenciju

cena i da kreira tržište gde je povrat investicija prosečan, konkurencija ograničena i gde će kupci plaćati više cene onih proizvoda koje doživljavaju kao jedinstvene.

Tehnologija Internet marketinga

Internet marketing se zasniva na upotrebi jedne ili više softverskih tehnologija:
1. dnevnici Web transakcija,
2. kolačići i Web bubice,
3. baze, skladišta podataka i istraživanje podataka,
4. mreže za reklamiranje,
5. sistemi za upravljanje odnosima s kupcima.

Dnevnici Web transakcija

Dnevnici Web transakcija (*Web Transaction Logs*) ugrađeni su u softver Web servera, koji snima korisničke aktivnosti na nekom Web sajtu. Obezbeđuju dragocene informacije za potrebe marketinga, posebno lične podatke koji se prkupljaju formama za registraciju i podatke o izabranoj robi, kupovini i plaćanju, koji se prikupljaju u bazi podataka korpe za kupovinu. Omogućava da se ustanove glavni obrasci interesovanja kupaca i načini kupovine, npr. na koju stranicu korisnici prvo prelaze nakon osnovne stranice sajta, gde se premeštaju nakon toga i slično.

Kolačići i Web bubice

Kolačići (*cookies*) su mali tekst fajlovi, koje Web sajtovi postavljaju na klijentski računar posetioca svaki puta kada poseti Web sajt i pristupi određenim stranicama. Omogućavaju stručnjacima Web marketinga vrlo brz način za identifikaciju i praćenje ponašanja kupaca u prethodnom periodu. Lokacija ovih fajlova na računaru zavisi od operativnog sistema i Web čitača.

Web bubice (*Web bugs*) su mali grafički fajlovi, veličine obično od samo jednog piksela. Upotrebljavaju se za automatsko slanje informacija o korisniku i tekućoj Web stranici na server, radi nadgledanja njegovih aktivnosti. Ugrađuju se u Web stranice i *e-mail* poruke na poseban način, korišćenjem posebne HTML naredbe koja označava hipervezu ka izabranom serveru.

Illustracija: Korišćenje Web bubica za praćenje Internet korisnika

Primer dela sadržaja zahtevane Web stranice [17]:

```
<img src="http://ad.doubleclick.net/ad/pixel.quicken/NEW" width=1 height=1 border=0>

<img width=1 height=1 border=0 SRC="http://media.preferences.com/ping?ML_SD=
IntuitTE_Intuit_1x1_RunOfSite_Any &db_afcr=4B31-C2FB-10E2C&event=reghome&
group=register& time=1999.10.27.20.5 6.37">
```

Informacije koje prikuplja Web bubica su:

- IP adresa računara koji je tražio Web stranicu,
- URL adresa Web stranice sa Web bubicom,
- URL adresa slike Web bubice,
- Datum i vreme pregleda Web bubice,
- Tip Web čitača koji je preuzeo stranicu sa Web bubicom,
- Prethodnu vrednost kolačića (*cookie*).

Različiti tipovi Web bagova uključuju obične rasterske slike u formatu GIF, izvršne bagove i bagove koji se zasnivaju na skriptu.

Baze, skladišta podataka i istraživanje podataka

Relaciona baza podataka predstavlja podatke u obliku skupa dvodimenzionalnih tabela sa zapisima organizovanim u redove i atributima entiteta organizovanim u kolone. Podaci različitih tabela se mogu međusobno povezivati, ako dele zajednički element podataka, koji se naziva primarni ključ (ili deo primarnog ključa). Jezik SQL (*Structured Query Language*) je standardni jezik za postavljanje upita i manipulisanje bazom podataka, koji se koristi u relacionim bazama podataka.

Skladišta podataka čine podaci izdvojeni za potrebe poslovnih analiza. Istraživanje podataka za potrebe marketinga bavi se otkrivanjem novih obrazaca u velikim skupovima podataka metodima veštačke inteligencije, mašinskog učenja, statistike i baza podataka.

Mreže za reklamiranje

Mreže za reklamiranje (*Advertising Networks*) mogu da prikazuju korisnicima reklamne banere, na osnovu sadržaja baza podataka o ponašanju korisnika. Server za reklamiranje (*Ad server*) bira odgovarajuće banere na osnovu kolačića, Web bagova i profila iz baze podataka korisnika.

Najpoznatiji primer mreže za reklamiranje je *DoubleClick*. Na slici je prikazan način funkcionisanja mreže za reklamiranje kao što je *DoubleClick*.

Sistemi za upravljanje odnosima s kupcima

Sistemi za upravljanje odnosima s kupcima (*Customer Relationship Management Systems*, CRM) su skladišta informacija o klijentima, gde se beleže svi kontakti korisnika sa kompanijom i stvara profil klijenta, koji je dostupan svima koji treba da poznaju kupca (*know the customer*). Korisnički profili mogu da sadrže:

- kartu odnosa klijenta sa firmom,
- rezime podataka o proizvodima i njihovoj upotrebi,
- demografske i psihološke (psihografske) podatke,

- mere profitabilnosti,
- istorijat kontakata,
- informacije o marketingu i prodaji.

10.2 Inteligentni sistemi u e-trgovini

Inteligentni sistemi se intenzivno koriste u e-prodaji i e-marketingu kao elementi praćenja i analize ponašanja klijenata, predviđanja njihovih akcija, poboljšanja komunikacije i automatizaciju procesa trgovanja.

Inteligentni sistemi u e-prodaji

Najpoznatije primene inteligentnih sistema u e-prodaji su sistemi za preporučivanje proizvoda i sistemi za automatizaciju aukcija.

Preporučivanje proizvoda

Sistemi za preporučivanje proizvoda (*recommender systems*) [42] su programski alati i tehnike koje daju sugestije kupcima prilikom pregleda kataloga, odnosno online kupovine. Koriste se za preporučivanje proizvoda u sistemima elektronske trgovine, ali i kao podrška u izboru, npr. restorana, saradnika, vrste finansijskih usluga, usluga osiguranja i stranica društvenih mreža.

Osnovni cilj sistema za preporučivanje u e-trgovini je podrška donošenju odluka u procesu kupovine proizvoda, kao što je izbor konkretnog proizvoda/usluge, odnosno načina plaćanja. Za razliku od klasične maloprodaje, koja je ograničena na ponudu manjeg broja najpopularnijih artikala, koji se prodaju u velikom broju primeraka, sajtovi e-trgovine mogu da ponude izuzetno veliki broj proizvoda i ostvaruju značajan profit od prodaje relativno malog broja primeraka proizvoda koje kupci teško pronalaze na neki drugi način, tzv. efekt dugog repa (*long tail*). Zbog toga ovi sistemi predstavljaju važno sredstvo elektronske trgovine, koji omogućava *online* korisniku (kupcu) da se izbori sa prevelikim obimom informacija i znatno povećavaju *online* prodaju.

Osnovne tehnike preporučivanja proizvoda su kolaborativno filtriranje, filtriranje po sadržaju (*content-based*), metodi istraživanja podataka (*data mining*) i kontekstualni metodi (*context-aware*) [42].

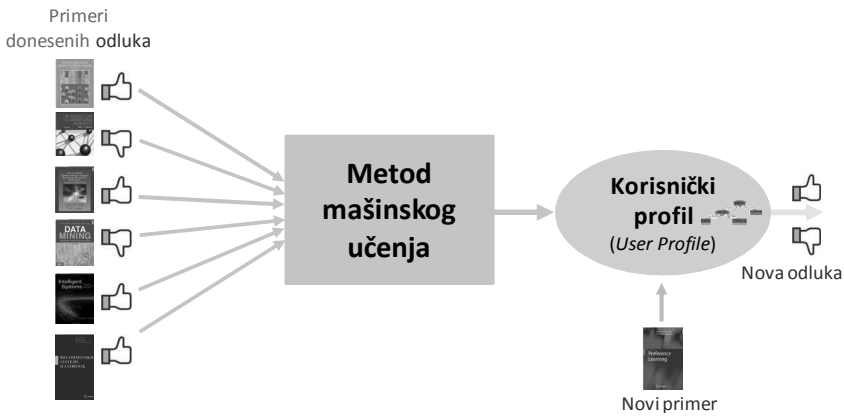
Primena i evaluacija sistema za preporučivanje zasniva se na oceni kvaliteta preporuka, koje zavise od izabranog pristupa i adekvatnosti primenjenih algoritama.

Proces preporučivanja metodima mašinskog učenja se sastoji iz dve faze, Sl. 80:

1. Učenje preferencija korisnika, odnosno pravila odlučivanja na osnovu primera donesenih odluka određenog korisnika, koja se ažuriraju u korisničkom profilu.

2. Primena pravila odlučivanja iz korisničkog profila za predviđanje budućih odluka određenog korisnika.

Kada prijavljeni ili prepoznati korisnik pristupi katalogu proizvoda ili usluga, sistem koristi model iz njegovog profila kao mehanizam klasifikacije ili ocene proizvoda. Sistem koristi tako dobijeno mišljenje za kreiranje personalizovanog spiska proizvoda u katalogu koji se korisniku prikazuje. Na taj način se korisniku olakšava izbor proizvoda ili usluge i povećava verovatnoća kupovine, pošto se nude proizvodi koji klijenta verovatno zanimaju.



Sl. 80. Princip induktivnog učenja i upotreba profila korisnika

Primer: Induktivno učenje preferencija korisnika

U primeru metod iduktivnog učenja stabala odlučivanja iz sistema Weka koristi za učenje modela odlučivanja, koji se nakon toga može koristiti za predviđanje budućih odluka, odnosno preferencija istog korisnika u odnosu na sve proizvode iz asortimana.

Skup podataka koji se koristi u ovom primeru je u formatu Weka, izgrađen na osnovu baze podataka proizvoda i korisnika:

```
@relation  RedWhiteWine-weka.filters....
@attribute 'user' {user1, user2, user3, user4, user5}
@attribute 'fixed acidity' numeric
. . .
@attribute alcohol numeric
@attribute R/W {R,W}
@attribute quality {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}

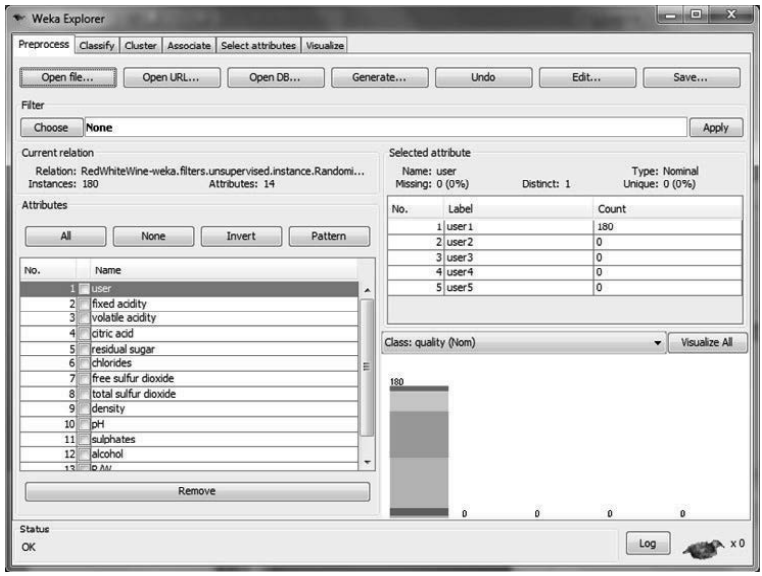
@data
user1,7.4,0.29,0.5,1.8,0.042,35,127,0.9937,3.45,0.5,10.2,W,7
user1,10,0.41,0.45,6.2,0.071,6,14,0.99702,3.21,0.49,11.8,R,7
. . .
```

Podaci sadrže ocene različitih vrsta vina koje su dala četiri različita korisnika. Svojstva ocenjenih vina opisana su pomoću 13 tributa, a same ocene predstavljaju klasifikacioni atributa *quality*.

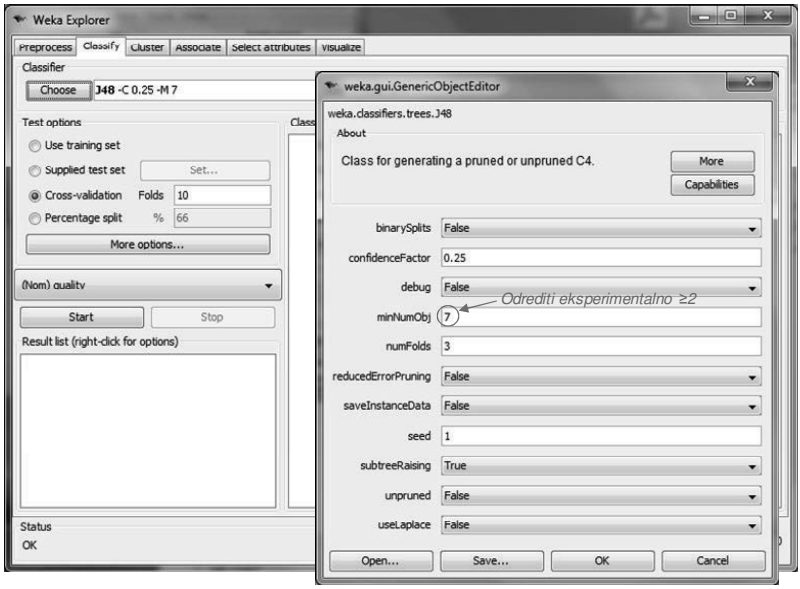
Procedura učenja pravila klasifikacije obuhvata sledeće korake:
1. Pokrenuti sistem WEKA i učitati skup primera *User1Opinion.arff*;
2. Izabrati metod učenja i postaviti parametre učenja;
3. Naučiti profil korisnika kao model klasifikacije, npr. skup *pravila*;
4. Sačuvati profil korisnika, odnosno model, npr. *pravila*;
5. Primeniti model na tekućeg korisnika i proizvode iz tekuće ponude;

Procedura se može automatizovati npr. u jeziku Java radi korišćenja u okviru softvera Web sajta.

(1) Pokretanje sistema WEKA i učitavanje skupa primera:



(2) Izbor metoda učenja i skupa parametara:



(3) Učenje pravila u obliku stabla odlučivanja, koja su razumljiva i lako se koriste.

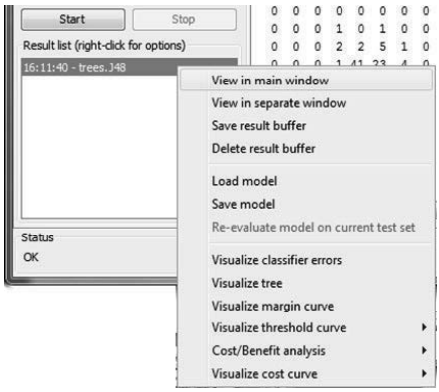
Naučeno stablo odlučivanja se vidi u tekstualnom obliku:

```
J48 pruned tree
------------------
alcohol <= 10.9
|   citric acid <= 0.49
|   |   total sulfur dioxide <= 152
|   |   |   R/W = R
|   |   |   |   sulphates <= 0.55: 5 (8.0/2.0)
|   |   |   |   sulphates > 0.55
|   |   |   |   |   residual sugar <= 2: 6 (7.0/1.0)
|   |   |   |   |   residual sugar > 2: 5 (9.0/2.0)
|   |   |   R/W = W: 6 (48.0/24.0)
|   |   total sulfur dioxide > 152
|   |   |   residual sugar <= 6.9: 6 (6.0/3.0)
|   |   |   residual sugar > 6.9: 5 (24.0/8.0)
|   citric acid > 0.49: 5 (13.0/3.0)
alcohol > 10.9
|   alcohol <= 12.1
|   |   alcohol <= 11.1: 5 (10.0/6.0)
|   |   alcohol > 11.1
|   |   |   volatile acidity <= 0.36
|   |   |   |   chlorides <= 0.043
|   |   |   |   |   fixed acidity <= 7.2: 6 (6.0/2.0)
|   |   |   |   |   fixed acidity > 7.2: 5 (8.0/5.0)
|   |   |   |   chlorides > 0.043: 6 (10.0)
|   |   |   volatile acidity > 0.36: 7 (8.0/2.0)
|   alcohol > 12.1
```

```
|   |   free sulfur dioxide <= 31: 7 (17.0/7.0)
|   |   free sulfur dioxide > 31: 8 (6.0/2.0)

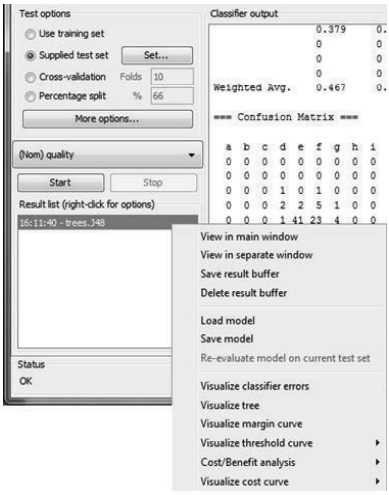
Number of Leaves  :         14
Size of the tree :  27
```

(4) Pamćenje modela (stabla odlučivanja ili pravila) vrši se desnim klikom na *Results list*, pa zatim na model koji se želi sačuvati. Izabere se opcija *Save model* i model se sačuva na fajl sa nastavkom **.model**.



(5) Primena pravila na tekućeg korisnika.

Skup primera za testiranje (testni skup) se može programski pripremiti u formatu Weka ili se može upotrebiti neki već postojeći testni skup. Za vežbu, u sistem Weka se može učitati testni fajl .arff s podacima o proizvodima za koje želimo mišljenje. Izabere se opcija testiranja *Supplied test set*: desni klik na *Results list*, izbor *Load model* i **.model** fajla, a zatim se na modelu pokrene *Re-evaluate model* za taj skup primera:



Rezultat testiranja su klasifikacija testnih primera i statistika testiranja.

Interakcija s korisnikom obezbeđuje prikaz, pregled, objašnjenja i vizualizaciju preporuka, kao i tehnike strukturiranja dijaloga u procesu preporučivanja. Sistemi grupnog preporučivanja omogućavaju pretvaranje različitih izvora i oblika preporuka u nove vrste pouzdanijih preporuka.

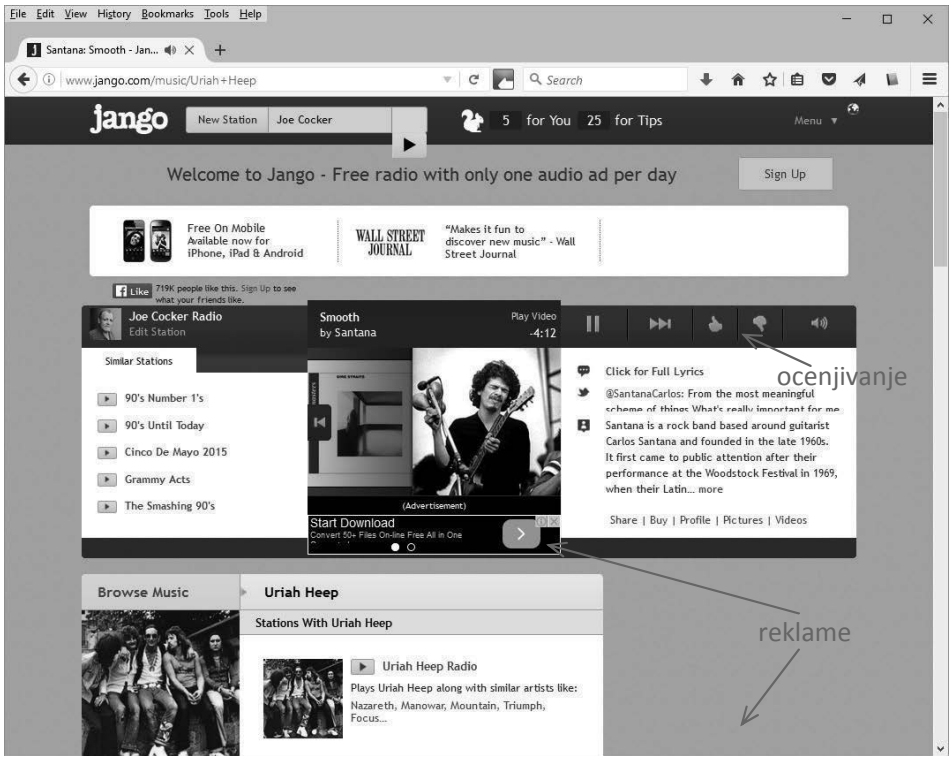
Illustracija 2: Jango - Internet muzički radio i društvena mreža

Sistem *Jango* [66] koncipiran je kao društveni Internet radio (*social Internet radio*), slično sajtovima *Pandora* i *Last.fm*. Omogućava korisniku kreiranje personalizovanih radio programa i pronalazi nove izvođače, koji se dopadaju slušaocima sličnog muzičkog ukusa.

Društvena mreža slušalaca omogućava dodavanje prijatelja (*friends*) i slušanje njihovih programa, pregled sugestija za izbor izvođača, ocenjivanje kompozicija pojedinih izvođača, otkrivanje slušalaca sličnih opredeljenja, posebno onih sa sličnim muzičkim ukusom.

Upotreba je potpuno besplatna. Nije potrebno preuzimanje softvera, a emitovanje muzike je licencirano kod organizacija ASCAP, BMI, SESAC i SoundExchange.

Sajt ostvaruje prihode preko partnerske prodaje muzike na osnovu ugovora sa sajtovima *Amazon* i *iTunes*, a koristi i *online* reklamiranje (*Google*).



U prikazanom primeru samo je unesen osnovni kriterijum (izvođač *Joe Cocker*), a sistem koristi zaključivanje po analogiji da samostalno nudi repetoar po sličnosti sadržaja, koji dalje preporučuje i emituje. Koristeći pregledne komande, klijent može da preskoči ponuđenu kompoziciju ili je samo loše oceni, kako bi modifikovao kriterijum za izbor narednih.

Primeri sistema sa preporučivanjem

Svaki ozbiljniji Web sajt e-trgovine koristi neki vid preporučivanja proizvoda. Primer preporučivanja knjiga je sajt amazon.com, muzičkih sadržaja music.yahoo.com, restorana entree.com, a za filmove netflix.com, imdb.com, alkindi.com, Movielens.com.

Sistemi za automatizaciju aukcija

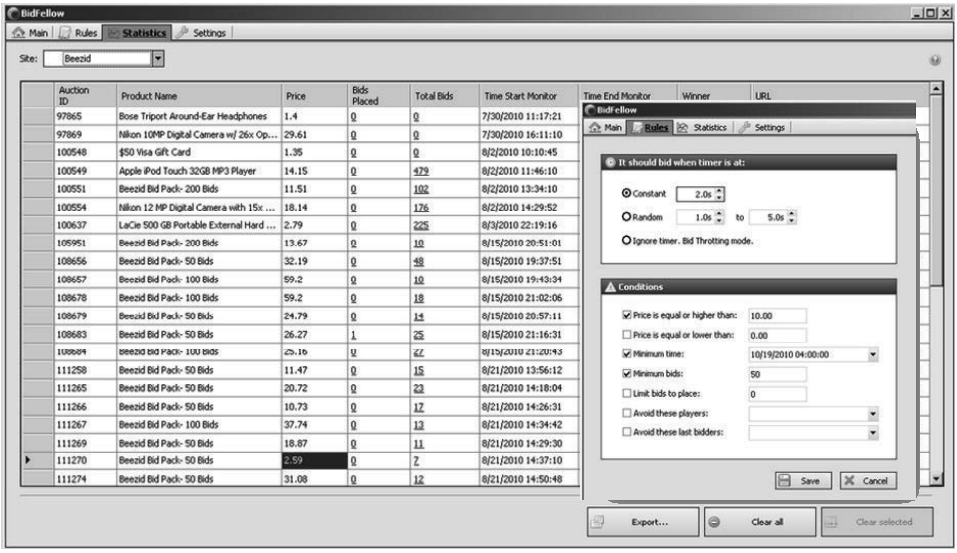
Aukcija je institucija tržišta sa jasnim skupom pravila koja određuju alokaciju i cene resursa na osnovu ponuda učesnika na tržištu. Predstavlja jedan od najstarijih oblika trgovanja, koji je bio poznat još u antička vremena, oko 500 godina pre naše ere. Danas se uz pomoć aukcija prodaju različite robe, npr. radio-frekventni spektar (FCC), nafta (OPEC) i mrežni resursi se prodaju posredstvom automatizovanih aukcija, za koje se koriste softverski agenti.

Za modeliranje aukcija pogodni su metodi teorije igara, koji razmatraju učešće konačnog broj učesnika sa suprotstavljenim interesima. Primeri prvih sistema za automatizaciju aukcija su *Internet AuctionBot* (University of Michigan, 1998), *eMediator* (Washington University, 2000) i *Secure Electronic Auction Protocol* (Ohio State University, 1996).

Automatizovane Internet aukcije obavljaju se korišćenjem softverskih agenata, kao što su npr. *BidPal*, *Auction Sniper*, *EZSniper*, *Gixen*, *BidFellow* i dr. U modeliranju aukcija, koje podsećaju na kocku, koristi se teorija igara.

Illustracija: BidFellow - sistem za automatizaciju Internet aukcija

Peni aukcije (*penny auctions*) su aukcije kod kojih se plaća davanje svake ponude, kojom se podiže cena za mali iznos. Aukcija se završava kad neko kratko vreme (10-20s) nema novih ponuda.



Pošto se može uložiti dosta novca i izgubiti, model podseća kocku ili igru, pa je potreban softver za optimizaciju strategije.

Inteligentni sistemi u e-marketingu

Najpoznatije primene inteligentnih sistema u e-marketingu su za personalizaciju, izradu profila metodima istraživanja podataka, upravljanje odnosima s kupcima i automatizaciju marketinga [17], [35].

Personalizacija

Personalizacije je prilagođavanja potrebama, interesovanjima i preferencijama korisnika. Može se zasnivati na automatizovanom (mašinskom) učenju

korisničkih profila. Personalizacija se može razmatrati i kao deo uspostavljanja odnosa s potrošačima (*Customer Relationship Management*, CRM).

Illustracija: Personalizacija na sajtu eBay.com

Web sajt e-trgovine eBay.com analizira kriterijume pretraživanja klijenta na sajtu, pri čemu koristi više od 100 atributa (npr. proizvod, brend, cena) za:

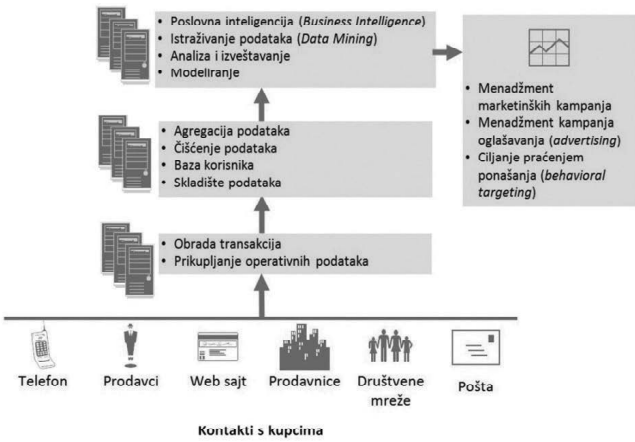
- *kreiranje lista proizvoda na početnoj stranici*, koju korisnik vidi kad god pristupi sajtu, tako da uvek vidi najnovije proizvode bez novog pretraživanja;
- *izbor reklama* po kriterijumu pretraživanja;
- *ponudu dnevnih e-mail poruka* o proizvodima koje je tražio, kako bi se vraćao da pregleda detalje ponude.

Izrada profila metodima istraživanja podataka

Metodi istraživanja podataka (*Data Mining*) i mašinskog učenja koriste se za izradu modela ili profila korisnika na osnovu primera njegove interakcije sa sistemom i ocenjenih proizvoda. Model ili profil korisnika može se koristiti za preporučivanje proizvoda, filtriranje informacija i predviđanje ponašanja korisnika u kupovini.

Odnosi s kupcima i automatizacija marketinga

Metodi istraživanja znanja, posebno mašinsko učenje, koriste se u CRM sistemima za prediktivno modeliranje.



Vrši se analiza istorijskih podataka za kvalitativno i kvantitativno predviđanje budućih događaja, npr. rezultata marketinških kampanja, lojalnosti pojedinih kupaca i optimalnu segmentaciju tržišta [17], [35].

10.3 Inteligentni interfejsi u e-trgovini

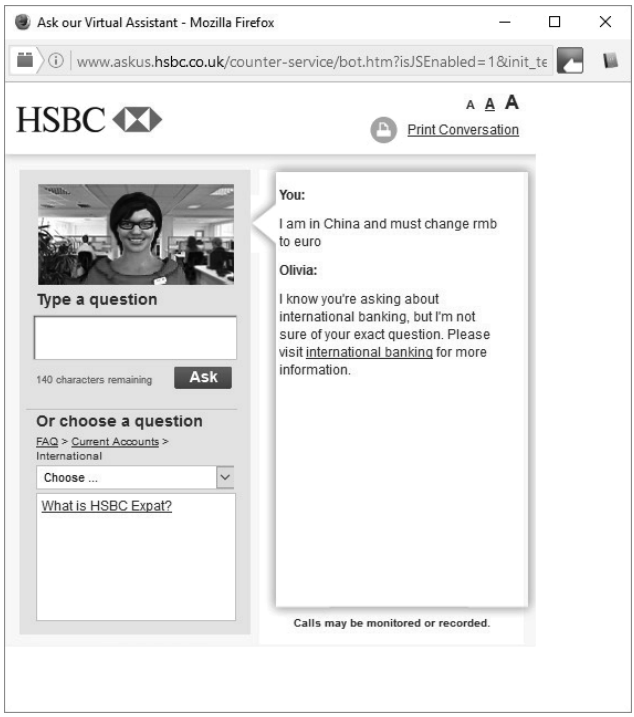
Inteligentni intefejsi omogućavaju udobniju komunikaciju, veću slobodu izražavanja, pomažu licima s posebnim potrebama i povećavaju zadovoljstvo kupaca u procesu e-trgovine.

Komunikacija u prirodnom jeziku

Tekstualna i govorna komunikacija povećavaju zadovoljstvo kupaca i pomažu licima s problemima u vizuelnoj ili taktilnoj komunikaciji da koriste računarsku tehnologiju.

Primeri programa koji koriste govornu komunikaciju su automatizovani recepcioneri i centri za pomoć i informacije.

Primeri zanimljivih programa za inteligentnu tekstualnu komunikaciju u poslovanju su inteligentni agenti za pružanje pomoći u korišćenju Internet usluga, kao što je inteligentni agent HSBC banke Olivia:



Inteligentni asistenti

Inteligentni asistenti su softverski sistemi koji pomažu korisniku u obavljanju rutinskih poslova, npr. za prijem, raspoređivanje i odgovaranje na elektronsku poštu, filtriranje neželjene pošte, davanje preporuka pri kupovini putem Interneta i organizaciju poslovnih sastanaka [36].

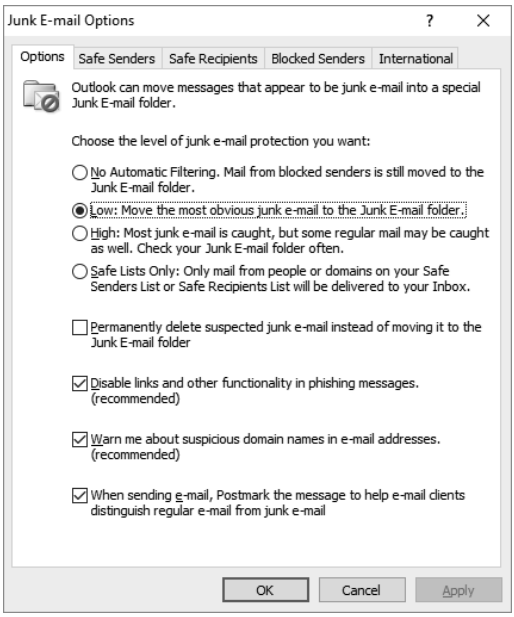
Asistenti su vezani za neposrednu interakciju i prilagođavanje potrebama ljudskog korisnika. Razlikuju se od inteligentnih *agenata*, koji mogu da funkcionišu i bez interakcije s čovekom.

Primeri inteligentnih asistenata su Microsoft *Windows i Office* meniji (*SmartMenus*), Microsoft Outlook *anti-spam* filter, *SpamBayes*, *WebWatcher* i *iShopper*.

Primer: Inteligentni asistent za filtriranje neželjene pošte

Ilustrativan primer je inteligentni asistent za rukovanje neželjenom poštom, koji je ugrađen u sistem Microsoft Outlook 2003 (i novije verzije) u obliku posebnog *anti-spam* filtera [37].

Filter je potrebno aktivirati i podesiti parametre funkcionisanja u dijalogu *Tools/Options/Junk E-mail ...*, Sl. 81, nakon čega se filtriranje poruka vrši na osnovu postavljenog nivoa osetljivosti i spiskova dozvoljenih (*Safe Senders*, *Safe Recipients*) i nedozvoljenih (*Blocked Senders*) e-adresa.



Sl. 81. Parametri rada inteligentnog asistenta za rukovanje neželjenom poštom

Spisak nedozvoljenih adresa ažurira kompanija *Microsoft*. Može preuzeti funkcijom *Help/ Microsoft Office Online/ Check for Updates*.

Ovaj filter se ne obučava na strani klijenta, korisnika *Outlook*-a, već *anti-spam* asistent koristi rezultate obučavanja *NaiveBayes* klasifikatora koje vrši *Microsoft* na svom serveru. Obučavajući skup e-poruka, klasifikovanih kao legitimne i *spam* poruke, pripremilo je više stotina hiljada *MSN Hotmail* korisnika, dobrovoljnih saradnika projekta obučavanja. Svaki *MSN*

Hotmail korisnik može se uključiti u projekt obučavanja tako što pre slanja poruke upotrebi ponuđeni taster *Report Junk Email*.

Direktno obučavanje klijenta moguće je vršiti pomoću drugih softverskih alata, kao što je npr. inteligentni *anti-spam* asistent *SpamBayes* [38], koji se može instalirati kao dodatak za Microsoft *Outlook* (iako ne i za *Outlook Express*).

10.4 Primeri inteligentne podrške e-trgovini

Inteligentna podrška elektronskoj trgovini će se ilustrovati na primeru poznatog sajta elektronske trgovine i eksperimentalnom inteligentnom asistentu za mobilne platforme.

Elektronska prodavnica knjiga Amazon

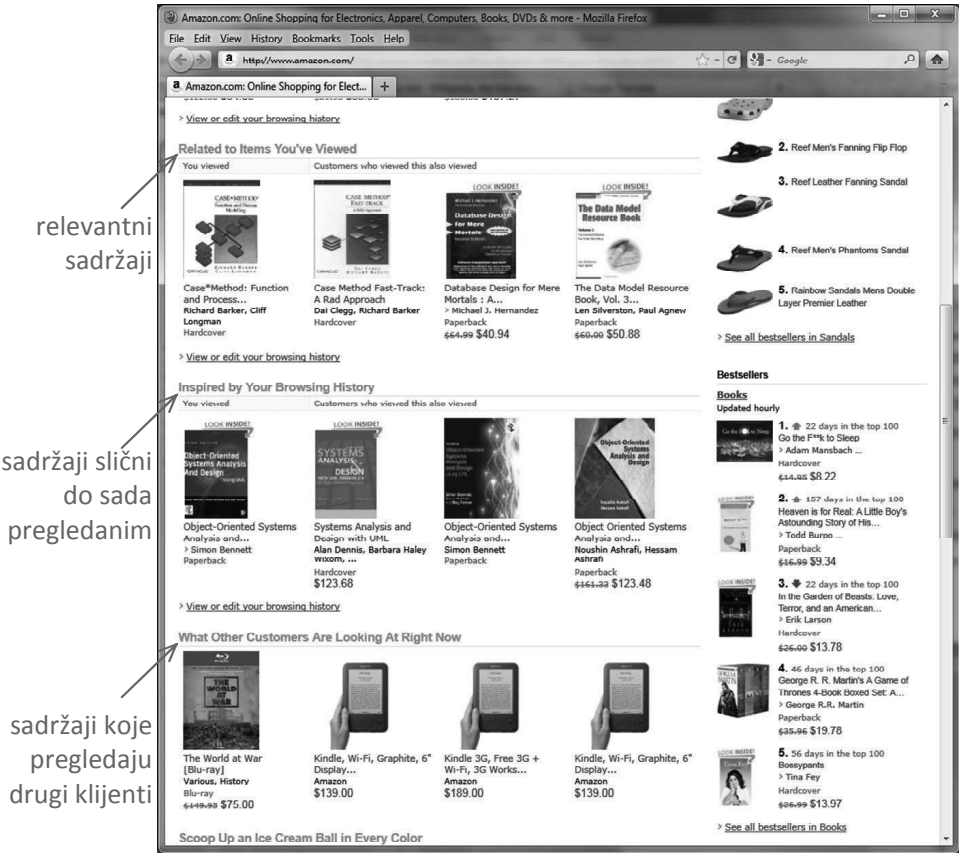
Najstarija i najveća elektronska prodavnica svoje trajanje i uspeh zasniva na primeni najsavremenijih softverskih rešenja, u koja spada i primena inteligentnih sistema za preporučivanje proizvoda, Sl. 82 [73].

Prodavnica Amazon istovremeno koristi više metoda preporučivanja, gde se preporučuju

- relevantni sadržaji, u odnosu na istorijat pretrage,
- sadržaji slični do sada pregledanim, i
- sadržaji koje upravo pregledaju drugi korisnici.

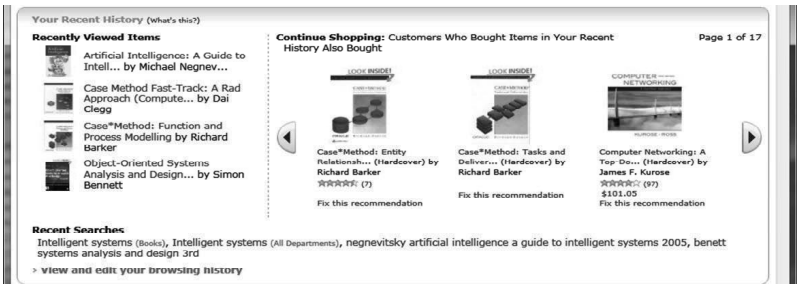
Iza prikazanih (traženih) proizvoda se nudi mala anketa (*Search Feedback*).

Na kraju Web stranice se preporučuju nedavno pregledani proizvodi (*Your Recent History*) i proizvodi koje su pregledali kupci sličnih navika (*Customers Who Bought Items in Your Recent History Also Bougtht*).



Sl. 82. Struktura kataloga sajta prodavnice Amazon

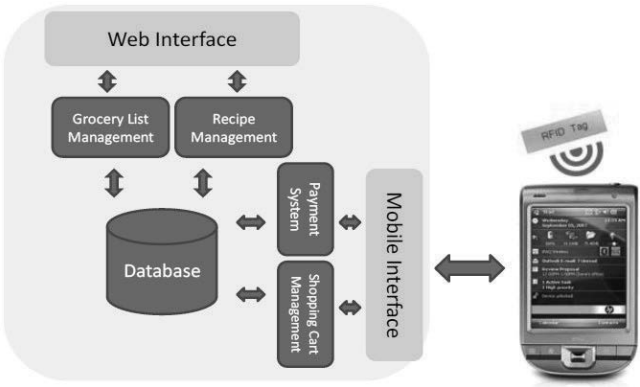
Mogu se pregledati i podaci o sopstvenom online ponašanju (*View and edit your browsing history*).



Inteligentni pomoćnik u kupovini iShopper

Sistem iShopper (*Intelligent Shopper*) pomaže kupcima u procesu kupovine u fizičkim prodavnicama, a samim trgovcima u tim prodavnicama da analiziraju individualne potrebe kupaca radi privlačenja njihovog većeg broja, Sl. 83.

Sistem povećava prodaju proizvoda korišćenjem ciljanih preporuka i promocija. Potencijalno smanjuje troškove prodavnica smanjenjem broja potrebnih prodavaca (ljudski resursi) i pomaže u izgradnji dugoročnih odnosa sa klijentima.



Sl. 83. Princip funkcionisanja inteligentnog pomoćnika u kupovini

Koristi se RFID tehnologija označavanja proizvoda i sistemi poslovne inteligencije (BI).

Pitanja za ponavljanje

1. Da li je praćenje rada korisnika standardna funkcija Web sajta e-trgovine?
2. Koje vrste direktne komunikacije s predstavnicima kompanije koriste Web sajtovi e-trgovine?
3. Gde se nalazi programski kod korpe za kupovinu?
4. Koje su dodatne komponente modela ponašanja online potrošača?
5. Kako se naziva online evivalent ponašanja (kretanja) potrošača u fizičkoj prodavnici?
6. Gde se u sistemu prati/evidentira ponašanje online potrošača?
7. Navedite elemente inteligentnog ponašanja
8. Šta su inteligentni programski sistemi?
9. Navedite bar tri metoda realizacije inteligentnih sistema
10. Navedite primene inteligentnih agenata u B2C e-trgovini
11. Koja je osnovni cilj sistema za preporučivanje proizvoda u e-trgovini?
12. Koji sistemi preporučivanja pretvaraju različite izvore i oblike preporuka u nove, pouzdanije preporuke?
13. Koja se matematička disciplina koristi u modeliranju aukcija?
14. Koji se ulazne informacije koriste za izradu profila korisnika metodima istraživanja znanja?

15. Koje su osnovne prednosti komunikacije u prirodnom jeziku u sistemima e-trgovine?
16. Koja je osnovna razlika između inteligentnih agenata i inteligentnih asistenata?

11 Implementacija sistema za podršku odlučivanju

1. Uvod u implementaciju sistema za podršku odlučivanju
2. Softverske tehnologije i alati za implementaciju
3. Sistemi zasnovani na elektronskim tabelama
4. Implementacija ekspertnih sistema
5. Sistemi mašinskog učenja
6. Menadžment razvoja sistema za podršku odlučivanju

U ovom poglavlju ukratko se izlažu teorijski i praktični aspekti projektovanja i realizacije sistema za podršku odlučivanju. Opisuju se osnovne tehnologije i rešenja za implementaciju nekih klasa sistema za podršku odlučivanju. Daje se pregled sistema mašinskog učenja, kao i neki pristupi vrednovanju sistema za podršku odlučivanju.

11.1 Uvod u implementaciju sistema za podršku odlučivanju

Projektovanje i implementacija softverskih sistema obuhvata proces konverzije specifikacije sistema u izvršni program.

- *Projektovanje* softvera se sastoji u projektovanju softverske strukture koja realizuje specifikacije softverskog sistema.
- *Implementacija* softvera je proces prevođenja projektovane softverske strukture u izvršni program.

Aktivnosti projektovanja i implementacije su tesno povezane i mogu biti međusobno isprepletene.

Tehnologija implementacije sistema za podršku odlučivanju

Nestrukturirani i delimično strukturirani problemi nemaju opšte, sveobuhvatno rešenje. Posledica je da se za njihovo rešavanje razvijaju manji sistemi za podršku odlučivanju, specifične namene.

Postoje tri tehnološka nivoa ili načina realizacije sistema za podršku odlučivanju (DSS):
1. Upotreba specifičnih sistema za podršku odlučivanju,
2. Upotreba integrisanih alati (generatora DSS), kao što je npr. *Excel*,
3. Upotreba programskih jezika za programiranje novog sistema (npr. Java).

Softverski alati koji se koriste za razvoj sistema za podršku odlučivanju (DSS) su, npr.

- Programski jezici opšte namene, kao što su C/C++, Java, C# i Python,
- Programski jezici IV generacije (4GL),
- Softver za interaktivnu analitičku obradu (OLAP) sa skladištem ili velikom bazom podataka,
- Integrisani razvojni alati (sadrže generator, *engine*),
- Specifični namenski generatori pojedinih klasa kompletnih sistema za podršku odlučivanju,
- Razvojni CASE alati, i
- Integrisana razvojna okruženja (kombinovani alati).

Primer efikasnog rešenja konkretnog problema jedne organizacije i način realzacije razvoja ilustrovaće se na malom primeru [1].

Ilustracija: OSRAM

Industrija osvetljenja *Osram Sylvania* je severnomeričko odeljenje *Osram Gmbh*, sa 12.500 zaposlenih, teritorijalno raspoređenih 30 lokacija, s godišnja prodajom od 3,7 milijardi dolara.

Kompanija je uočila problem funkcionsanja menadžmenta ljudskih resursa, odnosno pronalaženja, zapošljavanja i praćenja naučnika-specijalista, inženjera i privremeno angažovanih lica. Rutinsko praćenje promena se radilo ručno, što je postupak produžavalo na nekoliko meseci.

Izabrano rešenje problema je razvoj softverske podrške menadžmenta ljudskih resursa, odnosno izrada malog interaktivnog Web portala i odgovarajućih intranet aplikacija korišćenjem kolaborativne platforme IBM *Lotus Notes*. Razvoj je realizovao mali, motivisan tim, a kao metodologija je izabran evolutivni prototipski razvoj.

Rezultat je sistem za podršku čija je primena ostvarila interne uštede u administraciji od 500.000 dolara, a efekti planiranja u periodu od 3 godine su procenjeni na 1,5 milion dolara [1].

Veza sa drugim sistemima

Veza sa drugim elementima informacionog sistema se realizuje upotrebom odgovarajuće arhitekture i softverskih alata za integraciju informacionih sistema. Softverski alati za integraciju sa informacionim sistemima (ERP, CRM) obuhvataju načelno sve Web tehnologije, specijalizovani softver za

integraciju (*Enterprise Integration Application*, EAI) i odgovarajuće programske biblioteke aplikativnog sloja (*middleware*).

Jedan od savremenih pristupa realizaciji sistema je primena Web servisa i servisno orijentisana arhitekture. Web servis je diskretna jedinica ponovo upotrebljive poslovne (ili tehničke) funkcije. Servisno orijentisana arhitektura (SOA) je arhitekturni stil projektovanja prilikom razvoja i integracije aplikacija. U ovom pristupu se aplikacije sklapaju (*assembly*) od dobro definisanih komponenti. Jedan o načina za opis Web servisa je WSDL (*Web Services Description Language*) baziran na jeziku XML.

Softverski paketi i sistemi za podršku menadžmentu (MSS)

Savremeni sistemi za podršku menadžmentu su specijalizovani softverski paketi za podršku menadžmentu (MSS *suites*). Posmatraju se kao kombinacija sistema za podršku odlučivanju (DSS) i sistema poslovne inteligencije (BI). Sistemi poslovne inteligencije mogu biti uključeni i/ili korišćeni u razvoju MSS.

Na Sl. 84. prikazan je međusobni odnos svojstava najpoznatijih softverskih alata poslovne inteligencije i analitike (Gartner, 2016).



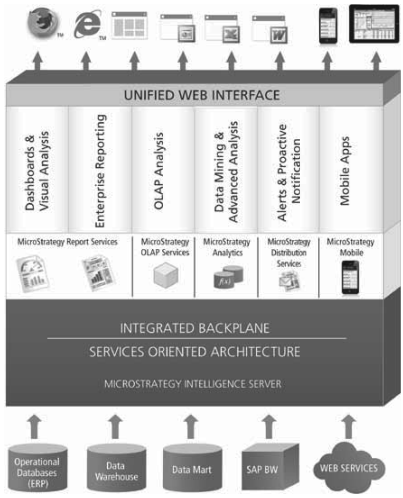
Sl. 84. Softverski paketi poslovne inteligencije i analitike

Specijalizovani softverski paket za podršku menadžmentu (*MSS suite*) je skup integrisanih alata za razvoj sistema za podršku odlučivanju, koji obuhvataju upravljačke table, modele procesa i softver za različite analize i izveštavanje.

Poznati proizvođači specijalizovanih softverskih paketa za podršku menadžmentu su:

1. Microstrategy Analytics Platform,
2. Oracle Hyperion System 9 BI+,
3. SAP BusinessObjects XI,
4. Microsoft BizTalk Server 2013,
5. Oracle E-Business Suite,
6. IBM WebSphere Commerce Suite.

Servisno orijentisana arhitektura softvera za podršku menadžmentu kompanije Microstrategy [68] prikazana je na Sl. 85.



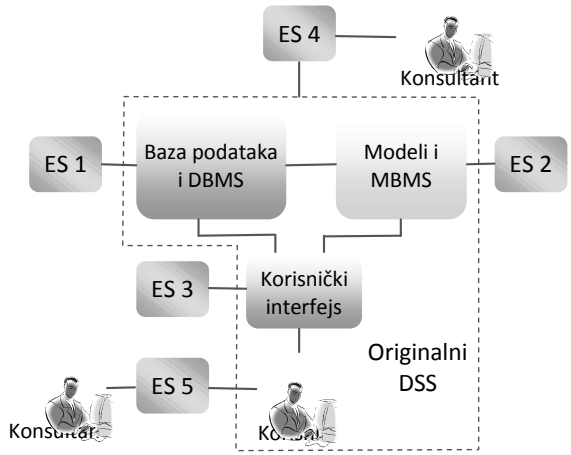
Sl. 85. Arhitektura softvera za podršku menadžmentu kompanije Microstrategy

Integracija s drugim informacionim sistemima

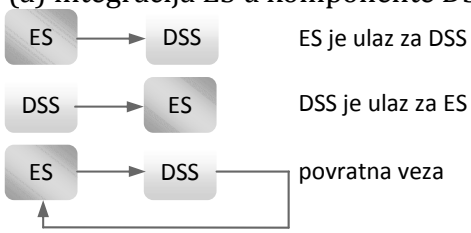
Integracija informacionih sistema povećava efikasnost i efektivnost podržanih zadataka. Osnovne vrste integracije su [1]:

1. *Funkcionalna integracija* - istovremena upotreba različitih aplikacija za rešenje zadatka (npr. DSS+ES), koja se najlakše tehnološki realizuje korišćenjem Web tehnologija.
2. *Fizička integracija* - kombinovanje hardvera, softvera i komunikacija.

Sisteme za podršku menadžmentu moguće je komponovati integracijom sistema za podršku odlučivanju (DSS) i ekspertnih sistema (ES). Sistem za podršku odlučivanju je svaki sistem koji podržava donosioce odluka, dok je ekspertni sistem inteligentni sistem zasnovan na znanju (*knowledge-based*). Pri tome se ekspertni sistem se može integrisati u neke (ili sve) komponente DSS, Sl. 86 (a) ili se može koristiti kao posebnu komponentu u okviru šireg DSS, Sl. 86 (b) [1].



(a) integracija ES u komponente DSS



(b) ES kao posebna komponenta DSS

Sl. 86. Načini integracije sistema za podršku odlučivanju i ekspertnih sistema

Performanse sistema i računarstvo u oblaku

Sistemi za podršku odlučivanju često se zasnivaju na metodima istraživanja podataka ili numeričkim metodima, koji za visoki kvalitet rezultata zahtevaju znatne procesorske i memorijske resurse.

Pojedinačni računarski sistemi nemaju dovoljno resursa za obradu velikih skupove podataka (terabajti, petabajti itd.), kao ni za složene numeričke obrade. Osim toga, standardni softverski alati, koji se koriste u podršci odlučivanju, mogu efikasno da obrade samo podatke koji se nalaze u radnoj ili virtuelnoj memoriji jednog, eventualno skupog višeprocesirskog računarskog sistema.

Dodatni računarski resursi mogu se obezbediti nabavkom ili iznajmljivanjem računarskog sistema zadovoljavajućih performansi, ali su pojedinačni serveri skupi i uvek ograničenih mogućnosti. Mreže ili farme servera omogućavaju relativno jeftino povećanje raspoloživih resursa. Računarstvo u oblaku (*cloud computing*) je pristup koji omogućava korišćenje računarskih resursa putem Interneta po prihvatljivoj ceni.

Osnovni problem korišćenja masivnih mrežnih računarskih konfiguracija je u distribuiranoj arhitekturi ovih sistema, pošto se sastoje od mreže povezanih

servera ili čvorova (*nodes*). Vremensko i prostorno distribuiranje metoda obrade na elemente sistema je poseban problem računarstva u oblaku. Komercijalni sistemi za istraživanje podataka, koji mogu efikasno da analiziraju ogromne skupove podataka, kao npr. *SAS Enterprise Miner* i *IBM SPSS Modeler*, mogu samostalno da prilagode obradu podataka distribuiranoj arhitekturi računarskog oblaka. Jednostavniji sistemi za uspešno korišćenje računara u oblaku zahtevaju dodatno programiranje, za šta postoje programske biblioteke kao što je *Apache Hadoop* [39], namenjene sistemima otvorenog koda, kao što je sistem *R* [40], [69].

Primeri načina implementacije sistema za podršku odlučivanju

Kao ilustracije različitih pristupa razvoju i realizaciji sistema za podršku odlučivanju opisaće se sistem za podršku menadžmentu kompanije Sterngold i inteligentni sistem za podršku logistike hongkonškog aerodroma.

Primer 1: Sterngold MSS

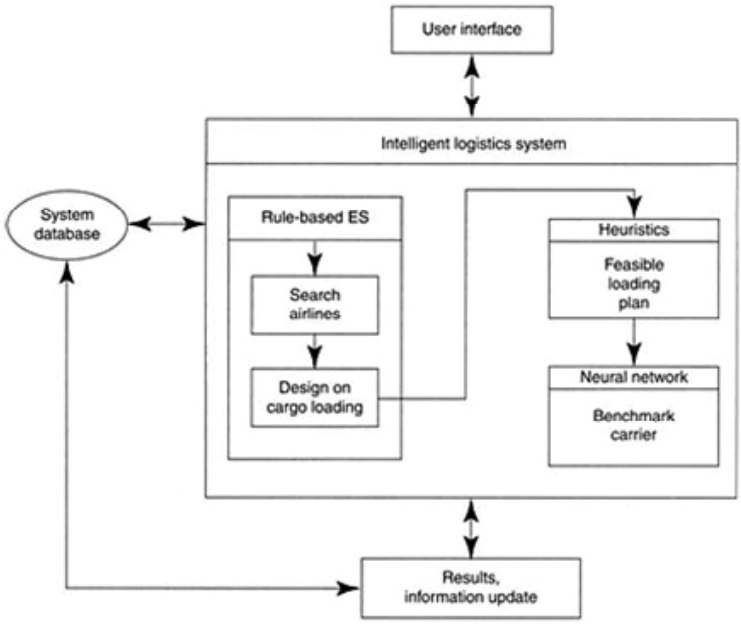
Kompanija postoji čitav vek, proizvodi više od 4.000 stomatoloških proizvoda i ima oko 5,5 milona potrošača. Narudžbe su male vrednosti, ali su česte i hitne.

Kompanija je zaključila da joj je potreban sistem za podršku menadžmentu (MSS) radi bržeg pristupa tehnologijama i kupcima. Pošto ima svega 90 zaposlenih, od čega samo jednog u IT sektoru, izabran je metod izmeštenog razvoja (*outsourcing*), što je omogućilo kratko vreme razvoja (samo 3 meseca), najsavremeniju e-tehnologiju (na slici) i brz izlazak na tržište.



Primer 2: Inteligentni sistem za podršku logistike (ILSS, Hong Kong)

Inteligentni sistem za podršku logistike hongkonškog aerodroma ILSS (*Intelligent Logistics Support System*) daje ekspertske savete u oblasti vazdušnog transporta, između ostalog za pronalaženje prevoznika i kontejnera, *optimizaciju utovara* i planiranje praćenja tereta. Realizovan je integracijom ekspertnih sistema i neuronskih mreža, kao na slici [1]:



Ekspertni sistem zasnovan na pravilima obezbeđuje stručne savete i generiše heuristike za podršku izrade planova utovara. Neuronska mreža se koristi za predviđanje neočekivanih događaja, npr. promene reda letenja.

Sistem je testiran i procenjeno je da je oko 30% uspešniji od tima veoma zauzetih planera koji broji 90 ljudi.

11.2 Softverske tehnologije i alati za implementaciju

U softverske tehnologije i alate za implementaciju sistema za podršku odlučivanju spadaju::

- Programski jezici,
- Objektne programske biblioteke,
- Softverski alati za istraživanje podataka i razvoj ekspertnih sistema,

- Softverski paketi i sistemi za podršku menadžmentu (MSS).

11.3 Sistemi zasnovani na elektronskim tabelama

Sisteme za podršku odlučivanju je moguće implementirati u različitim programskim jezicima i tehnologijama. Posebno je praktična upotreba elektronskih tabela (*spreadsheets*) i njihovih dodataka. Elektronske tabele omogućavaju krajnjim korisnicima relativno jednostavnu implementaciju različitih matematičkih modela zahvaljujući strukturi podataka tipa tabele, formulama i velikim bibliotekama ugrađenih funkcija za staističke, logičke i druge obrade podataka. Sistemi elektronskih tabela imaju svoj programski jezik, brojne ugrađene funkcije i dodatke (*Add-In*).

Sistem Microsoft *Excel* raspolaže velikom bibliotekom od oko 400 ugrađenih matematičkih, statističkih, finansijskih, inženjerskih i drugih funkcija, koje se mogu koristiti u formulama tabele za kreiranje matematičkih modela. Generator pseudoslučajnih brojeva omogućava simulaciju slučajnih događaja, a programski jezik *Visual Basic for Application* programiranje složenih dinamičkih modela i izradu aplikacija, koje imaju pristup i drugim elektronskim tabelama, fajlovima i bazama podataka.

Postoje softverski sistemi, kao što su softver za rešavanje modela linearnog programiranja *Solver* i softver za analizu rizika Palisade @Risk, koji su implementirani kao Microsoft *Excel* dodaci.

Krajnji korisnik može direktno da koristi standardne analize kao što su *what-if* i *goal-seek*, kao i da postavlja upite, kojima može radi analize da izdvoji deo podataka iz neke obimnije baze.

Sistemi elektronskih tabela omogućavaju lako i brzo kreiranje grafičkih prikaza i izveštaja, kojima se rezultati analize prikazuju jasno i pregledno.

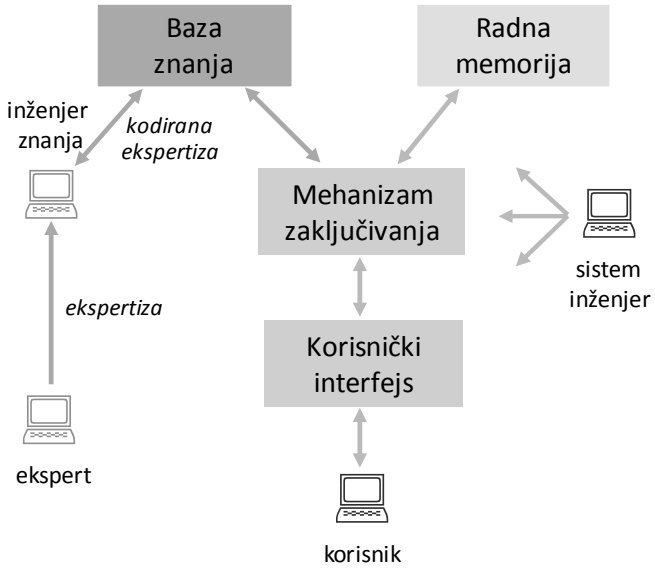
Ograničenje ovakvih sistema je pre svega ograničeni obim podataka koji se može analizirati, pošto elektronske tabele imaju ograničene dimenzije (do milion redova i 16 hiljada kolona), a svi podaci moraju da stanu u radnu memoriju računara.

11.4 Implementacija ekspertnih sistema

Ekspertni sistem obezbeđuje da se znanje ljudskog eksperta može uneti u računar i koristiti za rešavanje problema u nekoj oblasti primene. Očekuje se da će se takav sistem ponašati kao inteligentni asistent u nekoj usko specijalizovanoj oblasti ekspertize, odnosno da će rešavati iste probleme koje može da reši ekspert. Sistem treba da prihvata i integriše novo znanje, da ga prikazuje u razumljivom obliku, da prihvata jednostavne rečenice prirodnog jezika, kao i da objašnjava svoje zaključke na razumljiv način [18].

Struktura i organizacija razvojnog tima

Razvojni tim projekta razvoja ekspertnog sistema se sastoji od menadžera projekta, eksperta, inženjera znanja, softverskog inženjera (programera) i krajnjeg korisnika [18], Sl. 87.



Sl. 87. Uloga članova razvojnog tima ekspertnog sistema

Ekspert je obrazovana i obučena osoba, koja može da rešava probleme na nekom specifičnom području. Njegov zadatak je da svoju ekspertizu prenese u računarski sistem, što ga čini veoma važnim članom tima i od njega zahteva dosta vremena provedenog u radu na projektu.

Inženjer znanja (*knowledge engineer*) je zadužen za projektovanje, izgradnju i testiranje ekspertnog sistema. Definiše odgovarajući zadatak sistema i sarađuje s ekspertom da ustanovi koje metode rešavanja problema koristi i određuje način njihovog predstavljanju u ekspertnom sistemu. Bira razvojni softver, npr. ljusku ekspertnog sistema ili programski jezik za predstavljanje znanja, koji može i sam da kodira. Inženjer znanja testira sistem u toku razvoja, integriše ga u radno okruženje i učestvuje u održavanju nakon završetka razvoja.

Programer je osoba koja je odgovorna za programiranje sistema u nekoj od softverskih tehnologija, odnosno programskih jezika koji se koriste za ove namene. Primer su jezici koji se koriste za razvoj sistema veštačke inteligencije, LISP, Prolog i OPS5, ali se mogu koristiti i proceduralni programski jezici kao C/C++, Java, C#, Python ili Pascal. Ako se koristi ljuska ekspertnog sistema, ne moraju se programirati strukture za predstavljanje znanja, mehanizam zaključivanja i korisnički interfejs, ali se programer može uključiti u testiranje i instalaciju sistema.

Menadžer projekta brine o razvojnom timu, odgovara za realizaciju ciljeva razvoja, poštovanje rokova i budžeta.

Krajnji korisnik je osoba koja će koristiti ekspertni sistem kao savetnika u donošenju složenih odluka u oblasti za koju je razvijan. To može biti hemičar, koji analizira molekularnu strukturu materijala, lekar, geolog ili operater elektroenergetskog postrojenja. Korisnik sitema pomaže u razvoju i daje ocenu prihvatljivosti i uspešnosti ekspertnog sistema.

Upotreba ljuski ekspertnog sistema i pratećih razvojnih alata ukida ulogu programera, smanjuje ulogu inženjera znanja i omogućava da manje ekspertne sisteme može samostalno da razvije i jedna osoba.

Softverske tehnologije za realizaciju ekspertnih sistema

U realizaciji ekspertnih sistema koriste se programski jezici opšte namene, ljuske ekspertnih sistema i namenski ekspertni sistemi za šire oblasti ekspertize.

Programski jezici opšte namene

Programski jezici opšte namene se mogu podeliti na algoritamske, deklarativne, funkcionalne i jezike namenjene logičkom programiranju.

Algoritamski ili proceduralni programski jezici koriste se za realizaciju računskih postupaka ili algoritama. To su npr. C/C++, Java, C#, R i Python.

Deklarativni programski jezici se koriste za opis rezultata računanja, bez navođenja detalja postupka. U ovu grupu spadaju npr. SQL i Prolog.

Funkcionalni jezici su jezici koji računanje opisuju u obliku funkcija. U njih se ubrajaju npr. LISP, Erlang i F#.

Jezici *logičkog programiranja* koriste se tako što problem i rezultat računanja opisuju kao logičke izraze, a sam proces računanja odvija se kao procedura dokazivanja njihove istinitosti. Primeri ovih jezika su Prolog [51] i Datalog.

Mala ilustracija primene različitih programskih jezika, algoritamskih (C, Python), funkcionalnih (LISP, F#) i jezika logičkog programiranja (Prolog) za računanje funkcije faktorijel [94]:

Ilustracija: C (iterativna verzija)

```
unsigned long factorial(unsigned long n)
{
 if (n == 0)
   return 1;
 else
   return n * factorial (n - 1);
}
```

Ilustracija: Python (iterativna verzija)

```
def factorial(n):
    result = 1
    for i in range(1, n+1):
        result *= i
    return result
```

Ilustracija: LISP

```
defun factorial (N)
 (if (= N 1)  1
   (* N (factorial (- N 1)))))
```

Ilustracija: F#

```
let inline factorial n = Seq.reduce (*) [LanguagePrimitives.GenericOne .. n ]
```

Ilustracija: Prolog

```
factorial(X, 1) :- X<2.
factorial(X, F) :- Y is X-1, factorial(Y,Z), F is Z*X.
```

Ljuske ekspertnih sistema (Expert Systems Shells)

Najpopularnija tehnologija realizacije ekspertnih sistema su sistemi zasnovani na produkcionim pravilima (*rule-based*). Veliki broj ekspertnih sistema je realizovan za različite oblasti primene, kao što su poslovanje, inženjering, medicina, geologija, energetika i rudarstvo [18].

Osnovni softverski alati za ovu klasu sistema danas su ljuske ekspertnih sistema (*expert system shell*), posebno za sisteme zasnovane na pravilima. Koristeći ove alate, razvojni timovi se usmeravaju na znanje, kao najvažniji element sistema, dok se razvoj programske podrške veoma pojednostavljuje, a time skraćuje vreme razvoja sistema.

Ljuska ekspertnog sistema je ekspertni sistem koji ima praznu bazu znanja. Unosom odgovarajućih pravila, ekspertni sistem može da pruži podršku u rešavanju neke klase nestrukturiranih problema u izabranoj oblasti. Baza znanja se razvija za specifično područje ekspertize, korišćenjem tehnika prikupljanja, odnosno izvlačenja znanja (*knowledge acquisition/elicitation*)

Primeri poznatijih ljuski ekspertnih sistema su:

- EMYCIN (empty MYCIN) / TMYCIN,
- Exsys CORVID,
- CLIPS (*C Language Integrated Production System*),
- JESS (*Java Expert Systems Shell*),

- IBM ILOG (*Business Rule Management System*, C/C++,.Net,*Java*),
- Expertise2Go,
- JLog/JScriptLog (Prolog u jeziku Java/JavaScript).

Za kreiranje baze znanja koriste se dodatni programski alati, kao što su sistemi za istraživanje podataka, koji se u ovom kontekstu nazivaju sistemi za otkrivanje znanja u podacima (*Knowledge Data Discovery*).

Ljuska ekspertnog sistema Jess

Mehanizam zaključivanja za Java platformu predstavlja proširenje programskog jezika CLIPS, interpretativnog (skript) jezika koji omogućava pristup svim Java aplikativnim programskim interfejsima (API). Uključuje kompletno razvojno okruženje zasnovano na *Eclipse* platformi. Sadrži proširenu verziju algoritma zaključivanja *Rete* za obradu pravila [47]. Primer dela baze znanja u sintaksi sistema *Jess* je:

```
(deftemplate blood-donor (slot name) (slot type))
(deffacts blood-bank ; put names & their types into working memory
    (blood-donor (name "Alice")(type "A"))
    (blood-donor (name "Agatha")(type "A"))
    (blood-donor (name "Bob")(type "B"))
    (blood-donor (name "Barbara")(type "B"))
    (blood-donor (name "Jess")(type "AB"))
    (blood-donor (name "Karen")(type "AB"))
    (blood-donor (name "Onan")(type "O"))
    (blood-donor (name "Osbert")(type "O"))
)
(defrule can-give-to-same-type-but-not-self ; handles A > A, B > B, O > O, AB > AB, but not N1 > N1
    (blood-donor (name ?name)(type ?type))
    (blood-donor (name ?name2)(type ?type2 &:(eq ?type ?type2) &: (neq ?name ?name2)  ))
    =>
    (printout t ?name " can give blood to " ?name2 crlf)
)
(defrule O-gives-to-others-but-not-itself ; O to O cover in above rule
    (blood-donor (name ?name)(type ?type &:(eq ?type "O")))
    (blood-donor (name ?name2)(type ?type2 &: (neq ?type ?type2) &: (neq ?name ?name2)  ))
    =>
    (printout t ?name " can give blood to " ?name2 crlf)
)
(defrule A-or-B-gives-to-AB ; case O gives to AB and AB gives to AB already dealt with
    (blood-donor (name ?name)(type ?type &:(or (eq ?type "A") (eq ?type "B" ))))
    (blood-donor (name ?name2)(type ?type2 &: (eq ?type2 "AB")  &: (neq ?name ?name2)  ))
    =>
    (printout t ?name " can give blood to " ?name2 crlf)
)
;(watch all)
(reset)
(run)
```

Ljuska ekspertnog sistema Expertise2Go

Ljuska ekspertnog sistema *Expertise2Go* je komercijalni sistem, ali ima potpuno funkcionalnu besplatnu verziju. Sastoji se od više komponenti, koje se isporučuju u obliku Java arhiva:

- *e2gRuleEngine.jar* - mehanizam zaključivanja (*inference engine*),
- *e2gRuleWriter.jar* - interaktivni grafički orijentisan editor pravila,
- *e2gRuleInduction* - program za induktivno učenje metodom ID3, koji može da induktivno generiše kompletnu bazu znanja,
- *e2gDroid* - verzija ljuske ekspertnog sistema za mobilne uređaje (Android).

Baza znanja ekspertnog sistema je u tekstualnom formatu, koji je isti za standardnu i mobilnu verziju ljuske. Baza znanja se sastoji od produkcionih pravila i definicije interfejsa.

Sintaksa produkcionih pravila ljuse Expertise2Go je standardna:

RULE [*naziv*]
If [*atribut*] *relacija vrednost*
And|Or
...
Then [*klasifikacioni atribut*]="*oznaka klase*" @ *CF*

Elementi uslova pravila su logički izrazi, koji se sastoje od naziva atributa (u uglastim zagradama), relacije (=, >, >=, <, <=) i odgovarajuće vrednosti, koja zavisi od tipa atributa.

Neegzaktno znanje predstavlja se pomoću faktora pouzdanosti (*Certainity Factor*, CF), koji se zadaje procentualno, kao decimalna vrednost. Ako se izostavi, podrazumeva se da je pouzdanost pravila 100%.

Ako se na neki primer u procesu zaključivanja ne može primeniti nijedno pravilo, koristi se podrazumevajuće (*default*) pravilo, koje se zadaje posebnom deklaracijom:

DEFAULT [*klasifikacioni atribut*]="*oznaka klase*" @ *CF*

Sistem inicijalno ne koristi pravila čija je pouzdanost manja od 80%. Ako se u bazi znanja koriste i neka pravila čija je pouzdanost ispod zadanog praga, može se upotrebiti direktiva koja definiše novu vrednost praga:

MINCF *minimalna pouzdanost pravila*

Cilj zaključivanja definiše se posebnom deklaracijom:

GOAL [*klasifikacioni atribut*]

Sintaksa jezika za predstavljanje znanja predviđa i pisanje komentara u posebnoj deklaraciji, radi dokumentovanja baze znanja, npr:

REM *tekst komentara u bazi znanja*

Definicija interfejsa poseban je deo baze znanja, koji definiše način postavljanja upita korisniku, za svaki atribut posebno. Sintaksa dijaloga definiše se direktivom PROMPT čiji oblik zavisi od vrste atributa.

Za numeričke atribute koristi se dijalog tipa *Numeric*:

PROMPT [*atribut*] **Numeric CF**
"*pitanje korisniku*"
"minimalna vrednost atributa"
"maksimalna vrednost atributa"

Za nenumeričke atribute može se koristiti dijalog tipa *MultChoice*:

PROMPT [*atribut*] **MultChoice CF**
"*pitanje korisniku*"
vrednost 1 atributa
…
vrednost n atributa

Za atribute s logičkim odgovorima, može se upotrebiti skraćena forma:

PROMPT [*atribut*] **YesNo CF**
"*pitanje korisniku*"

Načelni redosled elemenata u bazi znanja ekspertnog sistema je:

1. definicije produkcionih pravila,
2. definicija podrazumevajućeg pravila,
3. definicija cilja,
4. definicije dijaloga, za svaki atribut iz uslova pravila,
5. ostale deklaracije.

Ekspertni sistem se na desktop računaru pokreće kao Java aplikacija, kojoj se kao parametar prenosi naziv fajla u kome je baza znanja. Detaljnije informacije o instalaciji i korišćenju sistema dati su u Prilogu C.

11.5 Sistemi mašinskog učenja

Sistemi mašinskog učenja se koriste kao alati za razvoj sistema za podršku odlučivanju, kao i za njihovu implementaciju. Metodi mašinskog učenja se koriste kao komponente ili podsistemi, koji obezbeđuju inteligentno ponašanje sistema za podršku prilikom rešavanja nestrukturiranih problema. Mogu se implementirati sasvim novi metodi, ali se često koriste programske biblioteke metoda mašinskog učenja u različitim tehnologijama.

Postoje brojni programski alati za mašinsko učenje i istraživanja podataka, koji mogu biti samostalni integrisani sistemi, samostalni programi mašinskog učenja, delovi komercijalnih sistema poslovne inteligencije (*Business Intelligence Suites*, BI) ili specifični Web alati.

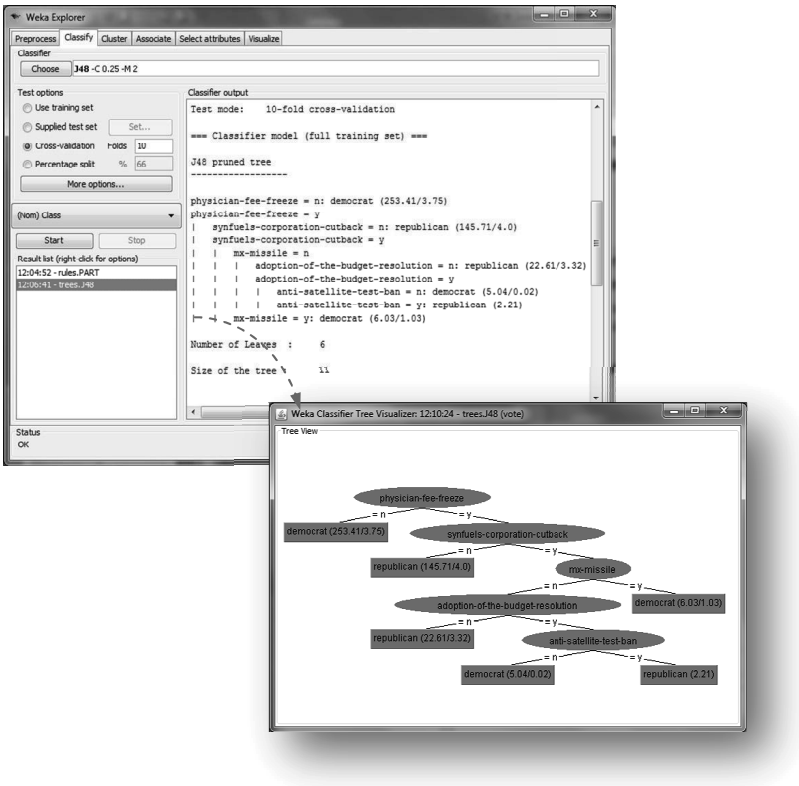
Samostalni integrisani alati

Poznati i popularni alati za istraživanje podataka otvorenog koda (*Open source Data Mining Tools*) su [64]

- Weka,
- KNIME,
- Rapid Miner.

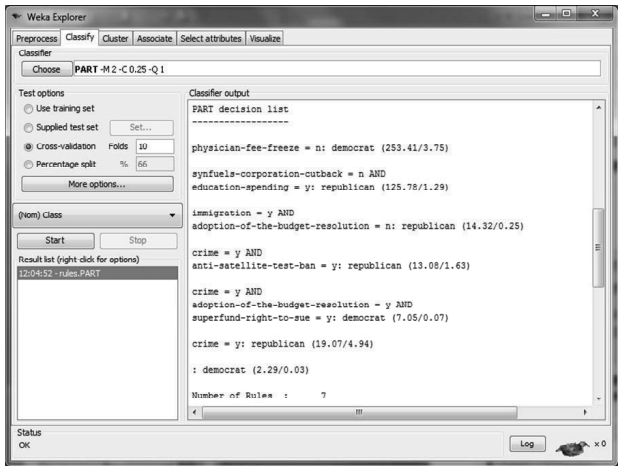
Primer: Sistem Weka

Naziv WEKA (*Waikato Environment for Knowledge Analysis)* asocira na retku zaštićenu novozelandsku pticu Meka [70]. Predstavlja softver otvorenog koda (*Java*), s velikim brojem ugrađenih metoda za mašinsko učenje *klasifikacija*, predviđanje kontinualnih veličina (*regresija*), grupisanje (*clustering*), analizu *asocijacija* i vizualizaciju podataka i naučenog znanja.



Sl. 88. Rezultat učenja stabala odlučivanja (vote.arff)

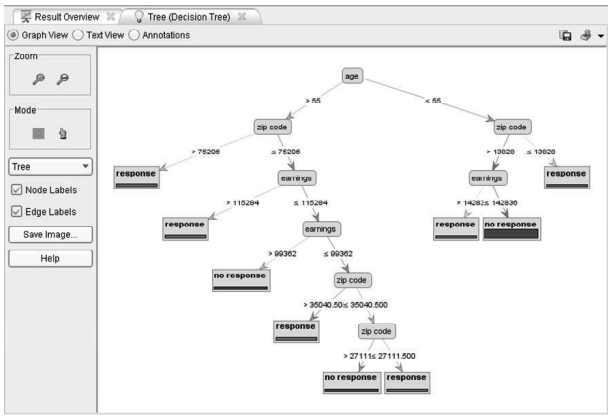
Kao ulazni podaci su uzeti javno dostupni rezultati glasanja iz jednog perioda rada kongresa (vote.arff). Cilj učenja je skup pravila ili stablo odlučivanja za određivanje partijske pripadnosti članova kongresa samo na osnovu glasanja o pojedinim pitanjima. Stablo odlučivanja se može vizualizovati. Grafički prikaz naučenog stabla odlučivanja je na Sl. 88.



Sl. 89. Rezultat učenja pravila odlučivanja (vote.arff)

Primer: Sistem Rapid Miner

Sistem otvorenog koda Rapid Miner [81] kompanije Rapid-I GmbH, Dortmund, Germany www.rapid-i.com. Prethodna verzija sistema, poynata kao *Yale*, razvijena kao univerzitetski projekt. Sistem predstavlja alat za istraživanje podataka i predviđanja (*Data Mining and Predictive Analytics*).



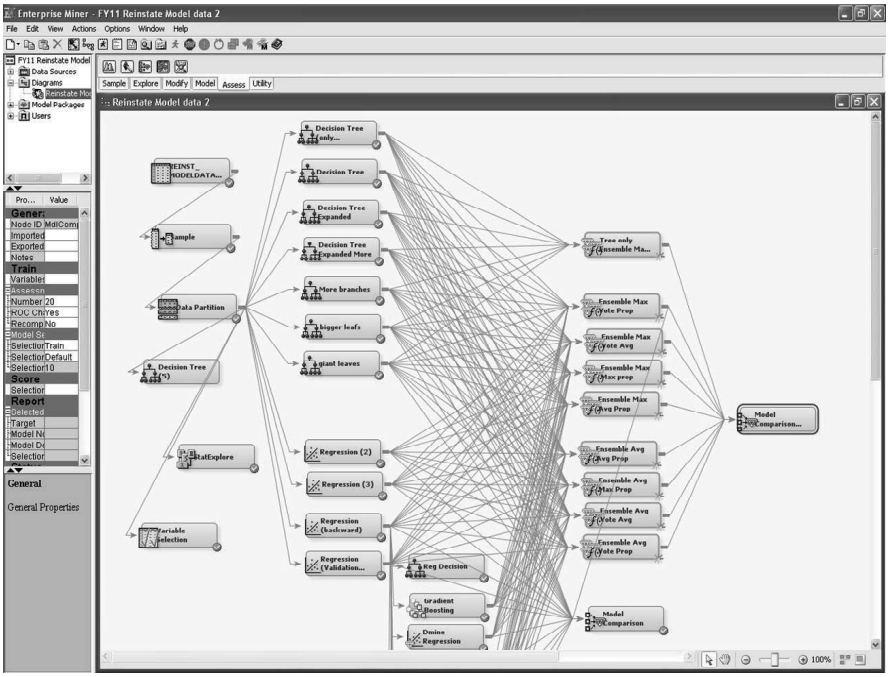
Integrisani sistem za istraživanje podataka metodima induktivnog učenja autora ove knjige pod nazivom *Empiric* prikazan je u Prilogu E.

Poznatiji komercijalni alati za istraživanje znanja (*Commercial Data Mining Tools*) su [64]:

- SAS Institute *Enterprise Miner*;
- IBM *Intelligent Miner* (tied to DB2);
- IBM *Cognos*;
- IBM *SPSS* Miner (ranije *Clementine*);
- Oracle, *Oracle BI 11g*;
- Microsoft *SQL Server 2005/2008 BIDS*;
- Angoss *KnowledgeSTUDIO*;
- KXEN *Analytic Framework*;
- Megaputer *PolyAnalyst*.

Ilustracija: SAS Institute Enterprise Miner

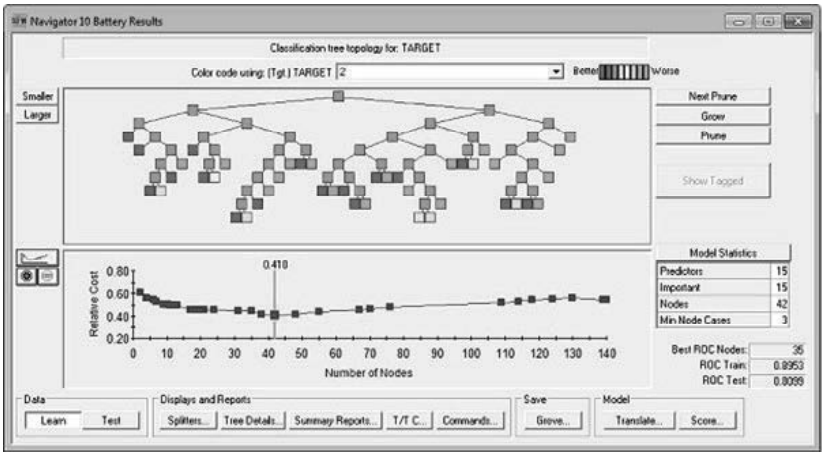
SAS Institute *Enterprise Miner* [82] je komercijalni sistem za istraživanje podataka, koji omogućava rad s veoma velikim obimom podataka i kompleksnim modelima, i koji može efikasno da koristi tehnologiju računarstva u oblaku.



Samostalni programi mašinskog učenja

Postoje brojni besplatni i komercijalni sistemi mašinskog učenja, npr. za učenje stabala odlučuvanja, produkcionih pravila, neuronskih mreža i nosećih vektora.

Jedan od najpoznatijih sistema je CART (*Classification and Regression Trees*), koji je nastao 1984. godine, kao realizacija matematičke teorije koju su razvili statističari kalifornijskog univerziteta, Leo Breiman (*Berkli*), Jerome Friedman, Richard Olshen, i Charles Stone (*Stanford*).



Sistem za induktivno učenje stabala odlučivanja je komercijalizovala kompanija Salford Sytems [83].

Osnovna verzija ovog sistema se nalazi kao jedan od metoda u sistemu za istraživanje podataka Weka (metod *Simple Cart*) [70].

Delovi komercijalnih sistema poslovne inteligencije (BI)

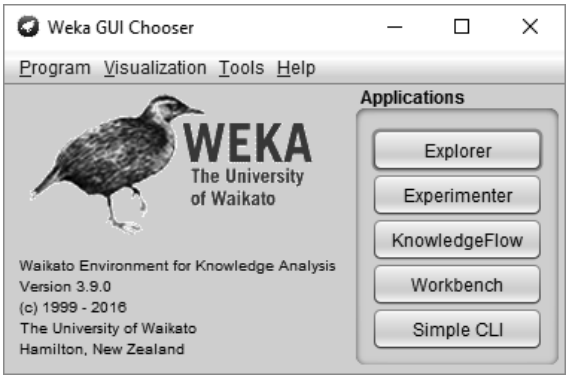
Komercijalni sistemi poslovne inteligencije koriste različite metode mašinskog učenja stabala odlučivanja (*decision trees*), pravila (*rules*), veštačkih neuronskih mreža (*artificial neural networks*, ANN), nosećih vektora (*support vector machines*, SVM) i sl.

Postoje sistemi poslovne inteligencije otvorenog koda, kao što su:
- Pentaho (sadrži sistem Weka),
- KNIME,
- Rapid Miner.

Komercijalni alati poslovne inteligencije su obično deo softverskih okruženja poznatih proizvođača za projektovanje i upravljanje bazama podataka, npr.
- Oracle BI 11g,
- IBM BI (DB2),
- Microsoft BI (SQL Server 2005/2008, *SharePoint*, *Office*).

Ilustracija: Pentaho Business Intelligence

Pentaho se zasniva na Java platformi i sistemu za istraživanje podataka otvorenog koda *Weka* [27], [48], [70]. Omogućava integraciju podataka iz različitih izvora, kreiranje izveštaja, vizualizaciju, upravljačke table (*dashboard*) i primenu metoda istraživanja podataka iz sistema Weka.



Sistemske funkcije su organizovane u tri podsistema: podsistem *Explorer*, za istraživanje i vizualizaciju podataka, kreiranje i preliminarnu evaluaciju modela, podsistem *Experimenter*, kreiranje obimnih eksperimenata i statistička evaluacija rezultata i podsistema *KnowledgeFlow*, koji ima grafički interfejs za kreiranje modela analize.

Najobimniji je podsistem za istraživanje podataka *Explorer*, koji raspolaže desetinama metoda za pripremu ulaznih podataka (pretprocesiranje) i više od stotine algoritama za istraživanje podataka metodima mašinskog učenja. Između ostalog, sistem raspolaže velikim brojem metoda učenja klasifikacija (*classification*), koji obuhvataju učenje stabala i pravila odlučivanja, veštačkih neuronskih mreža nosećih vektora (SVM), kao i metodima učenja stabala i pravila regresije (*regresion*) i metodima klasteringa (*clustering*).

Podsistem *Knowledge Flow* obezbeđuje grafički način programiranja složenih analitičkih postupaka.

Algoritmi mašinskog učenja/istraživanja znanja u sistemu SQL Server 2016

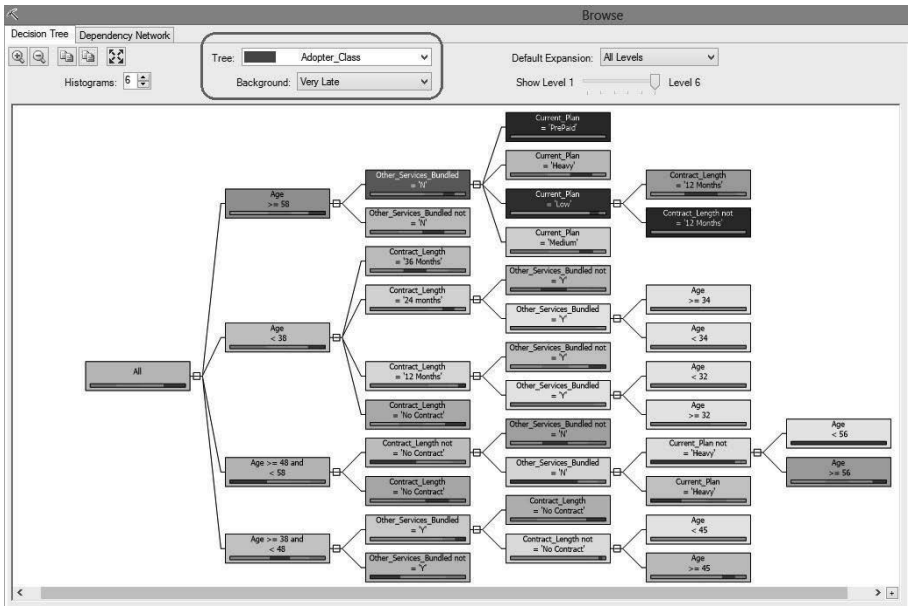
Sistem za upravljanje bazama podataka Microsoft *SQL Server* ima više osnovnih metoda mašinskog učenja, odnosno istaživanja podataka, kao deo sistema poslovne inteligencije, Tabela 3.

Vidi se da sistem nudi sve najčešće korišćene metode istraživanja podataka za učenje klasifikacija, pravila regresije i analizu asocijacija.

Tabela 3 - Osnovni metodi mašinskog učenja sistema SQL Server BI

Algoritam	Opis
Association Rules	Identifikacija značajnih relacija između primera
Clustering	Identifikacija grupa podataka sličnih karakteristika (*clusters*)
Decision Trees	Otkrivanje relacija u podacima učenjem stabala odlučivanja na osnovu primera
Linear Regression	Otkrivanje relacija između atributa za predviđanje vrednosti numeričkog atributa
Logistic Regression	Otkrivanje relacija između atributa radi ocene verovatnoće neke vrednosti atributa
Naïve Bayes	Učenje probabilističkih pravila klasifikacije
Neural Nework	Otkrivanje relacija u podacima učenjem veštačkih neuronskih mreža
Sequence Clustering	Otkrivanje grupisanja podataka (*clusters*) na osnovu sekvenci događaja
Time Series	Analiza i predviđanje vremenski zavisnih podataka za dugoročna (*long-term*) i kratkoročna (*short-term*) predviđanja metodima ARIMA i ARTXP (*Microsoft Research*), kao i optimizacija tačnosti predviđanja

Sistem za učenje stabala odlučivanja u sistemu Microsoft *SQL Server* generiše po jedno stablo odlučivanja za svaki atribut koji se koristi za predviđanje [49]. Primer grafičkog prikaza stabla odlučivanja je na Sl. 90.



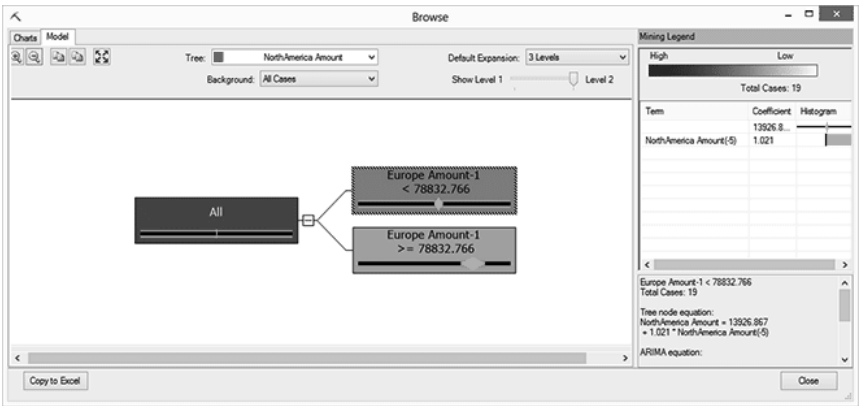
Sl. 90. Stablo odlučivanja u sistemu SQL Server BI

Dodatak za istraživanje podataka Microsoft Office 2007/2010 (Data Mining Add-In)

Podržava kompletan proces istraživanja podataka u sistemu Excel [50], tako što obezbeđuje funkcije za:

- *pripremu* podataka - istraživanje, čišćenje i i priprema podataka za istraživanje znanja;
- *modeliranje* podataka - izgradnja uzoraka i trendova na osnovu podataka radi predviđanja.
- testiranje i validaciju naučenih modela;
- upravljanje modelima - pregled, modifikacija i upravljanje postojećim modelima istraživanja znanja;
- dokumentovanje procesa istraživanja podataka - beleženja aktivnosti u obliku naredbi DMX (*Data Mining Extensions*) ili jezika ASSL (*Analysis Services Scripting Language*).

Na Sl. 91. je primer grafičkog prikaza prediktivnog modela mu obliku regresionog stabla.



Sl. 91. Stablo regresije u dodatku Excel Data Mining

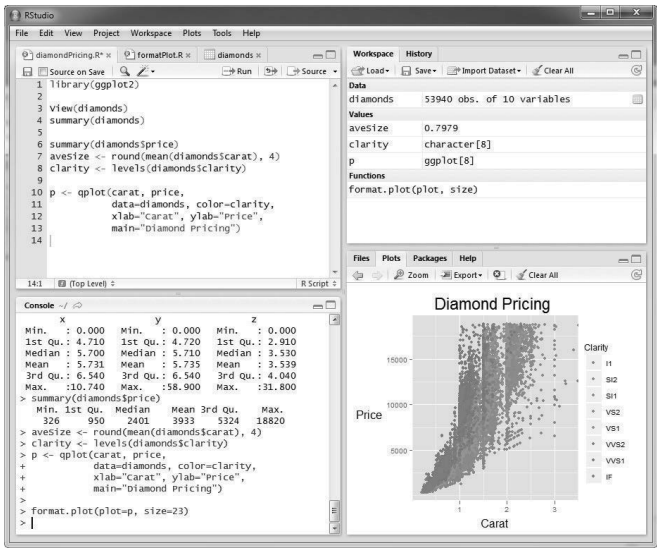
Objektne programske biblioteke i jezici

Programske biblioteke i dodaci omogućavaju upotrebu ispitanih metoda mašinskog učenja za izgradnu novih sistema, između ostalog sistema za podršku odlučivanju. Poznate programske biblioteke su za jezike R i Python, odnosno sisteme Matlab i Octave:

- *Sistem otvorenog koda R*. Osnova sistema je interpretativni programski jezik *R*, namenjen jednostavnom radu matricama i manipulisanju standardnim formatima podataka, kao i veoma bogat grafički prikaz rezultata. Predstavlja veoma popularan statistički softver i *de facto* standard

analize i istraživanja podataka. Zajednica korisnika je obezbedila preko 9.100 blioteka/paketa programa za statističke analize, istraživanje podataka i mašinsko učenje, koji se mogu preuzeti na Web sajtu CRAN (*Comprehensive R Archive Network*) [69]. Više o primeni sistema R u istraživanju podataka može se naći u Prilogu E.

Na Sl. 92 je *RStudio*, jedan od više grafički orijentisanih interfejsa razvijenih za ovaj sistem.

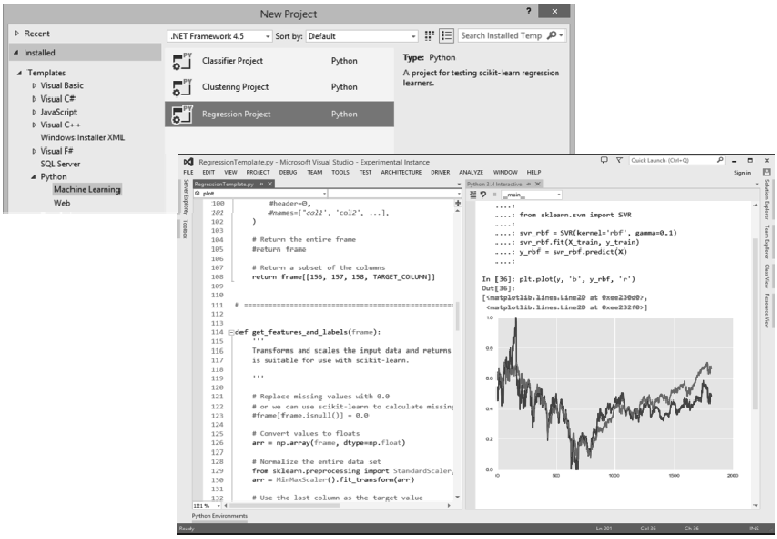


Sl. 92. Grafički orijentisani interfejs R Studio

- *Sistemi otvorenog koda zasnovani na jeziku Python* - osnova je interpretativni programski jezik *Python* opšte namene, koji omogućava izradu kompletnih Web aplikacija.

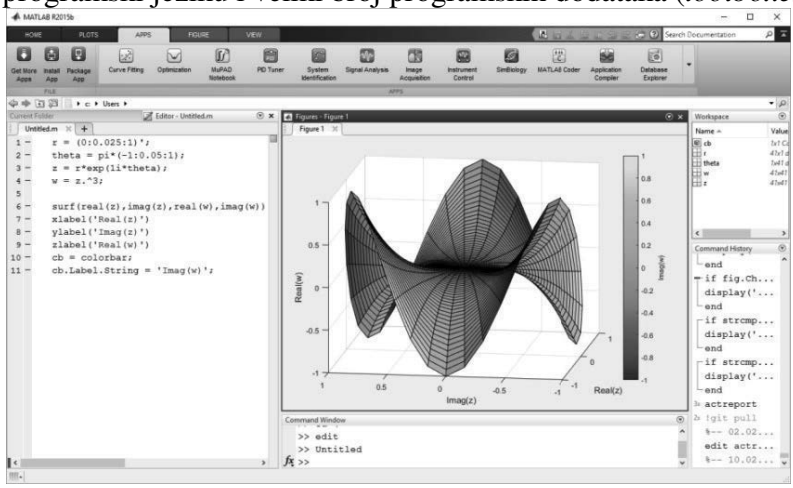
  Razvijen je veliki broj biblioteka otvorenog koda za mašinsko učenje i istraživanje podataka za ovaj jezik, kao što su npr. *scikit-learn* (najpoznatija, klasifikacija, regresija, klastering i drugi metodi istraživanja i analize podataka) [93], *Tensorflow* (veštačke neuronske mreže), *Theano* (neuronske mreže/*deep learning*), *Pylearn2* (nadgradnja biblioteke Theano) i *Pattern* (metodi obrade prirodnog jezika, klasifikacije i klasteringa), koji se mogu peruzeti s repozitorijuma *Github*.

  Postoji više programskih editora i integrisanih razvojnih okruženja za jezik Python. Neki od besplatnih alata su *IDLE*, Eclipse/*PyDev* i Microsoft Visual Studio (Community Edition)/*Python Tools for Visual Studio* (PTVS), Sl. 93.



Sl. 93. Microsoft Visual Studio dodatak Python Tools for Visual Studio

- *Komercijalni sistem Matlab* je alat za numeričke primene, rad s matricama i grafički prikaz rezultata, Sl. 94, koji ima sopstveni interpretativni programski jeziku i veliki broj programskih dodataka (*toolboxes*/libraries).



Sl. 94. Grafički orijentisani interfejs sistema Matlab

- Sistem *GNU Octave* zasniva se na sličnom programskom jeziku kao i sistem *Matlab*, tako da je moguća razmena određenih programa između platformi, ali je softverska podrška komercijalnog sistema *Matlab* bolja.

Web alati i mrežno računarstvo

Alati za mašinsko učenje na masivnim mrežnim serverima (*cloud-based machine learning tools*) koriste se u analizi podataka, radi dodavanja novih mogućnosti različitim Web aplikacijama.

- Google *Prediction* API omogućava programerima pristup naprednim algoritmima mašinskog učenja kompanije Google, pomoću kojih mogu da analiziraju empirijske podatke velikog obima i daju predviđanja ishoda koji su od interesa.

  Primeri primene su različiti inteligentni sistemi - automatizovani *spam* i bezbednosnu filteri, sistemi preporučivanja zasnovani na demografiji klijenata ili sistemi koji pronalaze skrivene obrasce u finansijskim podacima.

- Apache *Mahout* je programska biblioteka otvorenog koda, koja se u praksi najviše koristi za implementaciju sistema za preporučivanje (*recommender engines*), ali poseduje metode učenja klasifikacija i grupisanja (*clustering*). Osnovna namena sistema je obrada velikih skupova podataka, koji su preveliki za obradu na jednom računarskom sistemu.

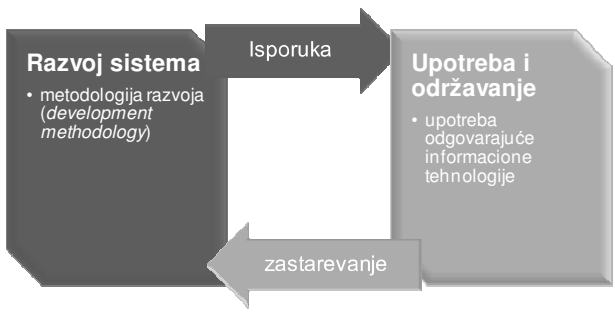
  Sistemi za preporučivanje su najbrže prihvaćena primena mašinskog učenja u praksi. Danas veliki broj sajtova e-trgovine svoju ponudu zasniva na prepoznavanju ponašanja klijenata u prošlosti i pogađanju njegovog ukusa i interesovanja, koji se vrši na osnovu automatizovanog naučenih pravila.

11.6 Menadžment razvoja sistema za podršku odlučivanju

Sistem za podršku odlučivanju je softverski proizvod. Metodologijom efikasnog razvoja pouzdanog softvera bavi se disciplina softverskog inženjerstva.

11.6.1 Podsetnik: projekt razvoja softvera

U opštem slučaju, životni ciklus svakog sistema (*system life cycle*) može se podeliti u dve faze, fazu razvoja i fazu upotrebe i održavanja, Sl. 95.



Sl. 95. Životni ciklus softverskog sistema

Proces razvoja softverskog sistema (*system development process*) je skup aktivnosti, metoda, prakse i automatizovanih alata koje učesnici u razvoju sistema koriste za razvoj i održavanje sistema i softvera.

Klasične faze razvoja sistema su [43]:
1. Definisanje zahteva (*Requirements*)
2. Analiza (*Analysis*)
3. Specifikacija (*Specification*)
4. Projektovanje (*Design*)
5. Implementacija (*Implementation*)
6. Testiranje (*Testing*)
7. Isporuka/uvođenje (*Deployment*)
8. Održavanje (*Maintenance*)

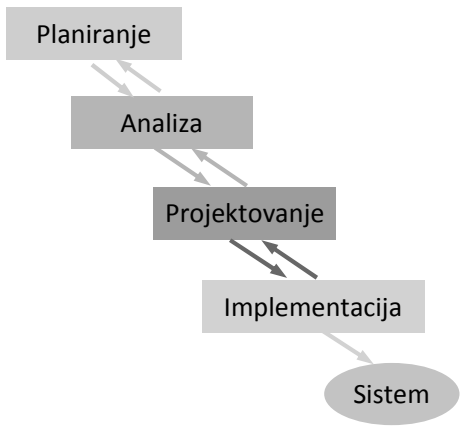
Metodologije razvoja softvera

Metodologija razvoja softverskih sistema (*system development methodology*) je standardizovani proces razvoja, koji definiše skup aktivnosti, metoda, iskustava, preporuka i automatizovanih alata koji se koriste za razvoj i neprekidno usavršavanje softverskog sistema.

Primer klasične metodologije razvoja je strukturna sistem analiza (*Structural System Analysis and Design*, SSAD), koja implementira sekvencijalni pristup razvoju, tzv. metod vodopada. Novije kategorije metodologija su spiralna, iterativna i inkrementalna, odnosno kombinovane metodologije.

Metodologije razvoja savremenih sistema su objektno orijentisame, npr. metodologija *Rational Unified Process* (RUP), koja je iterativno-inkrementalna. Često se traži da metodologije budu manje formalne i efikasne, kakve su tzv. agilne metodologije, kao što su ekstremno programiranje (*Extreme Programming*, XP) i SCRUM [43].

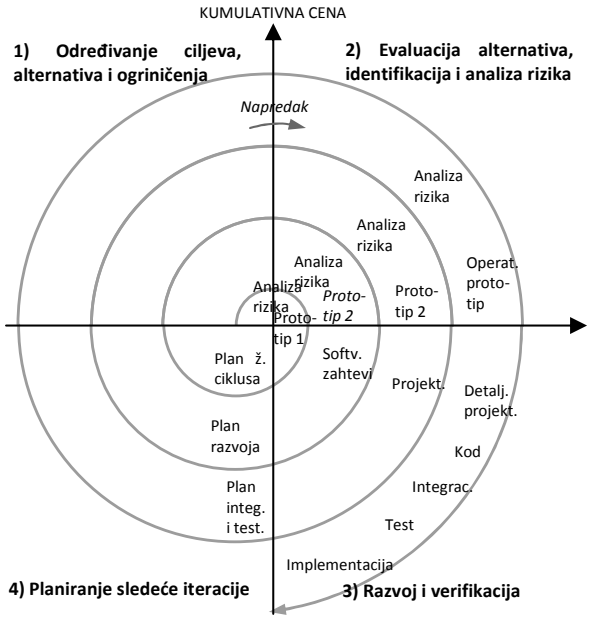
Sekvencijalni razvoj ili model vodopada, Sl. 96, nije adekvatan za razvoj složenih sistema jer produžava trajanje razvoja, ne može da prati sve veći obim sve složenijih aplikacija i neprekidne tehnološke inovacije, a uz to nije pogodan za velike ili distribuirane razvojne timove.



Sl. 96. Metodologija vodopada

Osnovni problem ovakvog pristupa što nije moguća identifikacija i smanjenje rizika u ranim fazama projekta razvoja. Primer je metodologija strukturne sistem analize i projektovanja. Prednost ove stroge, sekvencijalna metodologije je da daje rezultat visokog kvaliteta. Osnovni nedostatak je veoma dug razvoj i kasno otkrivanje problema.

Novije metodologije imaju za cilj smanjenje rizika. *Spiralna* metodologija (Boehm,1988) predviđa ciklično ponavljanje pojedinih faza, tako da se ciklično ponavlja i analizu rizika, Sl. 97 [43].



Sl. 97. Spiralna metodologija razvoja softvera

Iterativna metodologija predviđa da se životni ciklus softvera sastoji od nekoliko sekvencijalnih iteracija, koje se preklapaju u vremenu. Razlika između dve sukcesivne iteracije je inkrement. *Inkrementalna* metodologija je ona koja softver i razvija i isporučuje u inkrementima.

U praksi se često koriste kombinovane metodologije, gde se ukupna organizacija i sprovođenje pojedinih faza vrše na različite načine.

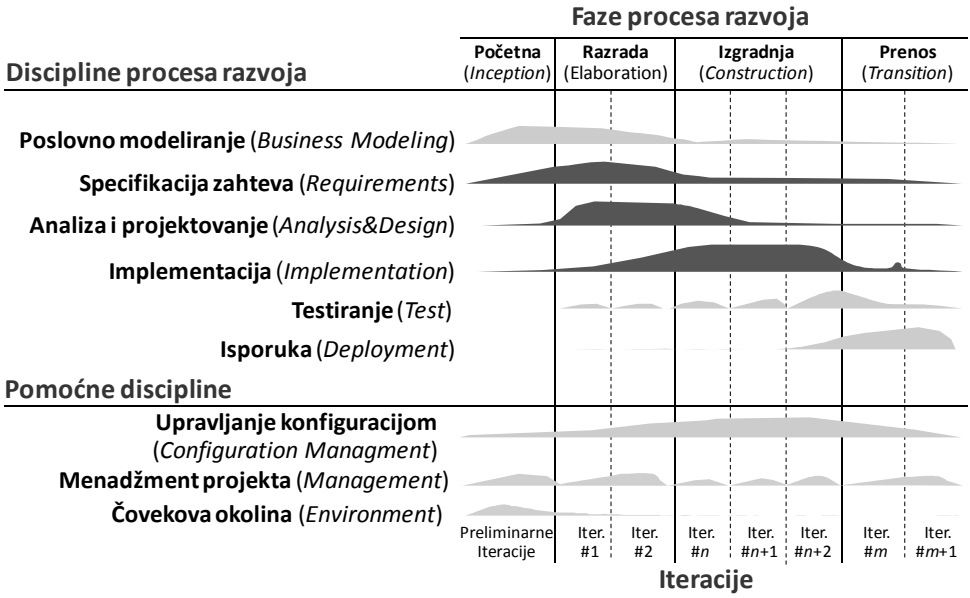
Objektno orijentisane metodologije, kao što je proces razvoja objektno orijentisanih sistema ili jedinstveni proces (*Unified Process*, UP) predviđaju iterativni razvoj kompleksnih sistema, koji se sastoji od niza ciklusa, od kojih svaki rezultuje izvršnom verzijom sistema (kodom) koji se može testirati.

Metodologija Rational Unified Process (RUP)

Metodologija *Rational Unified Process* (*RUP*) je iterativno-inkrementalna metodologija, koja ima sopstvene licencirane razvojne alate. Razvijena je kao

konkretizacija jedinstvenog procesa razvoja objektno orijentisanih softverskih sistema (*Unifed Process*, UP). Cilj jedinstvenog procesa je omogućavanje izrade visokokvalitetnog softvera, koji zadovoljava korisničke zahteve za predvidivo vreme i budžet. Za kompleksne sisteme, objedinjeni proces predviđa iterativni razvoj, koji se sastoji od niza ciklusa, od kojih svaki rezultuje izvršnom verzijom sistema, Sl. 98.

Sastoji se od četiri osnovne vremenske faze: početna faza (*inception*) gde se definiše obima projekta; faza razrade (*elaboration*), gde se vrši planiranje, izrada specifikacija i definiše osnovna arhitektura sistema; faza izgradnje sistema (*construction*) i faza prenosa sistema krajnjim korisnicima (*transition*).



Sl. 98. RUP metodologija procesa razvoja objektno orijentisanog softvera

Svaka faza i iteracija nosi određeni rizik i vremenski završava u nekoj kontrolnoj tački. Pregled projekta u kontrolnoj ili presečnoj tački ocenjuje stepen zadovoljenja korisničkih zahteva i potrebu za eventualnim prestrukturiranjem projekta. Svaka iteracija proizvodi novu verziju, stabilni izvršni podskup konačnog sistema.

Metodologija predviđa paralelno odvijanje faza razvoja, koje naziva disciplinama. U svakoj iteraciji se predviđa vremenski promenljiv rad na većini disciplina. Prve iteracije se pretežno bave zahtevima i projektovanjem, a naredne u sve manjoj meri, Sl. 98.

Agilne metodologije

Agilne metodologije su manje stroge metodologije razvoja softvera, pogodne za manje projekte i timove, koje ne opterećuju birokratijom, već predstavljaju razuman kompromis. Predviđaju adaptivno planiranje, evolutivni razvoj i inkrementalnu isporuka softvera. Primeri ovih metodologija su ekstremno programiranje (*Extreme Programming*, XP) i SCRUM.

Metodologija XP ne koristi precizno planiranje, realizuje se u tesnoj saradnji s korisnikom kroz male ili srednje razvojne timove. Predviđa intenzivno programiranje i testiranje, često u parovima.

Iterativna metodologija SCRUM dobila je naziv po ragbi formaciji od osam igrača, koji se kao jedinstvena celina (grupa) pomeraju po terenu. Predviđa dnevne sastanke i mesečne iteracije. Obuhvata upravljanje projektnim zahtevima, formiranje iteracija (planiranje sprinta), kontrolu implementacije i isporuke klijentu.

Microsoft Solutions Framework

Okvirna metodologija *Microsoft Solutions Framework* (MSF) obuhvata principe, modele, discipline, koncepte i preporuke za isporuku informatičkih rešenja kompanije Microsoft. Predstavlja samo okvir (*framework*) i dopušta izbor konkretne klasične ili agilne metodologije razvoja. Daje samo opšte principe visokog nivoa i koncentriše se na rezultate razvoja [92].

11.6.2Faze razvoja sistema za podršku odlučivanju

Proces razvoja sistema za podršku odlučivanju, koji se još naziva sistem za podršku menadžmentu (*Management Support System*, MSS) predviđa prilagođenu primenu neke od poznatih metodologija razvoja softvera, odnosno informacionih sistema.

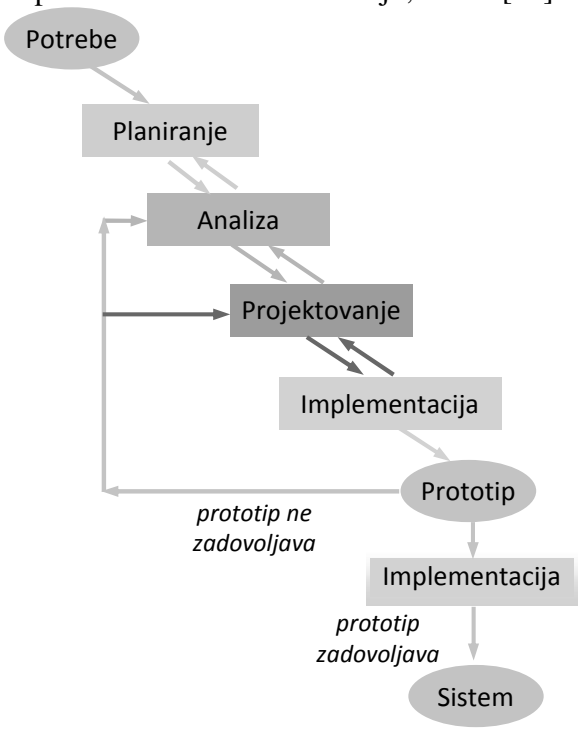
Osnovni koraci razvoja su [1], [10]:

1. Planiranje, identifikacija i obrazlaganje MSS,
2. Kreiranje arhitekture sistema,
3. Izbor organizacije razvoja (samostalno, narudžba, nabavka, iznajmljivanje, partnerski sistem),
4. Instalacija, testiranje i isporuka sistema,
5. Funkcionisanje, održavanje, izmene.

Prototipski razvoj

Sistem za podršku odlučivanju se razvija za delimično strukturirane i nestrukturirane probleme. Na početku novog projekta menadžeri i programeri obično ne razumeju problem odlučivanja u potpunosti, ne mogu da procene obim problema, vrstu modela ili tehnologiju koju treba primeniti, kao ni potrebe

za informacijama. Zbog toga se sistem razvija metodologijom prototipskog razvoja, izradom niza prototipova. Pristup se još naziva iterativni dizajn, evolutivni razvoj, adaptivni ili inkrementalni dizajn, Sl. 88 [43].



Sl. 99. Proces prototipskog razvoja softvera

Proces iterativnog prototipskog razvoja se tipično realizuje kao na Sl. 89 i obuhvata:

1. Prikupljanje projektnih zahteva, kojima se definišu se opšti ciljevi i identifikuju područja koja zahtevaju više istraživanja;
2. Brzo projektovanje, koje se usmerava na ono što će biti vidljivo korisniku: ulaze i izlaze sistema;
3. Izradu prototipa, zajedničke radne verzije sistema, za koju se koriste postojeći programski segmenti i generatori programa;
4. Ocenjivanje prototipa i preciziranje zahteva.

Prednosti prototipskog razvoja su:

- kratko vreme razvoja,
- kratko vreme reakcije korisnika,
- poboljšano razumevanje sistema od strane korisnika,
- niska cena.

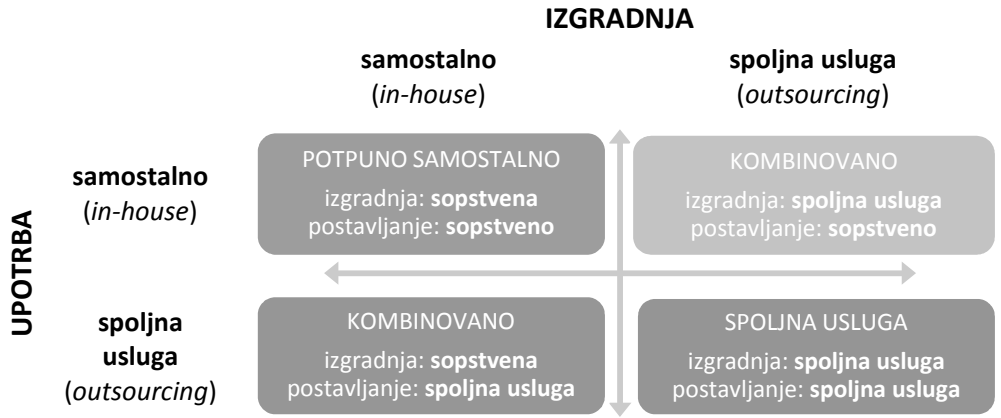
Nedostaci prototipskog razvoja se javljaju kad se u projektu postavljau dodatni zahtev, kao što su:

- detaljna analiza isplativosti informacionog sistema,
- detaljan opis informacionih potreba,
- dizajn sistema jednostavnog za održavanje,
- sveobuhvatno testiranje,
- dobra priprema korisnika.

Različiti pristupi organizaciji razvoja i upotrebi sistema

Razvoj sistema se može u celini prepustiti nekoj softverskoj firmi (*outsourcing*, koji uključuje i analizu i projektovanje sistema) ili realizovati u okviru kompanije (*in-house*, sopstvenim snagama). Izmeštanje razvoja znači angažovanje spoljne firme za usluge razvoja i izrade sajta, koje ne mogu da realizuju sopstveni kadrovi, Sl. 100.

Sopstveni razvoj zahteva stručni informatički kadar različitih specijalnosti (projektanti, programeri, Web dizajneri i menadžeri), računarsku opremu i softverske alate. Softverski alati mogu biti osnovni (Microsoft *Visual Studio*) ili posebni alati za izgradnju sistema za podršku odlučivanju, npr. ljuske ekspertnih sistema.



Sl. 100. Varijante organizacije izgradnje i funkcionisanja sistema

Prednosti sopstvenog razvoja su izrada sajta u skladu sa sopstvenim potrebama i usvajanje novih tehnologija, što kasnije omogućava brže prilagođavanje promenama poslovnog okruženja.

Nedostatak sopstvenog razvoja su rizici vezani za realizaciju složenih funkcija sistema. Sopstvena rešenja mogu da ponavljaju korake razvoja koji su drugi već završili, tako da se zbog učenja u toku razvoja sistema odlaže početak njegove upotrebe.

Prednost nabavke skupog namenskog rešenja je korišćenje najsavremenijeg i dobro ispitanog softvera, kao i brža realizacija projekta razvoja, iako ne treba zanemariti problem izbora takvog sistema, koji zahteva određeno vreme za analizu i ocenu većeg broja različitih softverskih rešenja.

Pristup putem prilagođavanja programskih rešenja (*customization*) takođe se može vršiti sopstvenim snagama ili angažovanjem spoljnih saradnika, s tim da svaka izmena načelno eksponencijalno povećava ukupne troškove.

Instalacija i upotreba sistema se takođe može realizovati na sopstvenom sistemu ili sistemu nekog provajdera. Spoljašnja usluga obezbeđuje načelno nižu cenu i visoke performanse, dok sopstvena konfiguracija načelno omogućava veću kontrolu i izgradnju sopstvenog sistema bezbednosti i zaštite podataka.

Odluka o izboru načina razvoja i upotrebe sistema se donosi na osnovu velikog broja kriterijuma, npr.

- *Fleksibilnost* - prilagodljivost specifičnim zahtevima;
- *Informacioni zahtevi* - informacione potrebe korisnika;
- *Pogodnost za upotrebu* - veoma važna za poslovne korisnike;
- *Hardverski i softverski resursi* - kompatibilnost s postojećim resursima;
- *Instalacija* - složenost i trajanje instalacije;
- *Održavanje* - različita ažuriranja, ko treba da vrši i kada;
- *Ugled proizvođača* - reputacija, finansijski podaci;
- *Procena troškova* - troškovi razvoja ili usvajanja, kao i cena instalacije, integracije, prilagođavanja i održavanja;
- *Kadrovi* - kadrovski zahtevi za svaku fazu; spoljni konsultanti su skupi;
- *Tehnološki razvoj* - planiranje prilagođavanja promenama brže od konkurencije;
- *Skalabilnost* - prlagodljiost rasta sistema, po broju korisnika, transakcija itd. (*horizontalna* / *vertikalna*);
- *Dimenzionisanje* - estimacija potrebnih resursa je vrlo složena, koriste se merenja i simulacije (*benchmark*);
- *Performanse* - definisanje vremena odziva (*latency*) ili trajanja operacije i propusnost (*throughput*) ili broj operacija za zadano vreme;
- *Pouzdanost* - rad bez otkaza/zastoja (redundansa);
- *Bezbednost* - fizička i proceduralna bezbednost sistema, komunikacija i podataka.

11.6.3Kriterijumi za ocenjivanje sistema za podršku odlučivanju

Evaluaciju projekata razvoja sistema za podršku odlučivanju vrše tehnički i poslovni menadžeri, s osnovnim ciljem da ustanove njihovu uspešnost [52].

Klasične tehnike evaluacije projekata su:

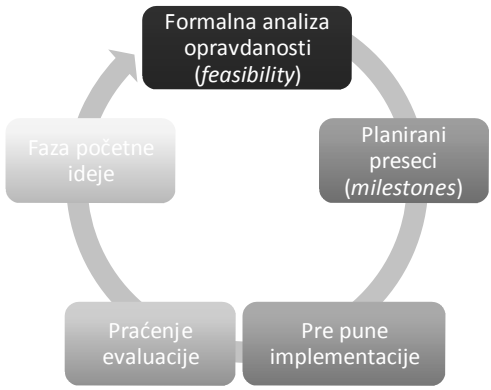
- povrat investicije (*Return on Investment*, ROI),
- sadašnja neto vrednost (*Net Present Value*, NPV) i
- diskontovani tok novca (*Discontinued cash flow*).

Sve one su namenjene racionalizaciji ulaganja. Kod projekata razvoja sistema za podršku odlučivanju nije lako kvantifikovati rezultate i dobitke, pa se koriste procene i pretpostavke. Ne postoje univerzalno prihvaćene mere vrednosti sistema za podršku odlučivanju (DSS) [41]. Tradicionalna analiza isplativosti (*cost-benefit*) obično nije primerena za DSS. Dobitak korišćenja DSS je obično *kvalitativan*: mogućnost razmatranja većeg broja alternativa, kreiranje ideja, poboljšan timski rad, učenje i povećano poverenje u donesene odluke.

U opštem slučaju je teško kvantifikovati ekonomsku vrednost sistema informacionih tehnologija (IT), što je važno priliko donošenja odluke o investicijama u IT [41], [43]. Evaluaciju projekta razvoja sistema za podršku vrši tehnički ili poslovni menadžer. Tipična pitanja su, npr.

- Kakav je povrat investicije u predloženom DSS projektu?
- Koji je rok otplate?
- Koji su očekivani troškovi?
- Koje su očekivane koristi?
- Šta možemo da učinimo sa novim sistemom ako ne može da radi zajedno sa postojećim informacionim sistemima ?
- Da li konkurenti imaju skladište podataka, OLAP ili EIS?

U ciklusu razvoja, evaluacija projekta DSS se vrši neprekidno, Sl. 101.



Sl. 101. Proces evaluacije projekata DSS

Aktivnosti evaluacije treba da budu srazmerne veličini, složenosti i ceni razmatranog projekta DSS. Sponzori i menadžeri projekta donose odluku koji obim i tip evaluacije je adekvatan i neophodan. Projekt se može revidirati ili otkazati u bilo kojoj fazi.

Jedan od mogućih načina da se proceni DSS je po skupu kriterijuma [53] koje koriste ljudi na različitim organizacionim nivoima: menadžeri (npr. korisnici), kreatori (npr. razvojni kadar) i teoretičari (npr. naučnici koji se bave teorijskim pitanjima kreiranja DSS) [41].

Menadžerski skup kriterijuma uspešnosti sistema za podršku proverava da li su:

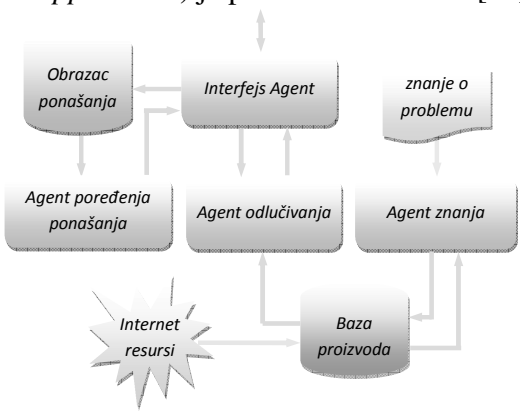
1. Adekvatni za problem odlučivanja,
2. Pružaju podršku menadžerima na svim nivoima,
3. Podržavaju odlučivanje kod međusobno zavisnih odluka,
4. Podržavaju sve faze procesa donošenja odluka (npr. Sajmon, 1960),
5. Nezavisni od određenog procesa donošenja odluka,
6. Jednostavni za korišćenje.

Kriterijumi su *deskriptivni* i teški za kvantifikovanje.

Model evluacije sistema za podršku odlučivanju

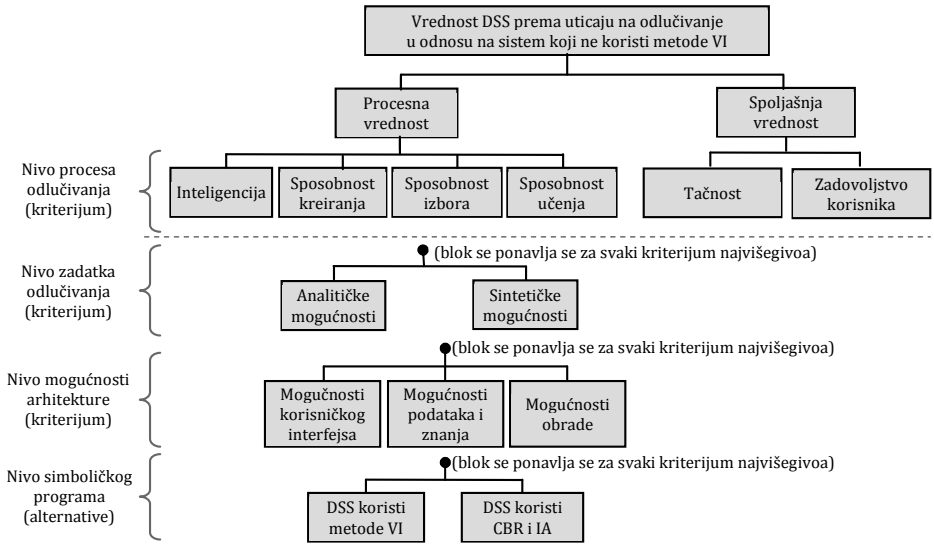
Koristi od sistema za podršku odlučivanju mogu biti npr. samo *efektivnost* ili *efikasnost*, ali i oba istovremeno. Efektivnost se bavi izlazima - rezultatima i meri stepen ostvarivanja postavljeih ciljeva, a efikasnost meri upotrebu ulaza i resursa koji se koriste za ostvarivanje ciljeva (npr. vreme, ljudski kapital, računarska moć).

Evaluacija DSS koristi *višekriterijumski model* koji omogućava poređenje alternativa hijerarhijskim strukturiranjem kriterijuma. Vrednost DSS na osnovu uticaja na odlučivanje (*decision value of DSS*) povezan je preko procesa i ishoda sve do nivoa računarskog programa. Primer tipičnog sistema za preporučivanje (*recommender system application*) je prikazan na Sl. 102 [41].



Sl. 102. Analizirani model sistema za preporučivanje

Na Sl. 103. prikazan je primer hijerarhijskog višekriterijumskog modela evaluacije sistema za preporučivanje sa Sl. 102 [41].



Sl. 103. Višekriterijumski model evaluacije sistema za preporučivanje

Literatura

1. Turban E., Aronson J.E., Liang T., Sharda R., *Decision Support and Business Intelligence Systems*, 9th Ed, Pearson Education,Inc, 2010
2. Sauter V.L., *Decision Support Systems for Business Intelligence*, 2ndEd, John Wiley and Sons, 2010
3. Parmigiani G., Inoue L., *Decision Theory: Principles and Approaches*, John Wiley & Sons, Ltd., 2009
4. Russell S., Norvig P., *Artificial Intelligence - A Modern Approach*, 3rd Ed, Prentice-Hall, 2010
5. Webb J.N., *Game Theory: Decisions, Interaction and Evolution*, Springer – Verlag, London, 2007
6. Anderson D.R, Sweeney D. J., Williams T. A, Camm J. D., Martin K., *An Introduction to Management Science*, Quantitative Approach to Decision Making, 13th Ed , South-Western Cengage Learning, 2012
7. Hillier F.S., Lieberman G.J., *Introduction to Operations Research*, 9th Ed, McGraw-Hill, 2010
8. Ragsdale C.T., *Spreadsheet Modeling & Decision Analysis*, Sixth Edition, South-Western Cengage Learning, 2012
9. Albright S. C., Winston W. L., Zappe C.J., *Data Analysis and Decision Making*, 4th Ed, South-Western, Cengage Learning, 2011
10. Vercellis C., *Business Intelligence: Data Mining and Optimization for Decision Making*, pp. 262-275, John Wiley & Sons Ltd, 2009
11. Seref M., Ahuja R.,Winston W., *Developing Spreadsheet-based Decision Support Systems*, Dynamic Ideas, 2007
12. Denardo E. V., *Linear Programming and Generalizations*, Springer, 2011
13. Cvetković D., Simić S., "Linearno programiranje", *Odabrana poglavlja iz diskretne matematike*, str. 59-77, Akademska misao, Beograd, 2004
14. Charniak E., McDermott D., *Introduction to Artificial Intelligence*, Reading: Addison-Wesley, 1985
15. Bouman R., van Dongen J., *Pentaho Solutions: Business Intelligence and DataWarehousing with Pentaho and MySQL*, Wiley Publishing, Inc., 2009
16. Hansson S. O., *Decision Theory: A Brief Introduction*, Royal Institute of Technology (KTH), Stockholm, 2005
    http://home.abe.kth.se/~soh/decisiontheory.pdf

17. Laudon K.C. Traver C. G., *E-Commerce: Business, Technology Society*, 6th ed., Boston, Addison-Wesley, 2010

18. Negnevitsky M., *Artificial Intelligence, A guide to Intelligent Systems*, 2nd Edition, Addison Wesley, 2005

19. Giarratano J.C., Riley G. D., *Expert Systems: Principles and Programming*, 3rd Edition, Course Technology, 1998

20. Sasikumar M., Ramani S., Muthu Raman S., Anjaneyulu K. S. R., Chandrasekar R., *A Practical Introduction to Rule Based Expert Systems*, Narosa Publishing House, New Delhi, 2007 http://sigai.cdacmumbai.in/files/ESBook.pdf

21. Vanderbei R.J., *Linear Programming: Foundations and Extensions*, 3rd Edition, Springer, 2008

22. Willig A., "A Short Introduction to Queueing Theory", Technical University Berlin, Telecommunication Networks Group, 1999.

23. Sugumaran R.,John DeGroote J., *Spatial Decision Support Systems, Principles and Practices*,Taylor and Francis Group, LLC, 2011

24. Larose D. T., *Data Mining Methods and Models*, John Wiley & Sons, 2006

25. Cherkassky V., Mulier F. M., *Learning from Data: Concepts, Theory, and Methods*, 2nd Ed, John Wiley - IEEE Press, 2007

26. Witten I.H., Frank E., Hall. M.A., *Data Mining: Practical Machine Learning Tools and Techniques*, 3rdEd, Elsevier Inc, 2011

27. Bohanec, M., *DEXi: A Program for Multi-Attribute Decision Making*, http://www-ai.ijs.si/MarkoBohanec/dexi.html

28. Michalski, R.,Carbonell,J., Mitchell T. (Eds.), *Machine learning: An artifcial intelligence approach (Vol. I)*, San Francisco, CA: Morgan Kaufmann,1983

29. Hart, A.,"Machine induction as a form of knowledge acquisition in knowledge engineering", in Forsyth, R. (ed), *Machine Learning: Principles and techniques*, Chapman and Hall, London, 1989

30. Breiman L., Friedman J.H., Olshen R.A., Stone C.J., *Classification and Regresssion Trees*, Wadsworth, Belmont, 1984.

31. Quinlan J. R. *C4.5, Programs for Machine Learning*, Morgan Kaufmann Publishers, 1993

32. Clark P., Boswell R., "Rule Induction with CN2: Some Recent Improvements", in *Machine Learning - Proc. of the Fifth European Conference (EWSL-91)*, pp. 151-163, Berlin: Springer Verlag, 1991

33. Cohen W. W.: "Fast Effective Rule Induction", In: *Twelfth International Conference on Machine Learning*, 115-123, 1995

34. Chang Chih-Chung, Lin Chih-Jen (), "LIBSVM: A library for support vector machines", *ACM Transactions on Intelligent Systems and Technology*, 2 (3), 2011

35. Milosavljević M., Miškovic V., *Elektronska trgovina*, Univerzitet Singidunum, 2016

36. Kaschek R. H. (editor), *Intelligent assistant systems:concepts, techniques, and technologies*, Idea Group Inc., 2007

37. Miškovic V., "Inteligentni asistenti: programski sistemi koji uče od svojih korisnika", *Singidunum Revija*, Vol. 5, No. 2, pp. 89-97, 2008

38. Meyer T.A., Whateley B.., "Spambayes: Effective open-source, Bayesian based, email classification system", In *Proceedings of the First Conference on Email and Anti-Spam (CEAS)*, 2004

39. McCallum E., Weston S., *Parallel R*, O'Reilly Media, Inc, 2012

40. Miškovic V., "Cloud-based Data Mining Using System R and Revolution R", 14. *Naučni skup s međunarodnim učešćem SINERGIJA 2013*, 2013

41. Phillips-Wren G., Mora M., ForgionneG. , "Assessing Today: Determining the Decision Value of Decision Support Systems", in *Schuff D.,· ParadiceD., · Burstein F.,·Power D. J.,· Sharda R., Decision Support: An Examination of the DSS Discipline*, Springer, 2011

42. Ricci F., Rokach L., Shapira B., Kantor P. B. (eds), *Recommender Systems Handbook: A Complete Guide for Research Scientists and Practitioners*, Springer Science+Business Media, 2011

43. Sommerville I., *Software Engineering*, Prentice Hall, 2011

44. Quintero A, Konaré D, Pierre S, Prototyping an intelligent decision support system for improving urban infrastructures, European Journal of Operational Research, 162 (3), p.654-672, 2005

45. Kohavi R., "A Study of Cross-validation and Bootstrap for Accuracy Estimation and Model Selection", in *Proc. of International Joint Conference on Artificial Intelligence*, 1995.

46. Pal S., Shiu S., *Foundations of soft case-based reasoning*, John Wiley & Sons, 2004

47. Friedman-Hill E., *Jess in Action: Java Rule-Based Systems*, Manning Publications, 2003

48. Roldan M. C., *Pentaho 3.2 Data Integration: Beginner's Guide*, Packt Publishing, 2010

49. Veerman E., Moss J. M., Knight B., Hackney J., *Microsoft SQL Server 2008 Integration Services: Problem-Design-Solution*, Wiley Publishing, 2010

50. Winston W. L., *Microsoft Office Excel 2007: Data Analysis and Business Modeling*, Microsoft Press, 2007

51. Meritt D., *Building Expert Systems in Prolog*, Springer-Verlag, 2000

52. Power D. J., "Evaluating Decision Support Systems Projects" in Power D. J., *Decision Support Systems Hyperbook, Cedar Falls, IA: DSSResources.COM*, 2000 (Ch 12)

53. Kerzner H., *Project Management Metrics, KPIs, and Dashboards - A Guide to Measuring and Monitoring Project Performance*, John Wiley & Sons, 2011

54. Bouyssou D. at all, *Evaluation and decision models: a critical perspective*, Kluwer Academic Publishers, 2000

55. Garland J., Anthony R., *Large-Scale Software Architecture*, John Waley & Sons, 2003

56. Miškovic V, Milosavljević M. M. , "Application of Hybrid Symbolic Ensembles to GeneExpression Analysis", in *Proceedings of 9th Symposium on Neural Network Applications in Electrical Engineering*, p.95-98, Belgrade, September 2008

57. Miškovic V, Milosavljević M. M., "Application of Symbolic Inductive Learning Methods to Gene Expression Analyses", in *Proceedings of 9th Symposium on Neural Network Applications in Electrical Engineering*, p.99-102, Belgrade, September 2008

58. Vapnik V., *Statistical Learning Theory*, John Wiley&Sons, 1998

59. Shawe-Teylor J., Cristianini N., *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004

60. Miškovic V., Adamović S., "Direktna primena metoda nosećih vektora na učenje pravila odlučivanja u berzanskom trgovanju", *Proceedings of 57th ETRAN Conference*, VI2.5.1-5, 2013

61. Bratko I., *Prolog Programming for Artificial Intelligence*, 4th Ed, Pearson Education, 2011

62. R. S. Michalski, I. Bratko, M. Kubat, *Machine Learning and Data Mining: Methods and Applications*, Jon Wiley & Sons, 1998

63. Kaklauskas A., *Biometric and Intelligent Decision Making Support*, Springer, 2015

Web reference

64. www.wikipedia.org, 2013
65. www.expertise2go.com, 2013
66. Facilitate Pro http://www.facilitate.com/
67. Annual Survey http://www.RexerAnalytics.com
68. Neuro Solutions www.neurosolutions.com/ , 2013
69. http://www.jango.com , 2013
70. http://www.alice.pandorabots.com
71. Microstrategy www.microstrategy.com, 2013
72. The Comprehensive R Archive Network http://cran.r-project.org/
73. Weka http://www.cs.waikato.ac.nz/ml/weka/
74. ACAFE - A Computerized Advancement For Emergency http://www.ostechnology.com.au/acafe-clinical-decision-support, 2013
75. www.ibm.com
76. http://www.amazon.com, 2012
77. Daden Limited http://www.daden-dev.co.uk/beta/chatbot/abi_3d.html
78. www.palisade.com , 2013
79. Simul8 www.simul8.com, 2013
80. www.dssresources.com, 2012
81. www.leadfusion.com/products/financialcalculators/Auto, 2012
82. GIS Internet Guide http://www.gis.com/ , 2012
83. http://dssresources.com/decisionaids/dsseval.html
84. Rapid Miner www.rapidminer.com/
85. SAS Institute www.sas.com
86. Salford Systems www.salford-systems.com/cart.php
87. http://www.chatbots.org/
88. Humeau P., Jung M., *Benchmark of e-Commerce solutions*, NBS System, 2013 https://www.nbs-system.com/wp-content/uploads/benchmark_short_english.pdf

89. http://sabr.org/sabermetrics/single-page
90. SSIS Tutorial http://mindmajix.com/ssis-tutorial
91. Download SQL Server Data Tools (SSDT) https://msdn.microsoft.com/en-us/library/mt204009.aspx
92. https://en.wikipedia.org/wiki/Microsoft_Solutions_Framework
93. scikit/learn http://scikit-learn.org/
94. Rosetta Code https://www.rosettacode.org/wiki/Factorial
95. Teorija Dempstera i Šafera https://en.wikipedia.org/wiki/Dempster–Shafer_theory
96. Nauka o podacima (Data Science) https://en.wikipedia.org/wiki/Data_science

Prilozi

A. Sistem za istraživanje podataka Weka
B. Okruženje za razvoj ekspertnih sistema Expertise2Go
C. Programsko okruženje R i Microsoft R Server
D. Microsoft Excel Solver
E. Sistem za podršku višekriterijumskom odlučivanju DEXi
F. Sistem za istraživanje podataka Empiric

A. Sistem za otkrivanje znanja Weka

Sistem za istraživanje podataka WEKA (*Waikato Environment for Knowledge Analysis*) je moderan sistem za istraživanje podataka otvorenog koda [1], [2]. Izrađen je u okviru projekata razvoja novih metoda istraživanja znanja koje je finansirala vlada Novog Zelanda (Univerzitet Waikato, Hamilton, Novi Zeland).

Razvijen je u jeziku Java i može se dalje proširivati preko dodataka (*plug-ins*). Predstavlja osnovu sistema poslovne inteligencije otvorenog koda *Pentaho*. Veoma je rasprostranjen, pošto se koristi u više od milion instalacija. Sistemi za istraživanje podataka kao *KNIME*, *RapidMiner*, *Matlab* i *R* imaju interfejse za korišćenje metoda iz sistema Weka.

Verzije sistema za različite operativne sistema, sa i bez Java virtuelne mašine, može se preuzeti sa http://www.cs.waikato.ac.nz/ml/weka/downloading.html.

Instalacija je jednostavna, pokrene se instalacioni program i prihvate ponuđene opcije. Kreira se prečica (*short-cut)* u obliku ptice, a na sistemskom folderu se još nalaze primeri podataka i obiman priručnik za upotrebu sistema.

Osnovne funkcije

Pokretanjem sistema, dobije se dijalog na kome se biraju grupe funkcija sistema, koje su raspoređene u tri podsistema:

- *Explorer* za istraživanje i vizualizacija podataka, kreiranje i preliminarna evaluacija modela,
- *Experimenter* za kreiranje obimnih eksperimenata i statistička evaluacija rezultata,
- *KnowledgeFlow* obezbeđuje grafički interfejs za vizuelno kreiranje modela.



U ovoj knjizi koriste se samo funkcije podsistema za istraživanje i vizualizaciju podataka *Explorer*. Osnovne funkcije ovog podsistema su:

- Priprema ulaznih podataka (69 metoda pretprocesiranja);
- Istraživanje metodima mašinskog učenja (više od 100 algoritama klasifikacije/regresije, desetak algoritama za klastering, dvadesetak metoda selekcije atributa i 6 algoritama analize asocijacija);
- Statistička evaluacija naučenih modela,

- Vizualizacija ulaznih podataka i rezultata.

Pristup ovim funkcijama se ostvaruje preko odgovarajućih panela:

- *Preprocess* - pristup podacima iz različitih izvora (fajlovi, SQL baze podataka, Internet);
- *Classify* - metodi mašinskog učenja (klasifikacija/regresija) i evaluacija podataka i rezultata;
- *Cluster* - učenje bez učitelja (clustering), estimacija i vizualizacija rezultata;
- *Associate* - učenje asocijativnih pravila;
- *Select attributes* - ocena korisnosti i selekcija grupa atributa;
- *Visualize* - prilagodljivi grafički prikaz ulaznih podataka.



Priprema podataka

Panel *Preprocess* služi za pristup podacima (*Open file*/*URL*/*DB*) i selekciju podataka (*Choose*). Standardni ulazni format fajla ima sufiks .arff (*Attribute-Relation File Format*) i sastoji se od zaglavlja i podataka:

- u zaglavlju je spisak atributa i njihovog tipa (numeric, real), odnosno za nenumeričke nabrajanjem skupa dozvoljenih vrednosti {...}; ako postoji klasa, obično je zadana kao poslednji atribut
- iza oznake @data slede podaci, odnosno primeri zadani kao vektori vrednosti atributa.

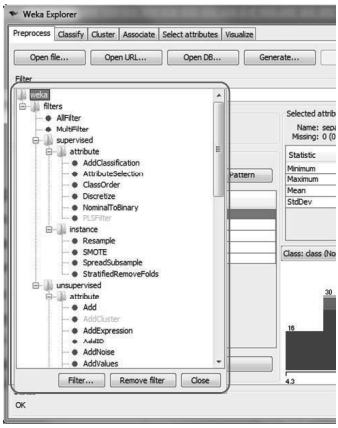
Primer sadržaja arff fajla:

```
@relation audit
@attribute ID numeric
@attribute Age numeric
@attribute Employment {Consultant, PSFederal, PSLocal, ...}
. . .
@attribute Adjusted {0, 1}
@data
1004641,38,Private,College,Separated,Service,71511.95,Female,0,...
1010229,35,Private,Associate,Unmarried,Transport,73603.28,Male,...
. . .
```

Ostali formati su CSV i interni formati samostalnih programa učenja, kao što su *C4.5*, *LibSVM* i *SVMLight*.

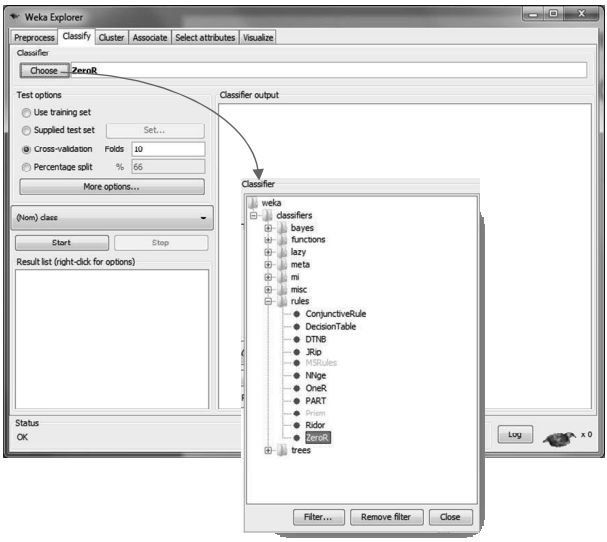
Pregled učitanih podataka daje uvid u statistiku svih atributa i histograme raspodele njihovih vrednosti. Postoji i funkcija za editovanje podataka.

Selekcija podataka omogućava softversko filtriranje redova i kolona koji će se koristiti za istraživanje, s velikim brojem različitih filtera:



Učenje klasifikacija

Učenje klasifikacija vrši se funkcijama panela *Classify* podsistema *Explorer*:



Metodi učenja klasifikacija biraju se pomoću tastera *Choose*. Svaki metod ima skup parametara, koji su simbolički označeni u prikazanoj komandnoj liniji. Dijalog za promenu parametara otvara se klikom na komandnu liniju.

Metod estimacije tačnosti učenja određuje se izborom opcije u *Test options*. Podrazumeva se krosvalidacija, ali se može izabrati metod procentualne podele obučavajućeg skupa ili zadati poseban testni skup tasterom *Set*.

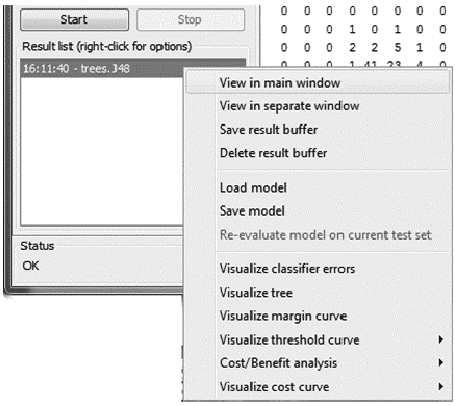
Potrebno je još proveriti izbor klasifikacionog atributa, jer sistem uvek nudi poslednji atribut iz liste.

Proces učenja i estimacije tačnosti pokreće se tasterom *Start*.

Rezultat učenja se dobija kao izveštaj u (novom) prozoru rezultata. Osim tekstualnog opisa problema i tekstualne reprezentacije naučenog modela (pravila, stabla, neuronske mreže i sl.), prikazuju se i rezultati estimacije tačnosti.

Naučeni modeli (stabla odlučivanja, pravila i sl.) mogu se sačuvati na na fajl sa sufiksom .model i kasnije koristiti za klasifikovanje drugih primera.

Operacija čuvanja realizuje se desnim klikom na naučeni model u prozoru *Results list* i izborom opcije *Save model* :



Za neke modele, kao što su stabla odlučivanja, mogu se dobiti grafički prikazi desnim klikom na *Result list* i izborom opcije *Visualize tree*.

Učenje pravila regresije

Pravila regresije se takođe uče metodima iz panela *Classify*. Pre izbora metoda učenja, izabere se *ciljni atribut*, koji mora biti numerički, nakon čega se u meniju *Choose* pojavljuju samo metodi učenja pravila regresije, npr. metod M5Rules.

Učenje bez učitelja i analiza asocijacija

Metodi učenja bez učitellja (*clustering*) mogu se naći u panelu *Cluster* , a metodi analize asocijacija u panelu Associate. U ovom materijalu se ne razmatraju.

[1] Weka http://www.cs.waikato.ac.nz/ml/weka/

[2] Witten I.H., Frank E., Hall. M.A., *Data Mining: Practical machine Learning Tools and Techniques*, 3rdEd, Elsevier Inc, 2011

B. Okruženje za razvoj ekspertnih sistema Expertise2Go

Ljuska ekspertnog sistema *Expertise2Go* je komercijalni alat za razvoj ekspertnih sistema, koji ima besplatnu verziju, koja se koristi u ovom materijalu.

1. Instalacija i pokretanje

Ljuska ekspertnog sistema e2g se može preuzeti sa www.expertise2go.com/. U gornjem levom uglu osnovne stranice, izabrati *Mini-Course: create expert systems with the FREE e2gRuleEngine/e2gDroid shells and e2gRuleWriter decision table software*. Izabrati Module 2, pronaći link iza tabele i preuzeti arhivu **e2g3g.zip**.

Softver ljuske ekspertnog sistema sastoji se od tri Java programa i jedne Android aplikacije:

- *e2gRuleEngine.jar* - mehanizam zaključivanja (*inference engine*),
- *e2gRuleWriter.jar* - interaktivni grafički orijentisan editor pravila.
- *e2gRuleInduction* - program za induktivno učenje metodom ID3, koji može da induktivno generiše kompletnu bazu znanja,
- *e2gDroid* - verzija ljuske ekspertnog sistema za mobilne uređaje (Android).

Instalacija nije potrebna, ali je neophodno da je na sistemu instalirana Java virtelna mašina (v.6 ili novija). Eventualno se još može i kreirati skraćenica na radnoj površini sistema za pokretanje editora pravila e2gRuleWriter.jar (desni klik, pa *Create Shortcut*). Mehanizam zaključivanja nema sopstveni korisnički interfejs i pokreće se iz iz komandne linije, odnosno .bat fajla. *Moguće je pokretanje kao Java apleta, iz HTML fajla, ali se neće koristiti, jer nije u skladu s HTML 5 standardom.*

2. Izgradnja baze znanja ekspertnog sistema

Baza znanja koju koristi ljuska ekspertnog sistema predstavlja standardni tekstualni fajl, za koji je uobičajeno da ima sufiks **.kb** (od *knowledge base*). Može se kreirati programski ili manuelno, na dva načina: pomoću nekog editora teksta i korišćenjem namenskog editora pravila ljuske ekspertnog sistema *e2gRuleWriter*.

(A) UNOS BAZE ZNANJA POMOĆU EDITORA TEKSTA

Uz pomoć običnog editora teksta mogu se uneti produkciona pravila i svi ostali elementi baze znanja: cilj zaključivanja (atribut čiju vrednost treba ustanoviti), specifikacija elemenata interfejsa za svaki atribut koji se korsti u uslovima pravila pojedinačno i eventualne dopunske deklaracije.

(B) UNOS BAZE ZNANJA POMOĆU GRAFIČKI ORIJENTISANOG EDITORA PRAVILA

Grafički orijentisani editor pravila *e2gRuleWriter* pokreće se iz komandne linije. Baza znanja se kreira u formi *tabele odlučivanja*, koja predstavlja jedan od načina specifikacije programske logike.

Pravila (RULES) unose se kao vertikalne kolone tabele. Uslovi i zaključci svih pravila čine *condition stub* (CONDITIONS) i *action stub* (ACTIONS).

	Rule1	Rule2	Rule3
Condition1	val_{11}	val_{21}	-
Condition2	val_{12}	-	val_{32}
Action1	act_1	-	act_1
Action2	-	act_2	-

Na slici su primeri dve različite varijante tabele za isti problem odlučivanja, u kojima se koristi (a) samo skup binarnih vrednosti atributa Y/N i (b) prošireni skup vrednosti, koji koristi numeričke i nenumeričke atribute:

	Pravilo 1	Pravilo 2	Pravilo 3
padavine	*Y*	*Y*	-
očekivana temperatura > 0	*Y*	-	-
očekivana temperatura < 1	-	*Y*	-
obucite mantil	*Y*	-	-
obujte čizme	-	*Y*	-
obucite šta želite	-	-	*Y*

(a) skup binarnih vrednosti

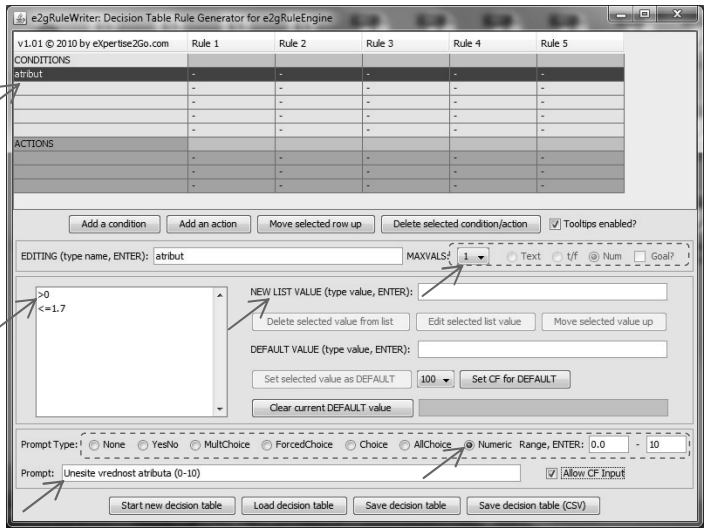
	Pravilo 1	Pravilo 2	Pravilo 3
padavine	*očekuju se*	*očekuju se*	-
očekivana temperatura	*> 0*	*< 1*	-
odluka	*obucite mantil*	*obujte čizme*	*obucite šta želite*

(b) prošireni skup vrednosti

Način predstavljanja pravila bira inženjer znanja. Pravila se unose tako da se prvo unesu svi atributi koji se koriste u uslovima pravila u podkolonu CONDITIONS, a atributi iz zaključaka pravila u podkolonu ACTIONS. Nnakon toga se u kolone pojedinačnih pravila za svaki par atribut-pravilo bira vrednost, koja se koristi u uslovu, odnosno zaključku pravila.

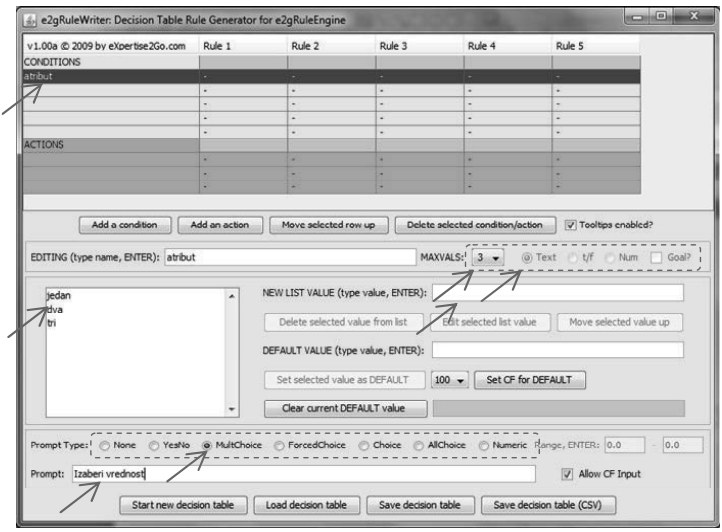
Unos uslova pravila se vrši dodavanjem atributa u kolonu CONDITIONS, ako već nije upotrebljen u nekom drugom pravilu, nakončega se unosi specifikacija domena vrednosti atributa, načina unosa podataka i vrste dijaloga, koja zavisi od tipa atributa (izbor ili unos vrednosti):

(a) Za *nenumeričke* atribute se pomoću polja NEW LIST VALUE unosi maksimalni broj mogućih vrednosti (MAXVALS) i skup dozvoljenih vrednosti atributa, npr. {jedan, dva, tri}.



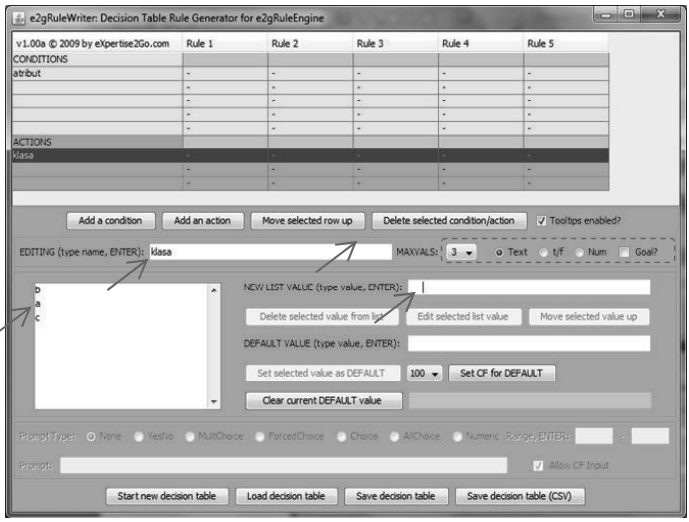
U donjem delu se zadaje oblik dijaloga i tekst pitanja (*Prompt*:) kojim se od korisnika traži unos vrednosti atributa. Unos teksta pitanja treba obavezno potvrditi tasterom Enter. Dijalog oblika *MultChoice* znači da korisnik bira jednu od ponuđenih vrednosti, u ovom primeru tri.

(b) Za *numeričke* atribute izabere se tip (Num), a zatim u polje NEW LIST VALUE unose vrednosti i relacioni operatori koji će se koristiti u uslovima pravila.

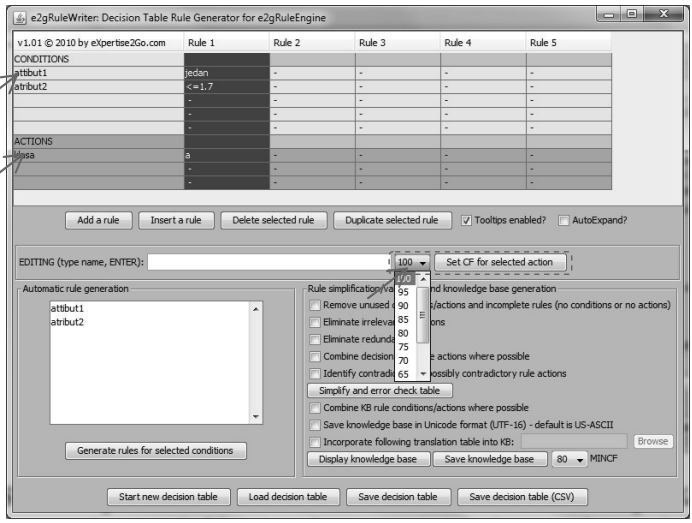


Nakon toga bira se oblik dijaloga (Numeric) i minimalna i maksimalna dozvoljena vrednost numeričkog atributa (Range, ENTER). Ako su obe vrednosti 0.0, smatra se da nema ograničenja. Pitanje korisniku unosi se uz obavezni Enter na kraju, a u tekst je poželjno dodati i informaciju o ograničenjima (od-do).

(c) Unos zaključaka pravila vrši se dodavanjem svakog od atributa upotrebljenih u zaključcima pravila u kolonu ACTIONS, na isti način kao i unos nenumeričkih atributa u uslovima pravila.



(d) Sama pravila (*rules*) unose se izborom neke od ponuđenih vrednosti za nenumeričke, odnosno vrednosti i relacionog operatora za numeričke atribute.Vrednosti onih atributa koji se ne koriste u nekom pravilu izostavljaju se.



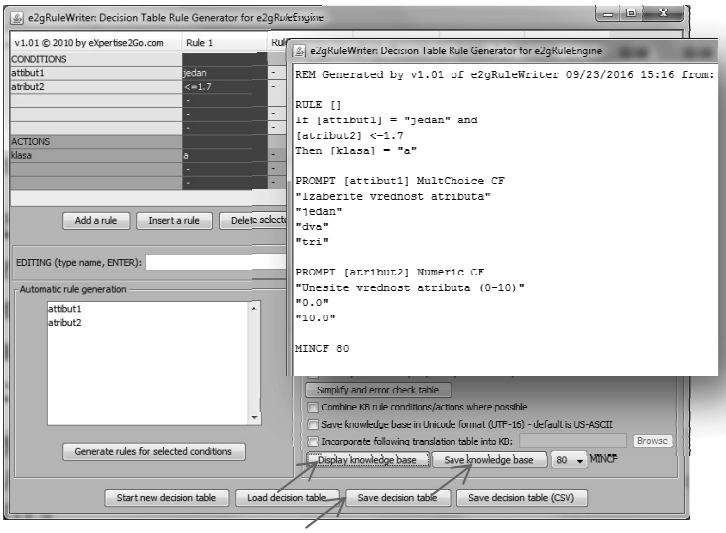
Za svako pravilo se može definisati faktor pouzdanosti pravila CF, u inkrementima od 5% (podrazumevajuća vrednost je 100%) i najniža dozvoljena vrednost pouzdanosti pravila (MINCF, podrazumeva se 80%).

(B) GENERISANJE BAZE ZNANJA POMOĆU PROGRAMA ZA INDUKTIVNO UČENJE

Baza znanja se može automatizovano generisati korišćenjem priloženog programa za induktivno učenje pravila na osnovu podataka *e2gRuleInduction*, koji je ograničen na probleme koji koriste isključivo nenumeričke atribute.

3. Pamćenje baze znanja

Tabela odlučivanja se može sačuvati na fajl sa nastavkom **.kbt** pomoću tastera *Save decision table* (sufiks je obavezni deo naziva fajla). Ovaj fajl je u internom formatu tabele odlučivanja i koristi se za nastavak rada ili buduće izmene baze znanja (taster *Load decision table*). Izgled baze znanja ekspertnog sistema se može pregledati pomoću tastera *Display knowledge base*, koja se pojavljuje nakon klika na neko pravilo.



Konačni oblik baze znanja čuva se pomoću komandnog tastera *Save knowledge base* na fajl sa sufiksom **.kb** (sufiks je obavezni deo naziva fajla u *Save* dijalogu). Ovaj fajl je deo ekspertnog sistema realizovanog uz pomoć ljuske ekspertnog sistema Expertise2Go.

4. Pokretanje ekspertnog sistema

Sistem se može pokrenuti kao samostalna Java aplikacija komandom:

```
java -jar e2gRuleEngine.jar -kb file:nazivbazeznanja.kb [argumenti]
```

Neki od argumenata u pozivu ljuske ekspertnog sistema su:

-w, -h - širina i dužina ekranskog prozora aplikacije u pikselima,
-bg -tc - boje pozadine i teksta, zadana kao naziv ili Web boje (#rrggbb),
-at -st - naziv aplikacije (*application title*) i namena sistema (*subtitle*),
-sb - tekst startnog tastera (*start button*).

5. Primer ekspertnog sistema

Kao primer će se izgraditi mali oftalmološki ekspertni sistem koji predlaže vrstu kontaktnih sočiva na osnovu podataka o dioptriji i stanju oka pacijenta. Upotrebiće se baza znanja s neegzaktnim pravilima, koja je kreirana za problem "kontaktna sočiva" (*contact-lenses*):

```
REM Baza znanja Kontaktna sočiva (kontaktna-sociva.kb)

RULE [1]
If [tear-prod-rate] = "smanjeno"
Then [kontaktna-sociva]="ne preporucuju se" @ 100

RULE [2]
If [astigmatism] = "ne"
Then [kontaktna-sociva] = "meka sociva" @ 85.71

RULE [3]
If [spectacle-prescrip] = "kratkovidnost"
Then [kontaktna-sociva] = "tvrda sociva" @ 100

DEFAULT [kontaktna-sociva] = "ne preporucuju se" @ 75

GOAL [kontaktna-sociva]

MINCF 75

REM Interfejs mehanizma zaključivanja

PROMPT [tear-prod-rate] MultChoice CF
"Kako oko suzi?"
"smanjeno"
"normalno"

PROMPT [astigmatism] MultChoice CF
"Da li na oku postoji astigmatizam?"
"ne"
"da"

PROMPT [spectacle-prescrip] MultChoice CF
"Prepisane su naočare za ..."
"kratkovidost"
"dalekovidost"
```

Zapokretanje sistema potrebno je iz arhive sistema dearhivirati mehanizam zaključivanja *e2gRuleEngine.jar* i folder *lib*. Pokretanje sistema se tada vrši pomoću komandi, koje se mogu uneti u mali .bat fajl:

```
set CLASSPATH=.; .\lib\plugin.jar
java.exe -jar e2gRuleEngine.jar -kb file:kontaktna-sociva.kb
 -w=400 -h=350 -bg=#006699 -tc="#3399FF" -at "Ocni lekar"
 -st "Demonstracija ljuske e2g"
 -sb "Odredjivanje vrste kontaktnih sociva"
```

Nakon pokretanja malog eskpertnog sistema, dobija se pozdravni ekran.



Konsultacija se pokreće pripremljenim tasterom. Nakon toga, sistem koristi mehanizam zaključivanja unazad, postavlja pitanja da ustanovi vrednosti atributa iz uslova primenjenih pravila.



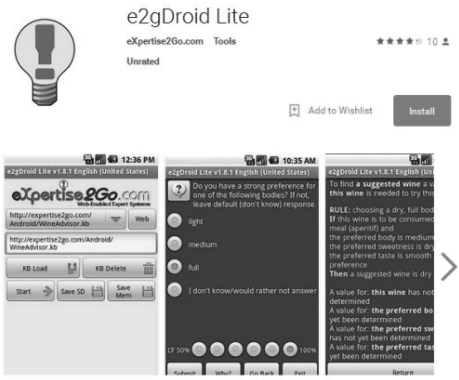
Na kraju sistem daje predlog odluke (dijagnozu), čije obrazloženje se može dobiti aktiviranjem tastera *Objasni*.

Prevod originalnog interfejsa može se izvršiti dodatnim direktivama TRANSLATE za prevod naziva tastera i tekstova sistemskih poruka, čiji tekst i oznake se mogu pronaći na sajtu *Expertise2Go*, npr.

```
TRANSLATE B_SUBMIT = "Potvrdi"
TRANSLATE B_EXPLAIN = "Objasni"
    ...
TRANSLATE TR_YES = "da"
TRANSLATE TR_NO = "ne"
TRANSLATE TR_NORESP = "Ne znam/radije ne bih odgovorio"
    ...
```

6. Pokretanje verzije za operativni sistem Android

Ista baza znanja može se koristiti i na mobilnim platformama (Android). Potrebno je samo na Android uređaj instalirati aplikaciju *e2gDroidLite* (besplatna verzija ljuske može se preuzeti na repozitorijumu Google *Play*).



Za testiranje se mogu koristiti i Windows emulatori, kao npr. *Droid4X* ili Web online emulatori, kao npr. *Manymo* www.manymo.com .

C. Programsko okruženje R i Microsoft R

Sistem *R* je programski jezik i okruženje za statistku, grafiku i istraživanje podataka. Nastao na univerzitetu u Oklandu, Novi Zeland, po ugledu na objektno orijentisani jezik *S*, koji je razvijen u Bel laboratorijama. Postoji novija verzija jezika S-PLUS u ponudi kompanija *Tibco*, dok je sistem *R* besplatan.

Prema anketama, programsko okruženje *R* je najčešće korišćen programski alat na području istraživanja podataka metodima mašinskog učenja. Sistem je besplatan i ima snažnu korisničku podršku, na osnovu koje je formirana velika bibiloteka od preko 9.100 programskih paketa, koja se neprekidno povećava.

Osim matematičkih i statističkih metoda, veliki broj paketa razvijen je za potrebe istraživanja podataka, npr. paketi: *caret* (učenje stabala odlučivanja), *randomForest* (učenje slučajnih šuma), *nnet* (učenje veštačkih neuronskih mreža), *e1071* i *ksvm* (učenje nosećih vektora).

Instalaciona verzija *R* se može preuzeti na referentnom sajtu *R* mreže [1].

Jezik *R* je interpretativni objektno orijentisani jezik, koji omogućava relativno jednostavno manipulisanje podacima, operacije s matricama, matematičke i statističke funkcije, kao i dobru grafičku podršku. Osnovni način upotrebe sistema je pomoću programskih naredbi, koje se mogu unositi u više grafički orijentisanih korisničkih interefejsa, kao što je *RStudio*. Sintaksa jezika je slična sintaksi jezika C.

Samo za ilustraciju, daje se nekoliko primera naredbi u jeziku *R*:

- Dodela vrednosti promenljivoj, aritmetički izraz i sadržaj promenljive

```
> x <- 5
> x <- 5*x^2+1
> x
[1] 126
```

- Osnovne operacije s matricama: kreiranje matrice 2x3 od dve matrice dobijene konkatenacijom skalarnih vrednosti, transponovanje i dodavanje skalarne vrednosti i prikaz sadržaja

```
> y <- rbind(c(1,2,3),c(4,5,6))
> y
     [,1] [,2] [,3]
[1,]    1    2    3
[2,]    4    5    6
> z <- t(y)+1
> z
     [,1] [,2]
[1,]    2    5
[2,]    3    6
[3,]    4    7
```

- Osnovne programske strukture: inicijalizacije matrice pomoću *for* petlje

```
> x <- as.matrix(10)
> for(i in 1:10) x[i]<- i*2
> x
 [1]  2  4  6  8 10 12 14 16 18 20
```

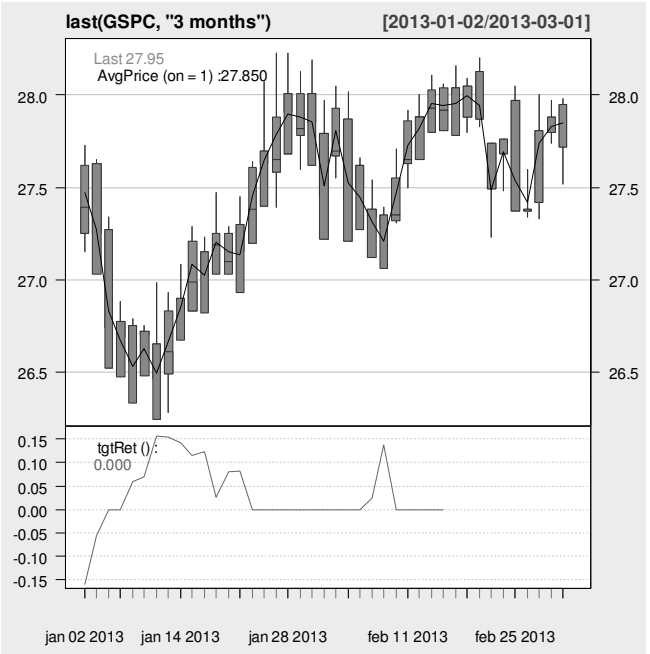
- Čitanje podataka prethodno sačuvanih u sistemu Excel u formatu CSV:

```
> library(xts)
> x <- as.xts(read.zoo('podaci.csv',header=T))
```

- Mašinsko učenje modela nosećih vektor (SVM) na osnovu podataka prethodno učitanih u matrice trainX (primeri) i trainY (oznake klasa)

```
> library(e1071)
> model <- svm(trainX,trainY,
           type="C-classification",kernel="radial",gamma=0.01)
> predv <- predict(model,testX)
```

Za grafički prikaz koriste se brojne funkcije za crtanje funkcija i dijagrama, kao i pripremljeni namenski modeli složenih grafikona, kao što je *candle chart*, grafikon za prikaz standardnih podataka o kretanju cena akcija na berzi.



Microsoft R (ranije Revolution R)

Postoji i komercijalna verzija Microsoft *R* (ranije *Revolution R* [2]) koja koristi integrisano razvojno okruženje Microsoft *Visual Studio* kao grafički orijentisani korisnički interfejs. Odgovarajući serverski softver omogućava istraživanje podataka veoma velikog obima korišćenjem paralelne obrade (*cloud computing*).

Sistem *Microsoft R* sastoji se od komponenti:

- Microsoft *R Open*
- Microsoft *R Client*
- Microsoft *R Server*
- SQL Server *R Services*

Namena i svojstva komponenti prikazani su u tabeli:

	Microsoft R Open	Microsoft R Client	Microsoft R Server
Obim podataka (*Big Data*)	Samo rad s podacima u memoriji	Samo rad s podacima u memoriji; veliki obim podataka povezivanjem na komponentu *R Server*	Rad s velikim obimom podataka, skalabilnost veličine diska
Brzina analize	Rad s više niti za matematičke funkcije	Rad s više niti za matematičke funkcije, do dve niti za *ScaleR* funkcije	Paralelni rad bez ograničenja
Vrsta podrške	Sami korisnici	Sami korisnici	Komercijalna podrška
Obim i kvalitet analitičkih metoda	Preko 9.000 programskih paketa otvorenog koda	Optimizacija R paketa otvorenog koda i *ScaleR* paketi pripremljeni za obradu velikog obima podataka (*Big Data-ready*), do dve niti	Optimizacija R paketa otvorenog koda, *ScaleR* paketi pripremljeni za obradu velikog obima podataka, veliki broj niti
Status	Softver otvorenog koda	Besplatan, za sve	Komercijalna licenca
Deploy Enterprise	Nije uključen	Nije uključen	Uključen

Kompanija Microsoft je podršku razvoju softvera u jeziku *R* obezbedila besplatnim dodatkom *R Tools for Visual Studio* za Visual Studio 2015, koji ima i besplatnu verziju Community Edition.

Microsoft R Open

Microsoft *R Open* je poboljšana verzija sistema *R*, ali je i dalje sistem otvorenog koda. Poptuno je kompatibilna sa svim standardnim *R* paketima i aplikacijama razvijenim za određenu verziju sistema *R*. Namenjen je za analizu ograničenog obima podataka, koji mogu stati u radnu memoriju računara.

Podrška korisnicima realizuje se putem posebnog portala MRAN (*Microsoft R Application Network*) https://mran.microsoft.com/.

Microsoft R Client

Besplatni softver za poslovnu analitiku i istraživanje podataka, nadogranja sistema Microsoft *R Open*, tako da može da koristi brojne besplatne metode (pakete). Između ostalog omogućava upotrebu svoje tehnologije *Scale R* za paralelizaciju aplikacija i korišćenje udaljenih servera, koja omogućava ograničen paralelni rad (do dve niti).

Microsoft R Server

Analitička platforma za obradu velikog obima podataka i isporuku rezultata zasnovana na jeziku *R*. Potpuno je kompatibilna s programima i paketima otvorenog koda na standardnom repozitorijumu CRAN. Omogućava razvoj i distribuciju *R* softvera visokih performansi na različitim platformama:

- *R Server for Hadoop* - pojednostavljeno korišćenje mogućnosti distribuirane obrade;
- *R Server for Teradata DB* - napredna analitika u okviru baze podataka;
- *R Server for Linux* - prediktivna analitika u operativnom sistemu Linux.

SQL Server R Services

Softver koji omogućava razvoj inteligentnih aplikacija korišćenjem jezika *R* i odgovarajućih softverskih biblioteka za rad s podacima u sistemu za upravljanje bazama podataka Microsoft *SQL Server*.

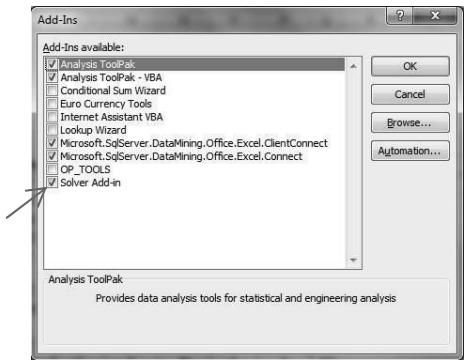
[1] The Comprehensive R Archive Network http://cran.r-project.org/.

[2] Revolution R Enterprise and RevoScaleR http://www.revolutionanalytics.com

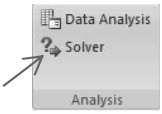
D. Microsoft Excel Solver

Microsoft Excel *Solver* je dodatak (*Add-in*) namenjen rešavanju optimizacionih problema linearnog, nelinearnog i celobrojnog programiranja. Sastoji se od više programa sa zajedničkim grafičkim korisničkim interfejsom.

Pre prve upotrebe je neophodno proveriti da li je aktiviran i po potrebi ga uključiti. Za Microsoft *Excel* 2007, aktivira se taster i izabere Excel options, opcija Add-Ins:



Solver se pokreće se iz menija sistema ili izborom u paleti alatki, npr. za Microsoft *Excel* 2007:



Sistem ima sopstveni interfejs za kreiranje i ažururanje modela, čija upotreba je u knjizi detaljnije opisana na nekoliko primera rešavanja problema linearnog programiranj.

Sličan dodatak postoji i za druge programe za rad s elektronskima tabelama, kao što je npr. Apache *OpenOffice Calc Solve.*

E. Sistem za podršku višekriterijumskom odlučivanju DEXi

Sistem DEXi je namenjen podršci višekriterijumskog odlučivanja korišćenjem simboličkog neparametarskog metoda. Simbolička ili kvalitativna reprezentacija elemenata problema odlučivanja najčešće odgovara ljudskom načinu razmišljanja.

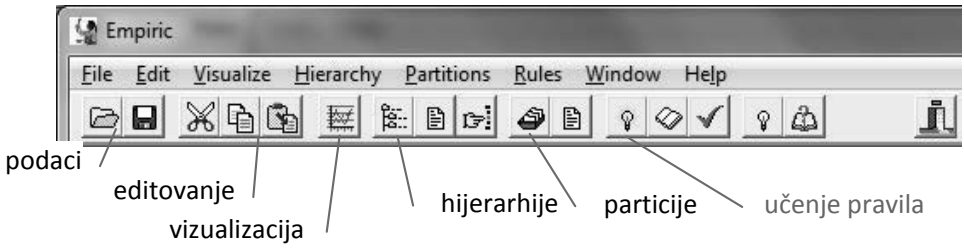
Sistem se može preuzeti na http://www-ai.ijs.si/MarkoBohanec/dexi.html.

Priručnik na engleskom jeziku za korišćenje poslednje verzije sistema DEXi može se preuzeti na http://kt.ijs.si/MarkoBohanec/pub/DEXiManual400.pdf.

F. Sistem za istraživanje podataka Empiric

Sistem za istraživanje podataka (*data mining*, *knowledge discovery*) *Empiric* je softverski alat za istraživanje podataka, odnosno otkrivanje skrivenih zakonitosti u podacima. Aktuelna verzija sistema nastala je kao deo izrade doktorske teze na Univerzitetu Singidunum, Beograd [1]. Implementiran je kao desktop aplikacija za operativni sistem Windows korišćenjem razvojnog alata Borland *Delphi*. Javna verzija sistema koristi podatke iz formatiranih tekstualnih datoteka i za njegovu upotrebu nije neophodna nikakva instalacija.

Sistem obezbeđuje osnovni skup alata za inteligentnu analizu podataka metodama induktivnog učenja. Osnovni meni sistema prikazan je na slici.

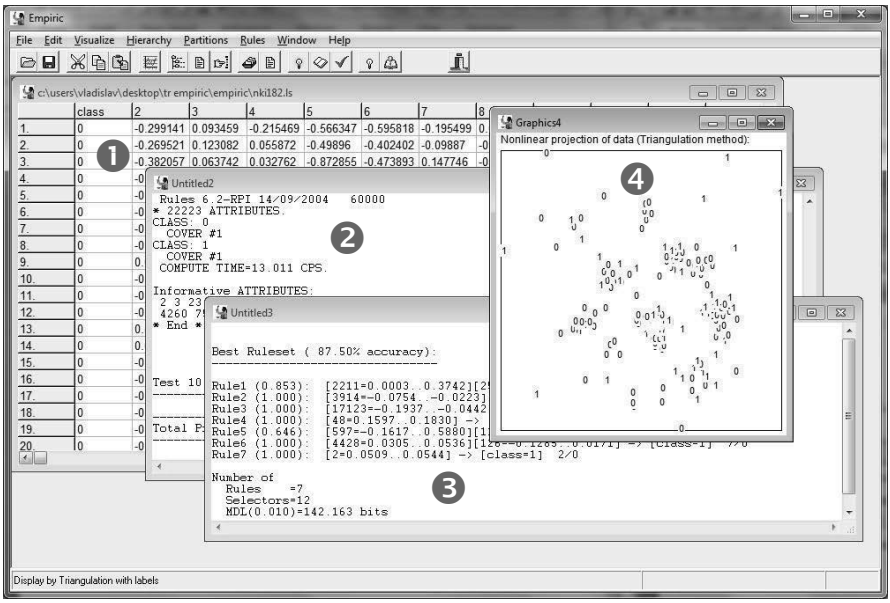


Sistem *Empiric* obezbeđuje osnovni skup alata za inteligentnu analizu podataka pomoću nekoliko pažljivo odabranih metoda induktivnog učenja. Sastoji se od podsistema:

1. *Editor primera*, za unos i ažuriranje modela problema i primera;
2. *Vizualizator*, koji strukturu obučavajućeg skupa primera prikazuje geometrijskim rasporedom tačaka na površini;
3. Podsistem za *induktivno učenje bez učitelja*, koji se sastoji od metoda učenja jednostavnih *(partitional clustering*) i hijerarhijski organizovanih klastera *(hierarhycal clustering*);
4. Podsistem za *induktivno učenje pravila* na osnovu primera (*induction of conjuctive rules*).

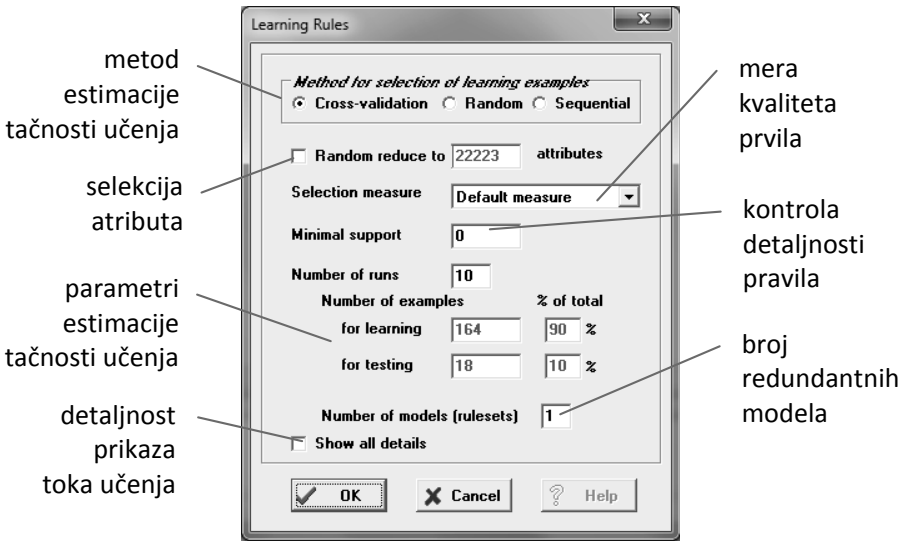
Svi podsistemi su prilagođeni jedinstvenom modelu predstavljanja znanja, tzv. atributnom modelu sa *kontinualnim* atributima i više tipova *diskretnih* atributa. Diskretni atributi su ugrađeni u tri varijante: *nominalni* (neuređen skup vrednosti), *linearni* (uređen skup vrednosti) i *strukturni* (parcijalno uređen skup vrednosti).

Na sledećoj slici prikazan je izgled ekrana sistema *Empiric* prilokom rešavanja problema učenja pravila dijagnostike na osnovu genskih ekspresija [2] (182 primera, predstavljenih vrednostima 22.224 atributa).



Obučavajući primeri su u tabeli ❶, dok su u ostalim prozorima prikazuje: tok izvršavanja algoritma induktivnog učenja pravila ❷ i rezultati učenja, odnosno pravila ❸. U grafičkom prozoru ❹ vidi se vizualizacija strukture obučavajućih primera, razvrstanih u dve klase, označene sa 0 i 1.

Unos podataka se vrši izborom iz menija File / Open ili aktiviranjem odgovarajuég tastera iz palete alatk, a funkcija induktivnog učenja pravila se izborom Rules|Learn ili aktiviranjem tastera iz palete alatki:



Naučena pravila se mogu pregledati iz menija Rules|View, odnosno tasterom iz palete altki. Logička pravila su predstavljena u obliku implikacije, a kao dodatna informacija se daje i ocena njihove pouzdanosti kroz tačnost na

obučavajućem skupu i broj pozitivnih primera iz kojih su izvedena. Na slici je primer naučenih pravila za problem učenja na osnovu genskih ekspresija:

```
Rule1 (0.853): [2211=0.0003..0.3742][2509=-0.3574..0.1696] -> [class=0] 81/14
Rule2 (1.000): [3914=-0.0754..-0.0223][172=-0.0729..0.1184] -> [class=0] 13/0
Rule3 (1.000): [17123=-0.1937..-0.0442][9=-0.0088..0.6085] -> [class=0] 8/0
Rule4 (1.000): [48=0.1597..0.1830] -> [class=0] 2/0
Rule5 (0.646): [597=-0.1617..0.5880][11151=<0.0293] -> [class=1] 53/29
Rule6 (1.000): [4428=0.0305..0.0536][126=-0.1285..0.0171] -> [class=1] 7/0
Rule7 (1.000): [2=0.0509..0.0544] -> [class=1] 2/0
```

U podsistem za induktivno učenje pravila sistema *Empiric* su ugrađene mere za ocenu kvaliteta pravila (hipoteze): informativnost atributa (*information gain*), relativna infomativnost atributa (*gain ratio*), entropija (*entropy*), Gini indeks (*gini*), logička pouzdanost (*logical sufficiency content*), mera kvaliteta Kaufman-Michalski (*Q-measure*), globalna mera Relief (*Relief*, *ReliefF*). Svaka od ovih mera se može upotrebiti za generisanje pojedinačnih ili višestrukih modela, odnosno jednostavnih ili višestrukih skupova pravila.

Priprema podataka problema induktivnog učenja vrši se izvan sistema, npr. u nekom editoru tekstualnih fajlova, npr. *Notepad*. Problem se zadaje u dva dela: opisom modela problema (definisanjem *naziva*, *tipa* i *domena* svih atributa) i spiskom obučavajućih primera, zadanih vektorom vrednostia atributa. Definicija problema se čuva u tekstualnom fajlu koji ima sufiks *.ls* (od *learning set*).

[1] Miškovic V., *Induktivno učenje razumljivog znanja na osnovu oskudnih obučavajućih skupova*, Doktorska disertacija, Univerzitet Singidunum, Beograd, april 2008

[2] Vladislav Miškovic, Milan M. Milosavljević, "Application of Symbolic Inductive Learning Methods to Gene Expression Analyses", in *Proceedings of 9th Symposium on Neural Network Applications in Electrical Engineering*, p.99-102, Belgrade, September 2008. http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4685578

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

005.311.6:004(075.8)

МИШКОВИЦ, Владислав, 1957-
Sistemi za podršku odlučivanju / Vladislav Mišković. - 3. izd. - Beograd : Univerzitet Singidunum, 2024 (Beograd : Caligraph). - X, 246, 22 str. : ilustr. ; 24 cm

Tiraž 300. - Napomene i bibliografske reference uz tekst. - Bibliografija: str. 241-246.

ISBN 978-86-7912-640-5

a) Одлучивање -- Примена рачунара

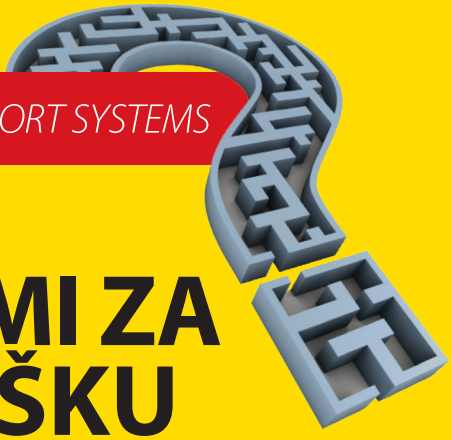
COBISS.SR-ID 149587209

© 2024.

Sva prava zadržana. Nijedan deo ove publikacije ne može biti reprodukovan u bilo kom vidu i putem bilo kog medija, u delovima ili celini bez prethodne pismene saglasnosti izdavača.

www.singidunum.ac.rs

9 788679 126405



DECISION SUPPORT SYSTEMS

Vladislav Miškovic

SISTEMI ZA PODRŠKU ODLUČIVANJU

Računarska podrška u rešavanju strukturiranih problema izlaže se kroz izabrane metode nauke o menadžmentu (ranije operacionih istraživanja), kao što su metodi linearnog programiranja, višekriterijumskog odlučivanja, mreža repova čekanja i računarske simulacije. Računarska podrška odlučivanju u rešavanju nestrukturiranih problema razmatra se kroz izgradnju inteligentnih sistema, kao što su ekspertni sistemi i sistemi za preporučivanje u elektronskoj trgovini, odnosno kroz primenu savremenih metoda poslovne analitike, pre svega istraživanja podataka metodima mašinskog učenja. Zajedničke osnove razmatranih metoda rešavanja nestrukturiranih problema predstavljaju metodi mašinskog učenja, koji su ukratko izloženi u posebnom poglavlju. Neophodni su za izgradnji inteligentnih sistema i predstavljaju osnovne metode istraživanja podataka, odnosno poslovne analitike u podršci odlučivanju.

Univerzitet Singidunum