

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 365

**PREZENTACIJA ZNANJA STABLOM
ODLUČIVANJA**

Mihita Cvitanović

Zagreb, lipanj 2012.

Za Riu

Zadatak diplomskog rada

Naslov: „**Prezentacija znanja stablom odlučivanja**“

Title: „Knowledge Presentation in Decision Tree“

Student: **Mihita Cvitanović**

Mentor: Prof. dr. sc. Branko Jeren

Zadatak

Radi postizanja motiviranijeg, obimnijeg i kvalitetnijeg formaliziranja i ekspliciranja vlastitog znanja u sustavima za upravljanje znanjima, potrebno je razmotriti korištenje prezentacije znanja kroz stablo odlučivanja (eng. Decision tree). Istražiti stanje znanosti i tehnike u tom području te evaluirati raspoložive programske alate. Naglasak staviti na jednostavnost upotrebe za korisnike s minimalnim informatičkim vještinama te na dobavlјivost i troškove korištenja alata. Prilagoditi jedan takav alat i istražiti njegovu upotreblјivost za savjetovanje pri odabiru literature za učenje ili odgovarajućeg obrazovnog programa.

Sadržaj

Uvod.....	1
1. Znanje.....	2
1.1. Definicija znanja.....	2
1.2. Vrste znanja.....	5
1.3. Presentacija znanja	7
1.4. Presentacija znanja u psihologiji obrazovanja	7
1.5. Presentacija znanja u umjetnoj inteligenciji	8
2. Načini presentacije znanja	10
2.1. Presentacija znanja logikom	10
2.1.1. Logičko programiranje.....	11
2.2. Proceduralna presentacija znanja	12
2.2.1. Ekspertni sustavi.....	13
2.3. Mrežna presentacija znanja.....	14
2.3.1. Semantičke mreže	14
2.4. Strukturalna presentacija znanja	15
2.4.1. Okviri	15
2.4.2. Skripte	16
3. Stabla odlučivanja	18
3.1. Stabla odlučivanja u strojnom učenju.....	18
3.2. Stabla odlučivanja u teoriji odlučivanja	19
3.2.1. Programski alati za izradu stabla odlučivanja	21
4. Konceptualne mape	22
4.1. Tematske mape.....	24
4.2. Mentalne mape	25
5. Aplikacija Decision Tree Resolver	27

5.1.	Odabrane tehnologije.....	27
5.1.1.	IHMC CmapTools	27
5.1.2.	Java	29
5.1.3.	HTML	29
5.1.4.	CSS	30
5.2.	Realizacija.....	30
5.3.	Mogućnosti nadogradnje.....	35
	Zaključak	36
	Literatura	37
	Sažetak	39
	Summary	40
	Skraćenice	41
	Privitak	42

Uvod

Tijekom cijelog svog života ljudi uče, stječu određena znanja i postaju stručnjaci za određena područja. Svaki pojedinac prikuplja podatke i informacije iz okoline, povezuje ih i pretvara u vlastito znanje. Može se zaključiti da svaka osoba pohranjuje jedan dio „sveukupnog znanja“. Postavlja se pitanje kako jednostavno prikazati svoje znanje i tako drugima omogućiti njegovo usvajanje i korištenje.

Računala se danas koriste svakodnevno i u gotovo svim sferama ljudskog djelovanja pa se upotreba računala za pohranu i prezentaciju znanja nameće kao sasvim prirodno rješenje. Međutim, iako su računala sposobna izvršavati složene programe i izračune, ljudsko je znanje i zaključivanje u većini domena još uvijek nezamjenjivo. Ipak, danas na raspolaganju već stoje mnoštva sustava za upravljanje znanjem i sustava za potporu odlučivanju. Svi se takvi sustavi oslanjaju na bazu znanja, podatke i informacije koji su formalizirani, tj. napisani u obliku koje računalo razumije.

Prirodni jezik predstavlja jedan od najstarijih i najekspresivnijih načina formalizacije znanja. Knjige, udžbenici i enciklopedije imaju isti cilj: prikazati i prenijeti znanje. Različite se znanstvene discipline bave problemom prezentacije znanja, od logike i filozofije do psihologije obrazovanja. U novije vrijeme, upravo zbog tehnološkog napretka i razvoja računala, ovim se problemom počela baviti umjetna inteligencija.

Ovaj rad će se posebno osvrnuti na stabla odlučivanja i konceptualne mape kao načine prezentacije znanja. Bit će objašnjeni osnovni pojmovi vezani za ove načine prikaza i opisani odgovarajući programski alati za izradu takvih prikaza. Naglasak je na jednostavnosti korištenja za korisnike s minimalnim informatičkim vještinama. U tu svrhu izrađena je web aplikacija koja pretvara konceptualnu mapu u stablo odlučivanja, intuitivno za korištenje svima s minimalnim poznavanjem rada na Internetu.

Struktura rada je sljedeća: prvo poglavlje se bavi konceptom prezentacije znanja, u drugom poglavlju su opisani načini prezentacija znanja, treće poglavlje opisuje stabla odlučivanja i daje pregled programskih alata za izradu stabla odlučivanja, četvrto poglavlje opisuje konceptualne mape te njihove podvrste, mentalne i tematske mape, a peto poglavlje predstavlja izrađenu web aplikaciju Decision Tree Resolver.

1. Znanje

Ovo poglavlje će dati definicije ključnih pojmova vezanih uz rad: znanje i prezentacija znanja. Poglavlje 1.1 definirat će pojam znanja, piramide i kontinuuma znanja, poglavlje 1.2 klasificirat će vrste znanja, poglavlje 1.3 objasniti će osnovne principe prezentacije znanja, a poglavlja 1.4 i 1.5 će se osvrnuti na prezentaciju znanja u psihologiji obrazovanja i umjetnoj inteligenciji.

1.1. Definicija znanja

Ukoliko osoba želi formalizirati i prikazati neko znanje, mora biti sigurna da razumije što je zapravo znanje i koji su njegovi osnovni elementi. Ta naizgled jednostavna zadaća to nimalo nije. Naime, ne postoji univerzalna definicija znanja. Platon je definirao znanje kao opravdano istinsko vjerovanje, tj. vjerovanje da se znanje može potvrditi činjenicama. Ostale definicije znanja kao ključne pojmove za razumijevanje znanja navode podatke i informacije, iskustvo, razumijevanje, generalizaciju i apstrakciju, te socijalno i efektivno djelovanje. U nastavku su navedene neke od poznatijih definicija znanja (Pleslić, 2012):

1. Znanje su informacije u kontekstu – znanje ima vrijednost ako se uklapa bez kontradikcija u širi okvir znanja (Aune, 1970), što znaci da se može promatrati i u širem kontekstu – racionalistički pristup. Znanje su informacije korisne u kontekstu situacija kao što su donošenje odluka ili samih akcija – pragmatični pristup.
2. Znanje je razumijevanje zasnovano na iskustvu – moderni pragmatizam (James, 1907).
3. Znanje su iskustva ili informacije preko kojih se može komunicirati i koje se mogu razmjenjivati (Allee, 1997).
4. Znanje, koje se sastoji od podataka i informacija, se može smatrati puno širim razumijevanjem situacija, odnosa, uzročnih fenomena, i teorija i pravila (eksplicitnih i implicitnih), koji leže u danoj domeni ili problemu (Bennet&Bennet, 2000).

5. Znanje se može smatrati kao da je sastavljeno od razumijevanja, poopćavanja (generalizacija) i apstrakcija, koje nosimo sa sobom na trajnoj ili polutrajnoj bazi i primjenjujemo za interpretaciju i upravljanje svijetom oko nas (Wiig, 1998).
6. Znanje je socijalno djelovanje (Stacey, 1996).
7. Znanje je kapacitet za efektivno djelovanje (Argyris, 1993).

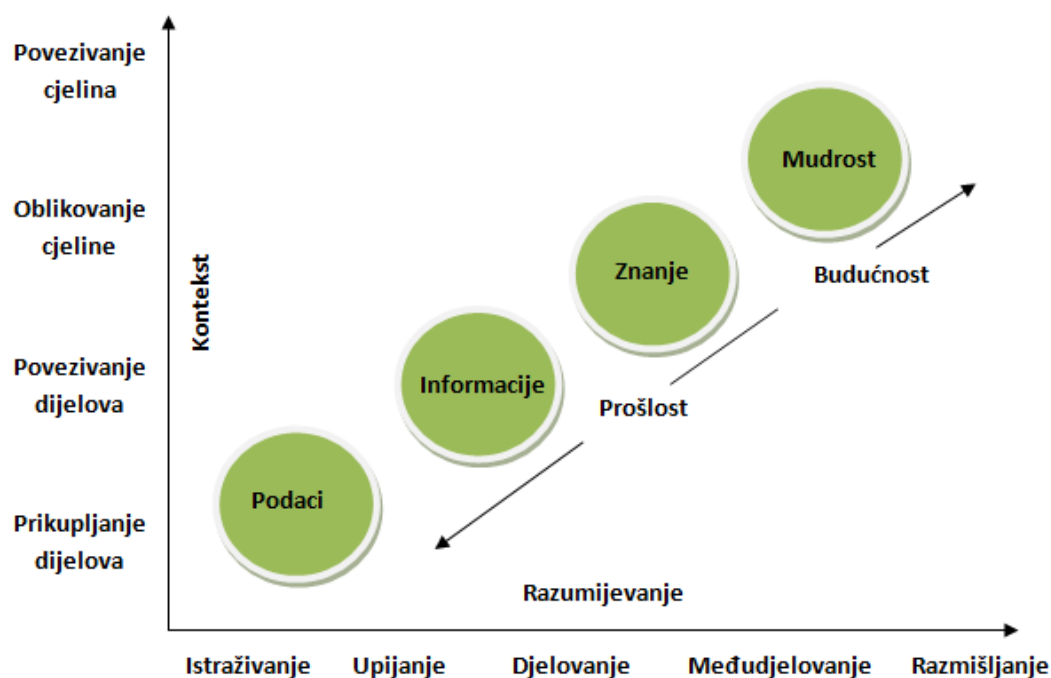
Gore navedene definicije kao sastavne elemente znanja navode podatke i informacije. Sljedeća definicija potvrđuje takvo stajalište: znanje je simbolički prikaz značajki imenovane okoline (engl. *Knowledge is the symbolic representation of aspects of some named universe of discourse*) dok su podaci simbolički prikaz jednostavnih značajki imenovane okoline (engl. *Knowledge is the symbolic representation of simple aspects of some named universe of discourse*) (Frost, 1986).

Treba razlikovati i podatke od informacija. Podatak je vrijednost opaženog, izmjenog ili izračunatog svojstva (atributa). Podaci su definirani kao više vrijednosti jednog takvog svojstva. Informacija je dana preko podatka ili više podataka zato što su podaci uvijek određeni u nekom konceptualnom kontekstu. Općenito, informacije čine podaci i konceptualne interpretacije. Informacije su najčešće podaci ekstrahirani, filtrirani i formatirani na neki način (Pleslić, 2012). Može se zaključiti da su podaci najjednostavniji tip, informacije složeniji tip koji se sastoji od podataka, a znanje najsloženiji tip koji se oslanja na informacije. Čest je prikaz ovog odnosa u obliku piramide znanja ili piramide mudrosti: podaci – informacije – znanje – mudrost piramida (engl. *Data – Information – Knowledge – Wisdom (DIKW)*), prikazane na slici (Slika 1.1). Piramida je strukturalni prikaz iz na čijem dnu su podaci, slijede informacije koje se temelje na podacima, znanje koje sačinjavaju informacije te mudrost kao sposobnost donošenja odluke.



Slika 1.1 Podaci – informacije – znanje – mudrost piramida (Zeleny, 1987)

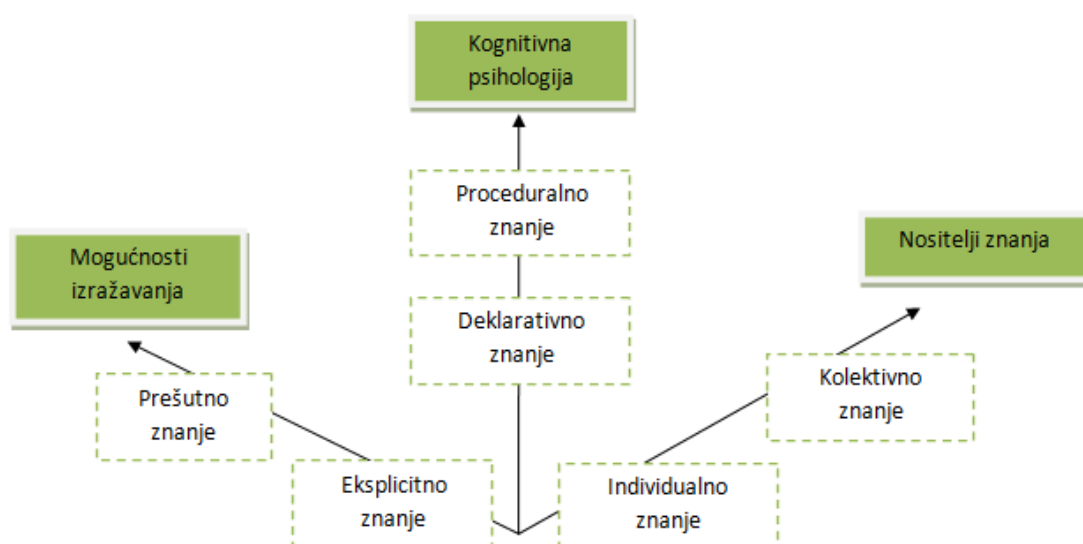
Promatra li se ova struktura u vremenskom okviru, dobit će se kontinuum razumijevanja prikazan na slici (Slika 1.2). Ako je znanje ono što sada posjedujemo, onda su podaci i informacije prikupljeni u prošlosti. Oni se temelje na prikupljanju činjenica i uklapanju činjenica u kontekst. Znanje se kroz dublje razumijevanje pretvara u mudrost koja omogućava donošenje odluka i tako predstavlja budućnost.



Slika 1.2 Kontinuum razumijevanja (Cleveland, 1982)

1.2. Vrste znanja

Brojne su klasifikacije znanja. Ovaj rad će prikazati klasifikaciju znanja u kognitivnoj psihologiji, prema mogućnostima izražavanja te prema nositeljima znanja, kako je prikazano na slici (Slika 1.3).

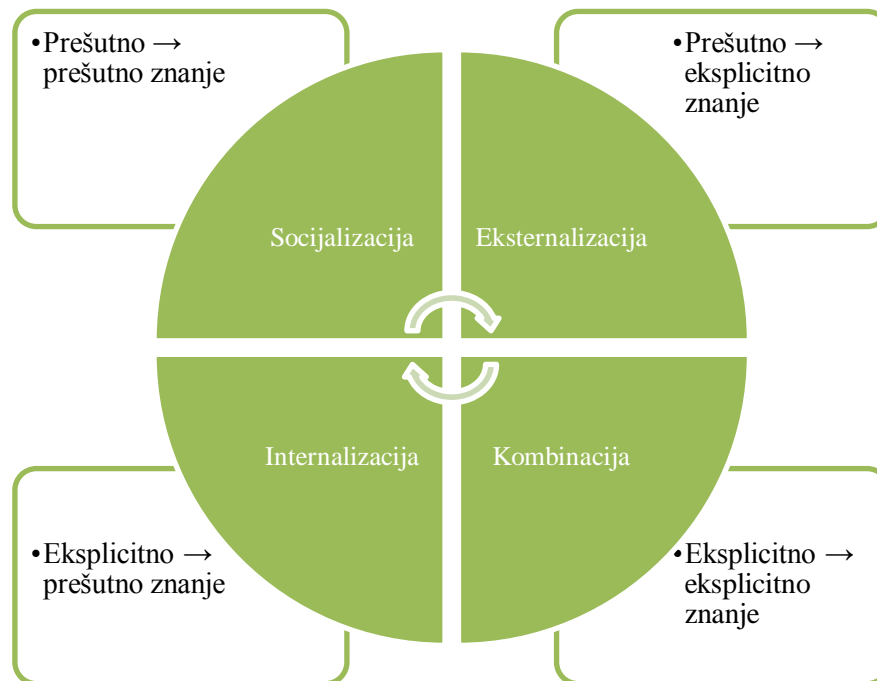


Slika 1.3 Vrste znanja (Prgomet, 2011)

Jedna od podjela znanja u kognitivnoj psihologiji je podjela na deklarativno i proceduralno znanje. Deklarativno znanje se odnosi na teorijsko znanje, znanje činjenica, tj „znanje-što“ (engl. know-that). Ovu vrstu znanja je jednostavno opisati riječima i ona predstavlja temelj međuljudske komunikacije. Deklarativno znanje je opisnog karaktera te uključuje teorije, metode i činjenice koje je potrebno znati kao podlogu praktičnom znanju i vještinama. Proceduralno znanje se odnosi na praktično znanje koje uključuje sposobnosti i vještine. To je primijenjeno znanje, tj „znanje-kako“ (engl. know-how). Proceduralno znanje je teško verbalizirati i vizualizirati, a usvaja se različitim aktivnostima, tj „učenjem kroz rad“ (engl. learning by doing) (Wilde, 2011).

Prema **mogućnostima izražavanja** razlikujemo prešutno od eksplicitnog znanja. Prešutno znanje (engl. *tacit knowledge*) se još naziva implicitno ili iskustveno, a opisuje se kao podsvjesno shvaćeno znanje koje se teško artikulira te je razvijeno iz direktne akcije ili iskustva (Pleslić, 2012). Ovakvo znanje se temelji na uvjerenjima, intuiciji i idealima pojedinca, dijeli se kroz konverzaciju te ga je stoga teško formalizirati i učiniti eksplicitnim. Eksplicitno znanje (engl. *explicit knowledge*) je kodirano i može biti precizno

i formalno artikulirano te je jednostavno za kodiranje, dokumentiranje, transfer, dijeljenje i komuniciranje (Pleslić, 2012). Većina eksplicitnog znanja su tehnički ili akademski podaci i informacije koje su opisane nekim formalnim jezikom. Problem formalizacije prešutnog znanja svodi se na prevođenje prešutnog znanja u eksplicitno. Najpoznatiji model koji opisuje ovaj prijelaz definirali su Nonaka i Takeuchi 1995, a prikazan je na sljedećoj slici (Slika 1.4).



Slika 1.4 SECI model (Nonaka/Takeuchi, 1995)

Ovaj model predstavlja četiri osnovna koraka u prijenosu znanja: socijalizacija, eksternalizacija, kombinacija i internalizacija. Socijalizacija (prešutno znanje u prešutno znanje) se sastoji od dijeljenja znanja kroz socijalne interakcije. Primjer su razgovori licem u lice ili konferencije. Eksternalizacija (prešutno znanje u eksplicitno znanje) vidljivu formu prešutnom znanju i konvertira ga u eksplicitno znanje. Ovaj korak se smatra najvažnijim jer prešutno znanje poprima oblike metafora, analogija, koncepata, hipoteza ili modela. Kombinacija (eksplicitno znanje u eksplicitno znanje) je proces rekombiniranja diskretnih dijelova eksplicitnog znanja u novu formu eksplicitnog znanja. U ovom se koraku ne stvara novo znanje već se poboljšava kvaliteta dosad skupljenog znanja. U zadnjem koraku, internalizaciji (eksplicitno znanje u prešutno znanje) novopristiglo znanje se ugrađuje u znanje pojedinca. Pojedinaac na taj način uči kroz rad (engl. *learning by doing*) i stvara vlastito mišljenje o domeni.

Prema **nositeljima znanja** razlikujemo individualno od kolektivnog znanja. Individualno znanje je znanje pojedinca koje nije ovisno o specifičnom kontekstu i individualno je kontrolirano. Kolektivno znanje je relevantno u specifičnom okruženju, može uključivati individualno znanje koje ima puni potencijal samo kada je kombinirano sa znanjem grupe ljudi. Također može uključivati znanje koje se dijeli sa svima i znanje koje je zajedničko pojedincima u grupi (Prgomet, 2011).

1.3. Prezentacija znanja

Krajnji cilj prezentacije znanja je formalizirati znanje, tj. učiniti ga eksplicitnim. Formalizacija znanja olakšat će sam proces prijenosa i dijeljenja znanja, što je od posebne važnosti u današnje doba kada se računala koriste za pohranu i upravljanje znanjem.

Prezentacija znanja se može definirati kao primjena logike i ontologije u zadaci konstruiranja izračunljivih modela neke domene (Sowa, 2000). Dakle, prezentacija znanja je multidisciplinarno područje koje primjenjuje teorije i tehnike iz logike (definira formalne strukture i pravila zaključivanja), ontologije (definira vrste koncepata koje postoje u području primjene) i izračunavanja (razlikuje koncept prezentacije znanja od čiste filozofije) (Garcia, 2006). Logika i ontologija opisuju mehanizme formalizacije potrebne za izradu eksplicitnih modela koji se lako dijele i koji su prilagođeni računalu. Mogućnost obrade takvih modela različitim izračunavanjima pogoduje automatizaciji velikih količina znanja.

Računala danas igraju važnu ulogu u prezentaciji znanja. Međutim, njihova uloga je najčešće obrada velikih količina informacija, a često su ljudi opet ti koji moraju donijeti krajnju odluku ili zaključak. Jedan od glavnih zadataka umjetne inteligencije je izgraditi takav sustav prezentacije znanja koji će biti potpora i procesima odlučivanja i zaključivanja.

1.4. Prezentacija znanja u psihologiji obrazovanja

U knjigama, priručnicima i enciklopedijama znanje je opisano riječima, ilustracijama, grafovima i tablicama. Proces prikupljanja i prezentacije znanja u obrazovnim materijalima istražuje psihologija odgoja i obrazovanja.

Nastavni materijali bi trebali prikazati znanje i modelirati njegovu strukturu na razumljiv i intuitivan način koji će potaknuti učenje i pamćenje. Mnogi autori podupiru Schmidtovu teoriju sheme koja kaže da ljudi uče na način da stvaraju mentalnu mrežu znanja u kojoj nove informacije postaju novi čvorovi u mreži koji se povezuju s već postojećim čvorovima. Kvalitetni obrazovni materijali prikazuju znanje tako da potiču povezivanje novih činjenica s već poznatim i tako omogućuju dublje razumijevanje cijele domene (Jonassen, 1986). Čitatelji razumiju tekst tako što konstruiraju odgovarajuće sadržajne sheme (organizirano znanje o objektima, događajima, situacijama) i tekstualne sheme (organizirano znanje o tekstovima od dijelova teksta do bajki i znanstvenih članaka) (Armbruster, 1986). Sadržajne sheme se ostvaruju uporabom analogija i primjera. Tekstualne sheme se ostvaruju kroz strukturu i organizaciju sadržaja. Signali predstavljaju sredstva i načine kojima se autori služe za strukturiranje tekstova, a neki od primjera su veznici (uzročni, sastavni, rastavni, vremenski), riječi naglašavanja i izražavanja stavova, tipografska naglašavanja poput podebljavanja i ukošavanja slova, podcrtavanje, pregledne rečenice (engl. *preview statements*) te sažeci. Također, namjena i struktura trebaju biti jasno izražene kroz naslove i podnaslove; a autorov stav mora biti iskazan samom strukturom. Struktura mora biti konzistentna kroz cijeli tekst.

Međutim, iako postoje smjernice za stvaranje kvalitetnih obrazovnih materijala, ne postoje jedinstveni standardi za procjenu kvalitete prikaza znanja (Nazarova et al., 2006). Tehnološki napredak bi trebao pomoći kako u razvijanju obrazovnih materijala, tako i u samom nastavnom procesu (pojava e-učenja i e-udžbenika).

1.5. Prezentacija znanja u umjetnoj inteligenciji

Prezentacija znanja u umjetnoj inteligenciji se definira preko pet svojih temeljnih uloga (Davis et al., 1993):

1. Prezentacija znanja je nadomjestak (engl. *surrogate*). Simboli se koriste za prikaz stvari i pojmova koji se ne mogu pohraniti na računalu, npr. fizički objekti, događaji, veze. Simboli su prema tome nadomjestak za vanjske objekte. Simboli i poveznice među njima zajedno čine model vanjskih objekata kojim se može upravljati te zaključivati na osnovu njega.
2. Prezentacija znanja je skup ontologijskih opredjeljenja (engl. *ontological commitments*). Ontologija je definirana kao znanost o postojanju, a u području prezentacije znanja ona određuje kategorije vanjskih objekata koje mogu postojati u

domeni primjene. Te kategorije predstavljaju ontologijska opredjeljenja koja je postavio inženjer znanja (engl. *knowledge engineer*).

3. Prezentacija znanje je fragmentarna teorija inteligentnog zaključivanja (engl. *a fragmentary theory of intelligent reasoning*). Kako bi se omogućilo zaključivanje na osnovi izgrađenog modela, prezentacija znanja mora opisati ponašanje i interakciju vanjskih objekata u modelu. Taj opis čini temelj za teoriju domene primjene izgrađenog modela.
4. Prezentacija znanja je medij za učinkovita izračunavanja. Znanje u inteligentnom sustavu mora biti prikazano na način da je pogodno za analize i izračunavanja pomoću dostupne računalne opreme. Iz tog razloga računalno i programsko inženjerstvo imaju velik utjecaj na prezentaciju znanja.
5. Prezentacija znanja je medij ljudskog izražavanja. Kvalitetna prezentacija znanja mora omogućiti lakšu komunikaciju između inženjera znanja koji upravljaju tehnologijama znanja i domenskih stručnjaka koji razumiju samu domenu primjene. Domenski stručnjaci moraju moći čitati, razumjeti i provjeriti definicije i pravila koja su inženjeri znanja ugradili u inteligentni sustav prezentacije znanja.

Ovaj rad će se orijentirati na prezentaciju znanja kao granu umjetne inteligencije. Sljedeće poglavlje (Načini prezentacije znanja) opisat će načine prezentacije znanja s aspekta umjetne inteligencije.

2. Načini prezentacije znanja

Više je načina prezentacije znanja te ne postoji jedan univerzalni formalizam pogodan za prezentaciju cjelokupnog znanja. Pri odabiru načina prezentacije znanja treba obratiti pozornost na ono što treba biti formalizirano i na način na koji prikazano znanje treba biti analizirano i upotrijebljeno.

Značajke dobre prezentacije znanja su sljedeće (Kerber, 2004):

- Važni objekti i odnosi među njima su jasno određeni
- Prikazani odnosi su prirodni i intuitivni; jasno se može izraziti i vidjeti kako jedan objekt utječe na drugi
- Manji detalji su skriveni na način da im se može pristupiti po potrebi
- Transparentnost; lako je razumljivo ono što se prikazuje
- Potpunost; može se izraziti sve što se želi izraziti
- Preciznost; može se učinkovito izraziti sve što se želi izraziti
- Brzina i izračunljivost; informacije se brzo pohranjuju, ali im se isto tako i brzo pristupa te ih se brzo obrađuje

Ovo poglavlje će opisati podjelu načina prezentacije znanja na prezentaciju znanja logikom (2.1), proceduralnu prezentaciju znanja (2.2), mrežnu prezentaciju znanja (2.3) te strukturalnu prezentaciju znanja (2.4).

2.1. Prezentacija znanja logikom

Prirodnim jezikom najlakše je prikazati, objasniti i prenijeti znanje. Prirodni jezik je izrazito ekspresivan, što je velika prednost, ali postaje nedostatak kada neko znanje želimo prenijeti računalu jer ne postoji jedan univerzalni način za prikazati to znanje. Pored toga, prirodni jezik može biti dvosmislen i sintaksa nerazumljiva što može uzrokovati poteškoće pri zaključivanju.

Logika se za prezentaciju znanja upotrebljava upravo kako bi se riješili problemi prikaza prirodnim jezikom, a predstavlja formalni jezik temeljen na matematičkim principima.

Logiku možemo definirati kao znanost o ispravnom zaključivanju. Ispravno zaključivanje čuva istinu. Dakle, logika je znanost o ispravnom zaključivanju s ciljem očuvanja istine (Reichgelt, 1991). Logički prikaz znanja mora biti izrazit, nedvosmislen, neovisan o kontekstu te imati jasno definirane elemente i pravila. Sljedeći koncepti moraju biti definirani (Garcia, 2006):

- **Rječnik:** kolekcija simbola koji mogu biti znakovi, riječi ili čak zvukovi
- **Sintaksa:** pravila koja određuju kako će se simboli kombinirati u izraze
- **Semantika:** određuje značenje izraza, provjerava njihovu istinitost te određuje odnose među simbolima
- **Pravila zaključivanja:** pravila koja određuju zaključivanje na temelju postojećeg znanja te tako formiraju novo znanje

S obzirom na definirane koncepte, postoji više vrsta logike, svaka prilagođena određenoj domeni primjene. Za prikaz znanja u umjetnoj inteligenciji koristi se **predikatna logika prvog reda**, prvenstveno zbog svoje semantike i ekspresivnosti. Predikatna logika pruža širu ontologiju nego propozicijska logika zbog upotrebe varijabli, predikata i funkcija obogaćenih kvantifikatorima.

Logički programski jezici temelje se na predikatnoj logici. U nastavku su opisane glavne karakteristike logičkog programiranja i najčešće upotrebljavan logički programski jezik Prolog.

2.1.1. Logičko programiranje

Logičko programiranje je poseban pristup programiranju, paradigma uspoređivana s imperativnim i funkcijskim programiranjem (Pfenning, 2006). Logičko programiranje se temelji na matematičkoj logici i glavni cilj mu je pronaći dokaz za neku činjenicu (engl. *proof search*). Ovisno o konkretnoj implementaciji jezika, pri traženju dokaza se koristi pravilo ulančavanja unaprijed (engl. *forward-chaining*) ili ulančavanja unatrag (engl. *backwards-chaining*). Ulančavanje unaprijed kreće od poznatih aksioma i pokušava dokazati pretpostavku dok ulančavanje unatrag kreće od pretpostavke i pokušava ju dokazati tako što će doći do važećeg aksioma. Ulančavanje unatrag je dominantan pristup logičkom programiranju jer se na njemu temelji i najpoznatiji logički programski jezik Prolog.

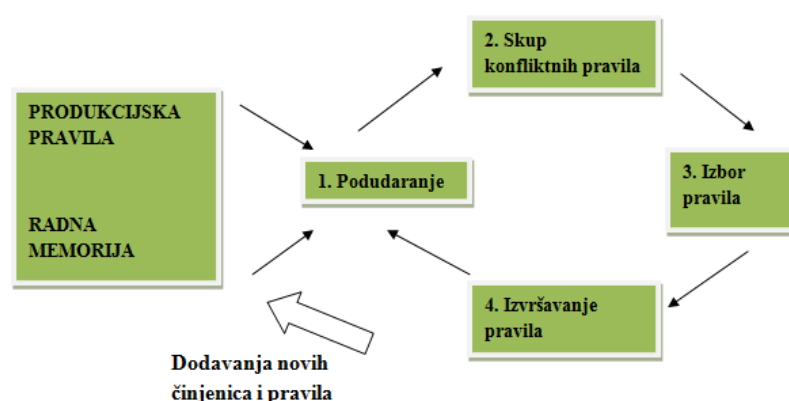
Prolog je jedan od prvih logičkih programskih jezika temeljen na predikatnoj logici i ulančavanju unatrag uz određene modifikacije. Činjenice su u Prologu izražene u Hornovom obliku, a negacija ne postoji, tj. ako se nešto ne može dokazati iz činjenica koje postoje, tada se to smatra netočnim. Ulančavanje unatrag implementirano je kao pretraživanje u dubinu koje pronalazi prvo rješenje, a ukoliko algoritam zapne, vraća se natrag i traži novi put ka rješenju.

2.2. Proceduralna prezentacija znanja

Sustavi temeljeni na pravilima tj. produkcijski sustavi (engl. *production systems*) predstavljaju proceduralnu prezentaciju znanja. Produkcijski sustav definiran je sa (Reichgelt, 1991):

- **Radnom memorijom:** baza činjenica koja sadrži opis tekućeg stanja svijeta u postupku zaključivanja; većina produkcijskih sustava sprema elemente u radnu memoriju po principu objekt-atribut-vrijednost ili atribut-relacija-vrijednost
- **Skupom produkcijskih pravila:** svako produkcijsko pravilo predstavlja jedan uvjet-akcija par koji prikazuje jedan dio domenskog znanja; ovako prikazano znanje je pohranjeno za stalno; produkcijska pravila su opisano kao AKO-ONDA pravila (engl. *IF-THEN rules*)
- **Interpreterom tj. ciklusom podudaranje-djelovanja:** upravljački mehanizam koji se sastoji od slijeda akcija: podudaranje uzoraka (engl. *pattern matching*) – izbor pravila rješavanjem konflikata - izvršavanje pravila

Shema rada produkcijskog sustava prikazana je na sljedećoj slici (Slika 2.1)



Slika 2.1 Shema rada produkcijskog sustava (Dalbello-Bašić, 2012)

Uzorci iz radne memorije uspoređuju se s uvjetnim dijelovima pravila iz baze pravila. Kada se uvjetni dio nekog pravila podudara s uzorcima iz radne memorije, onda je to pravilo omogućeno i ono se izdvaja u skup konfliktnih pravila. Na početku ciklusa raspoznavanja konfliktni skup je prazan. Nakon postupka podudaranja konfliktni skup može sadržavati jedno ili više omogućenih pravila. Ako je skup konfliktnih pravila prazan onda je postupak završen. Ako ima više omogućenih pravila izabire se JEDNO pravilo koje se izvršava, tj. koje pali. Odabir pravila koje pali naziva se razrješavanje konflikata. Paljenje pravila znači izvršavanje desnog dijela odabranog pravila koje mijenja sadržaj radne memorije. Nakon paljenja pravila upravljački ciklus ponavlja se s promijenjenim sadržajem radne memorije (Dalbelo-Bašić, 2012).

Izvršavanje pravila pod utjecajem je stroja za zaključivanje (engl. *inference engine*). Ono se može grafički prikazati u stablu odlučivanja. Stablo odlučivanja kao način prikaza znanja i kao alat za potporu odlučivanju biti će detaljno opisano u poglavlju Stabla odlučivanja.

2.2.1. Ekspertni sustavi

Ekspertni sustavi (engl. *expert systems*) su zapravo sinonim sustavima temeljenim na pravilima i produkcijskim sustavima. Njihova osnovna zadaća je sakupljanje i korištenje specijaliziranog znanje iz neke problemske domene da bi se riješio problem na razini ljudskog eksperta. Dvije osnovne komponente ekspertnog sustava su baza znanja i stroj za zaključivanje. Međutim, ekspertni sustavi još nisu stekli širok krug korisnika jer je potrebna uska suradnja domenskih i računalnih stručnjaka da bi se domensko znanja pretvorilo u bazu znanja ekspertnog sustava.

Postoje posebni programski alati za izradu ekspertnih sustava. Jedan od najčešće upotrebljivanih je **CLIPS** (engl. *C Language Integrated Production System*). Popularnost CLIPS-a leži u činjenici da je to besplatan, učinkovit i brz alat koji je napisan u programskom jeziku C. Razvijena je i objektno-orijentirana inačica CLIPS-a nazvana COOL. CLIPS koristi unaprijed ulančano zaključivanje, a svakom od pravila sadržanih u bazi znanja dodaje atribut istaknutosti (engl. *salience*), kako bi se odredio prioritet izvršavanja pravila.

2.3. Mrežna prezentacija znanja

Mrežna prezentacija znanja je koncept preslikan u umjetnu inteligenciju iz drugih disciplina poput psihologije, filozofije i lingvistike. U poglavlju Prezentacija znanja u psihologiji obrazovanja već je spomenuta Schmidtova teorija sheme koja kaže da ljudi uče nove pojmove tako što ih nastoje ugraditi u mrežu njima već poznatih koncepata. Mrežni modeli se tako u umjetnoj inteligenciji sastoje od tri glavna elementa: čvorova koji sadrže određeni koncept, informacija vezanih za koncept u čvoru te veza među čvorovima. Mrežni model je također hijerarhijski model koji uključuje nasljeđivanje atributa i različite vrste veza (disjunkcija, konjunkcija, podređenost, nadređenost). Mrežni modeli su najpogodniji za prikaz statičkog deklarativnog znanja. U nastavku su opisane semantičke mreže koje predstavljaju najpoznatiji mrežni prikaz znanja u umjetnoj inteligenciji.

2.3.1. Semantičke mreže

Semantička mreža je grafički prikaz znanja u obliku čvorova povezanih različitim vrstama asocijativnih veza koje su predočene usmjerenim lukovima u grafu (Sowa, 2006). Dakle, dvije su glavne komponente semantičkih mreža: čvorovi i veze. Čvorovi predstavljaju objekte ili klase objekata dok veze odgovaraju poveznicama između objekata. Veze obično imaju ime kako bi se objasnilo njihovo značenje. Sami čvorovi ne sadrže nikakve dodatne informacije o konceptu, ali zato postoje različite vrste veza koje mogu pobliže objasniti koncept i njegovu ulogu u cijelom modelu.

Veze među čvorovima mogu definirati pripadnost neke klase objektu, nasljeđivanje, uključivanje i isključivanje. Takav hijerarhijski model omogućuje jednostavno snalaženje, pretraživanje u mreži i točno lociranje traženih informacija. Međutim, iako imaju jasnu sintaksu i mehanizme nasljeđivanja, semantičke mreže nisu standardizirane. Konceptualne mape (engl. *concept maps*) predstavljaju oblik semantičkih mreža i bit će detaljno objašnjene u poglavlju Konceptualne mape. Podvrste konceptualnih mapa su mentalne (engl. *mind maps*) i tematske mape (engl. *topic maps*) koje će također biti opisane u nastavku. Tematske mape su jedini standardizirani oblik konceptualnih mapa.

2.4. Strukturalna prezentacija znanja

Strukturalna prezentacija predstavlja nadogradnju mrežnog modela prikaza jer ima učinkovite mehanizme za prikaz i deklarativnog i proceduralnog znanja, ali u osnovi ima iste elemente kao i mrežni prikaz. Cilj strukturalne prezentacije znanja je prikazati neki dio znanja na jasan i strukturiran način. Glavni elementi prikaza su strukturirani objekti s definiranim atributima. Također mogu postojati određene hijerarhijske veze između objekata. Te veze odgovaraju vezama u mrežnom modelu, a objekti odgovaraju čvorovima mreže. Mehanizmi za prikaz deklarativnog znanja nazivaju se okviri, a za prikaz proceduralnog znanja skripte. Okviri i skripte su opisani u nastavku.

2.4.1. Okviri

Okvire (engl. *frames*) možemo definirati kao kolekciju pitanja o nekoj hipotetskoj situaciji što će ukazati na načine na koje ta pitanja možemo riješiti (Minsky, 1974).

Okviri predstavljaju entitete tj. objekte kao skupinu polja (engl. *slots*) s pridruženim vrijednostima (engl. *value*). Svako polje može ograničiti vrijednosti i vrste atributa koje može sadržavati. Okviri mogu sadržavati opis samo jednog entiteta ili pak cijele jedne domene. Povezanost okvira se ostvaruje tako što vrijednost polja jednog okvira može biti cijeli drugi okvir.

Okviri su svojevrsni nastavak na semantičke mreže jer pored prikaza objekata i odnosa među njima, omogućuju i spremanje proceduralnih privitaka (engl. *procedural attachments*). To su funkcije napisane nekim programskim jezikom koje mogu biti spremljene kao vrijednost nekog polja. Ova funkcionalnost pridaje dinamičnosti prikaza znanja u okvirima, iako se za prikaz proceduralnog znanja prvenstveno koriste skripte, definirane kao podvrsta okvira.

Koncept okvira je bio preteča za razvoj objektno-orijentirane paradigme. Ta se paradigma temelji upravo na objektima i klasama, a veze među njima su definirane referencama na objekte u varijablama klase. Također, ostvareni su i mehanizmi nasljeđivanja, a proceduralni privitci iz okvira su predstavljeni kao metode klase.

2.4.2. Skripte

Skripte (engl. *scripts*) su strukturalni način prezentacije znanja sličan okvirima, ali više namijenjen prikazu proceduralnog znanja. Skripte su definirane kao strukturalni prikaz koji opisuje uobičajeni redoslijed događaja u određenom kontekstu (Luger et al., 1998). Dakle, skripte proširuju okvire tako što eksplicitno prikazuju očekivanja i ishode postupaka.

Najpoznatiji primjer iz literature je primjer restoranske skripte (Schank et al., 1977) koji opisuje posjet restoranu kroz četiri koraka: ulazak, narudžba, jelo i izlazak.

Slika (Slika 2.2) prikazuje parametre koje je potrebno definirati prije pisanja same skripte. U slučaju restoranske skripte to su: mjesto odvijanja radnje, rekviziti, uloge, početni uvjeti i mogući ishodi.

Skripta: posjet restoranu			
Mjesto radnje:	restoran	Početni uvjeti:	K gladan
			K ima novaca
Rekviziti:	stolovi		
	cjenik		
	H = hrana	Rezultati:	K ima manje novaca
	račun		V ima više novaca
	novac		K nije gladan
			K je zadovoljan (mogućnost)
Uloge:	K = klijent		
	W = konobar		
	C = kuhar		
	B = blagajnik		
	V = vlasnik		

Slika 2.2 Restoranska skripta 1 (Schank et al., 1977)

Slika (Slika 2.3) opisuje scene ulaska u restoran i procesa naručivanja, dok slika (Slika 2.4) opisuje čin jela i na kraju, izlazak iz restorana. Izrazi PTRANS, ATRANS, MBUILD, ATTEND, MOVE i GO predstavljaju predikate kojima se opisuju izrazi u skripti.

Scena 1: Ulazak	Scena 2: Narudžba	
K PTRANS K u restoran	(Jelovnik na stolu)	(K traži jelovnik)
K ATTEND pogled k stolovima	K PTRANS jelovnik K-u	K MTRANS znak W-u
K MBUILD gdje sjesti		W PTRANS W prema stolu
K PTRANS prema stolu		K MTRANS treba jelovnik W-u
K MOVE K u sjedeću poziciju		W PTRANS W prema jelovniku
		W PTRANS W prema stolu
		W ATRANS jelovnik K-u
	*K MBUILD izbor H	
	K MTRANS znak W	
	W PTRANS W prema stolu	
	K MTRANS "želim F" W	
		W PTRANS W C-u
	C MTRANS "nema H" W-u	C DO H
	W PTRANS W K-u	idi na scenu 3
	W MTRANS "nema H" K-u	
	(idi natrag na *) ili	
	(idi na scenu 4, neplaćanje)	

Slika 2.3 Restoranska skripta 2 (Schank et al., 1977)

Scena 3: Jelo	Scena 4: Izlazak iz restorana
C ATRANS H W-u	W MOVE (napisati račun)
W ATRANS H K-u	W PTRANS W K-u
K jede H	W ATRANS račun K-u
	K ATRANS napojnica W-u
	K PTRANS K W-u
	K ATRANS novac B-u
	K PTRANS K na izlazak iz restorana
(Opcija: Povratak na scenu 2 za dodatnu narudžbu; inače, idi na scenu 4)	

Slika 2.4 Restoranska skripta 3 (Schank et al., 1977)

3. Stabla odlučivanja

Stablo odlučivanja (engl. *decision tree*) je naziv za stablasto strukturirani dijagram koji može prikazivati sva moguća rješenja nekog problema, sve moguće ishode nekog događaja ili pak klasificirati određene podatke. Svaka grana stabla odlučivanja se obično može prikazati kao jedno AKO-ONDA (engl. IF-THEN) pravilo. Iako je struktura slična, bitno je razlikovati dvije potpuno različite domene primjene stabla odlučivanja: stablo odlučivanja u strojnom učenju i stablo odlučivanja u teoriji odlučivanja.

Ovaj rad će dati samo kratak uvod u stabla odlučivanja u strojnom učenju, a naglasak će staviti na stabla odlučivanja u teoriji odlučivanja.

3.1. Stabla odlučivanja u strojnom učenju

Stablo odlučivanja u strojnom učenju koriste se za klasifikacijske i predikacijske probleme, tj. stroj sam gradi stablo odlučivanja primjenjujući određeni algoritam na određeni skup podataka. Stablo odlučivanja jest klasifikacijski algoritam u formi stablaste strukture, u kojoj se razlikuju dva tipa čvorova povezanih granama:

- čvor odluke (engl. *decision node*): ovaj čvor definira određeni uvjet u obliku vrijednosti određenog atributa (varijable), iz kojeg izlaze grane koje zadovoljavaju određene vrijednosti tog atributa
- krajnji čvor (engl. *leaf node*): kojim završava određena grana stabla.; krajnji čvorovi definiraju klasu kojoj pripadaju primjeri koji zadovoljavaju uvjete na toj grani stabla

Stablo odlučivanja može se koristiti za klasifikaciju primjera, tako da se krene od prvog čvora odlučivanja u korijenu stabla i kreće po onim granama stabla koja zadovoljava primjer sa svojim vrijednostima sve do krajnjeg čvora koji klasificira primjer u jednu od postojećih klasa problema (Gamberger et al., 2001).

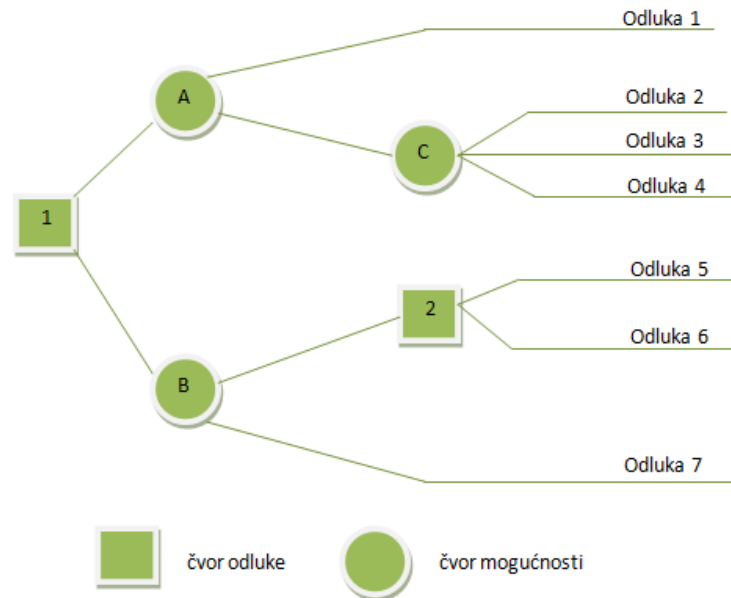
3.2. Stabla odlučivanja u teoriji odlučivanja

Stablo odlučivanja u teoriji odlučivanja služi kao grafička pomoć pri donošenju odluka, analiziranju mogućih ishoda ili procjeni rizika. Dakle, u teoriji odlučivanja mora postojati osoba koja će izgraditi stablo odlučivanja, primjerice ekspert iz danog područja, za razliku od stabla odlučivanja u strojnom odlučivanju koja gradi stroj na temelju ulaznog skupa podataka. U daljnjem tekstu će se pojam stabla odlučivanja odnositi na stablo odlučivanja u teoriji odlučivanja.

Stablo odlučivanja je u svojoj suštini dijagram koji predstavlja, na posebno organiziran način, odluke, glavne vanjske ili druge događaje koji unose nesigurnost, kao i sve moguće ishode svih odluka i događaja (Rahma, 2012). Glavne komponente stabla odlučivanja su:

- čvor odluke (engl. *decision node*): predstavlja odluke koje treba donijeti, a prikazan je kvadratom
- čvor mogućnosti/nesigurnosti (engl. *chance/uncertainty node*): predstavlja vjerojatnost određenog vanjskog događaja na koji osoba koja donosi odluku nema izravne kontrole
- odluke/posljedice/ishodi (engl. *consequences/outcomes*): na primjer mogu predstavljati troškove, zaradu, ili pak odrediti od koje bolesti pacijent boluje

Slika (Slika 3.1) prikazuje glavne komponente stabla odlučivanja.



Slika 3.1 Komponente stabla odlučivanja

Proces gradnje stabla odlučivanja sastoji se od sljedeća četiri koraka:

1. razgovor s domenskim stručnjacima i zainteresiranim strankama te konstrukcija preliminarnog stabla
2. prezentacija stabla korisnicima
3. diskusija o predloženom rješenju
4. ponovna gradnja stabla po potrebi

Brojne su prednosti uporabe stabla odlučivanja u teoriji odlučivanja:

- stabla odlučivanja jasno analiziraju problem tako da sve mogućnosti dolaze u obzir
- sadrže potpunu analizu svih posljedica neke odluke
- nude kvantitativnu analizu ishoda
- predstavljaju model bijele kutije (engl. *white-box model*) što znači da se jednostavno može vidjeti kako se došlo do određenog rezultata
- pomažu pri donošenju najboljih odluka temeljenih na postojećim informacijama i procjenama stručnjaka

Međutim, stabla odlučivanja imaju i svojih nedostataka, poput:

- male promjene u ulaznim podacima koji grade stablo mogu uzrokovati velike promjene u cijelom stablu
- odluke prezentirane u stablu su dijelom zasnovane na predviđanjima stručnjaka i očekivanjima koja ponekad mogu biti iracionalna ili neprecizna
- velika stabla odlučivanja mogu biti nezgrapna i složena za korištenje

Stabla odlučivanja imaju široku primjenu u području operacijskih istraživanja, discipline koja se bavi primjenom složenih analitičkih metoda za donošenje boljih odluka. Neki od ostalih sfera primjene su ekonomija, menadžment i projektni menadžment, zdravstvo i medicina te različiti pravni procesi i parnice.

3.2.1. Programski alati za izradu stabla odlučivanja

Stabla odlučivanja se mogu jednostavno nacrtati ručno na listu papira. Ipak, uporaba računala za izradu stabla odlučivanja je bolje rješenje jer olakšava promjene i izvođenje izračuna. Jednostavni alati za obradu teksta i izradu tablica poput MS Word-a i MS Excel-a mogu poslužiti u tu svrhu. Također, postoje brojni specijalizirani alati za izradu stabla odlučivanja. Jednom kada korisnika nauči koristiti odabrani alat, proces stvaranja stabla postaje brz i učinkovit. No, treba imati na umu da je alat dobar koliko i korisnik alata. Alat ne donosi odluku, nego samo slijedi korisnikove instrukcije, a na korisniku je da nauči davati dobre instrukcije. Slijedi kratka lista nekih od alata za izradu stabla odlučivanja:

1. Insight Tree (<http://www.visionarytools.com/index.htm>): jednostavan za uporabu ako je korisnik već upoznat s konceptom stabla odlučivanja; omogućava brzo dodavanje grana, optimiziranje čvorova, korištenje različitih stilova; besplatan je za osobnu uporabu
2. Lumenaut (<http://www.lumenaut.com/>): koristi se kao ugrađeni dodatak (engl. *addin*) MS Excel-u; pored gradnje stabla odlučivanja pruža i ostale alate za statističku analizu poput Monte Carlo simulacije te analize osjetljivosti; komercijalni alat čija je probna verzija dostupna
3. Vanguard Studio (<http://www.vanguardsw.com/products/vanguardsystem/components/vanguard-studio/>): zasebna aplikacija koja podržava posebni čarobnjak za izradu stabla odlučivanja koja korisnike vodi korak po korak kroz proces stvaranja stabla; također nudi i ostale slične alate za statističku analizu; komercijalni alat
4. SmartDraw (<http://www.smartdraw.com/>): komercijalni alat koji omogućava brzo i jednostavno crtanje različitih vrsta dijagrama, među kojima su i stabla odlučivanja; sadrži i veliku bazu primjera za svaku vrstu dijagrama

4. Konceptualne mape

Konceptualne mape ili pojmovne mape (engl. *concept maps*) su oblik mrežnog prikaza znanja i predstavljaju primjer semantičke mreže. Uz pojmove konceptualne mape vežu se i pojmovi mentalne i tematske mape. U ovom radu podjela će biti načinjena na način da se konceptualne mape smatraju širim konceptom od koncepta mentalnih i tematskih mapa jer alati za izradu konceptualnih mapa mogu kreirati i mentalne i tematske mape. U nastavku su opisane konceptualne mape, a u poglavljima 4.1 i 4.2 mentalne i tematske mape redom.

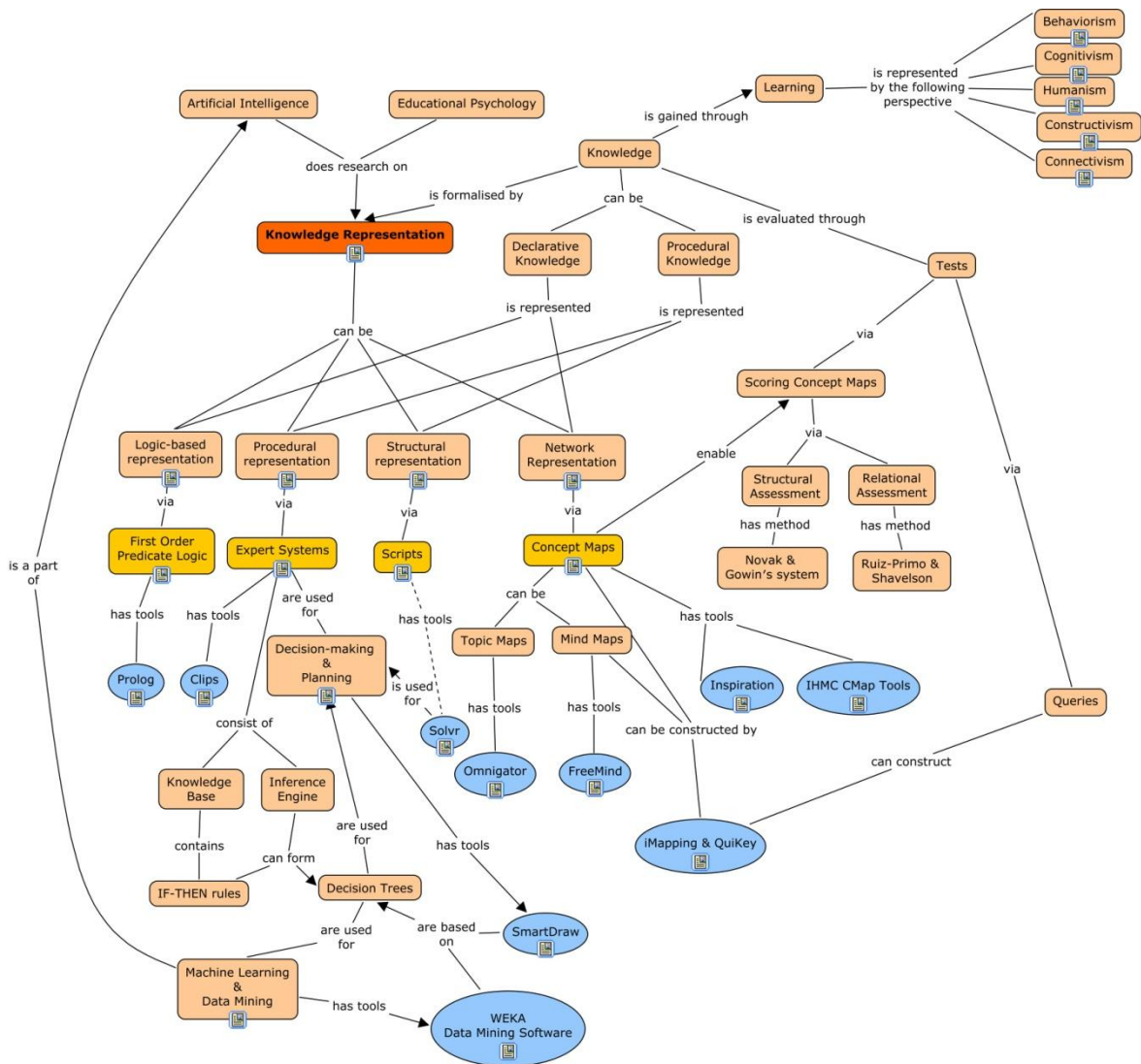
Konceptualne mape su grafički alat za organizaciju i prezentiranje znanja. One uključuju pojmove (obično zatvorene u kružnice ili okvire određenog tipa) (engl. *concepts*) i veze (engl. *relationships*) između pojmova koje su prikazane linijama koje povezuju dva pojma. Riječi na liniji, znane kao povezujuće riječi ili povezujuće fraze (engl. *linking phrases*), određuju vezu između dva pojma.

Važna karakteristika pojmovnih mapa je u tome da su pojmovi predstavljeni hijerarhijski sa najviše uključenim i najviše općim pojmovima na vrhu mape i sa specifičnijim, manje općim pojmovima hijerarhijski niže. Hijerarhijska struktura za određeno područje znanja također ovisi o sadržaju u kojem se to znanje koristi ili se uzima u obzir. Stoga, najbolje bi bilo konstruirati pojmovnu mapu s naglaskom na neka određena pitanja na koja se želi odgovoriti. Ta pitanja se nazivaju fokusna pitanja (engl. *focus questions*).

Sljedeća važna karakteristika pojmovnih mapa je uključivanje ukrštenih poveznica (engl. *cross-links*). To su veze između pojmova jednog segmenta ili domene s drugim segmentima ili domenama pojmovnih mapa. Ukrštene poveznice pomažu u shvaćanju kako su na mapi predstavljeni pojmovi u jednoj domeni znanja povezani s pojmovima u drugoj domeni. U stvaranju novog znanja, ukrštene poveznice često predstavljaju kreativne skokove u ulozi proizvođača znanja. Ukrštene poveznice omogućuju lakše pretraživanje mape, a zajedno s hijerarhijskom strukturom čine osnovu za razvoj kreativnog razmišljanja (Novak et al., 2006).

Slika (Slika 4.1) . prikazuje primjer konceptualne mape koja opisuje područje prezentacije znanja istraženo prilikom izrade ovog rada i ilustrira gore navedene karakteristike.

Tehniku konceptualnog mapiranja razvio je Joseph D. Novak i njegov tim na sveučilištu Cornell 70-ih godina prošlog stoljeća. Konceptualne mape imaju svoje porijeklo u učenju kretanja zvanog konstruktivizam. Konstruktivizam smatra da učenik aktivno gradi znanje. Novakov rad se bazirao na kognitivnim teorijama Davida Ausubela (teorija asimilacije), koji je naglašavao važnost predznanja kao mogućnost da se uče novi pojmovi.



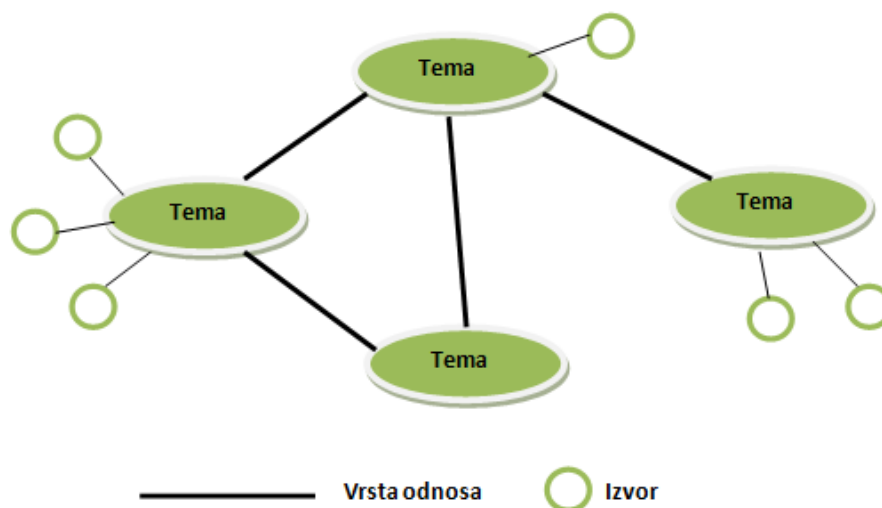
Slika 4.1 Konceptualna mapa Presentacija znanja

4.1. Tematske mape

Tematske mape, tematske karte (engl. *topic maps*) međunarodni su standard (*XTM Topic Maps ISO/IEC 13250:2000*) s glavnim ciljem organizacije, korištenja i razmjene sustava za organizaciju znanja. Mrežna verzija standarda poznata je pod nazivom XML Topic Maps s ciljem snalaženja u strukturama znanja određenog područja i njihovog povezivanja s informacijskim izvorima. Standardi omogućavaju predstavljanje vrlo složenih struktura znanja te su koristan alat za upravljanje i navigaciju te pretraživanje i pronalaženje informacija (Mihalić, 2008). Glavna zamisao tematskih mapa sastoji se u konceptu temeljnog modela koji se sastoji od:

- Tema (engl. *topic*): predstavlja bilo koji koncept, od ljudi, zemalja, organizacija, do događaja i programskih alata
- Izvora (engl. *occurrence*): predstavljaju izvore informacije relevantne za određenu temu
- Vrste odnosa (engl. *association*): predstavljaju veze među temama

Slika (Slika 4.2) prikazuje temeljni model tematskih mapa s opisanim elementima (engl. *TAO of Topic Maps – types, occurrences, associations*).



Slika 4.2 Komponente tematske mape

Jedan od poznatijih alata za kreiranje, prikaz i pretraživanje topic mapa je The Omnigator, dostupan na <http://www.ontopia.net/download.jsp>

4.2. Mentalne mape

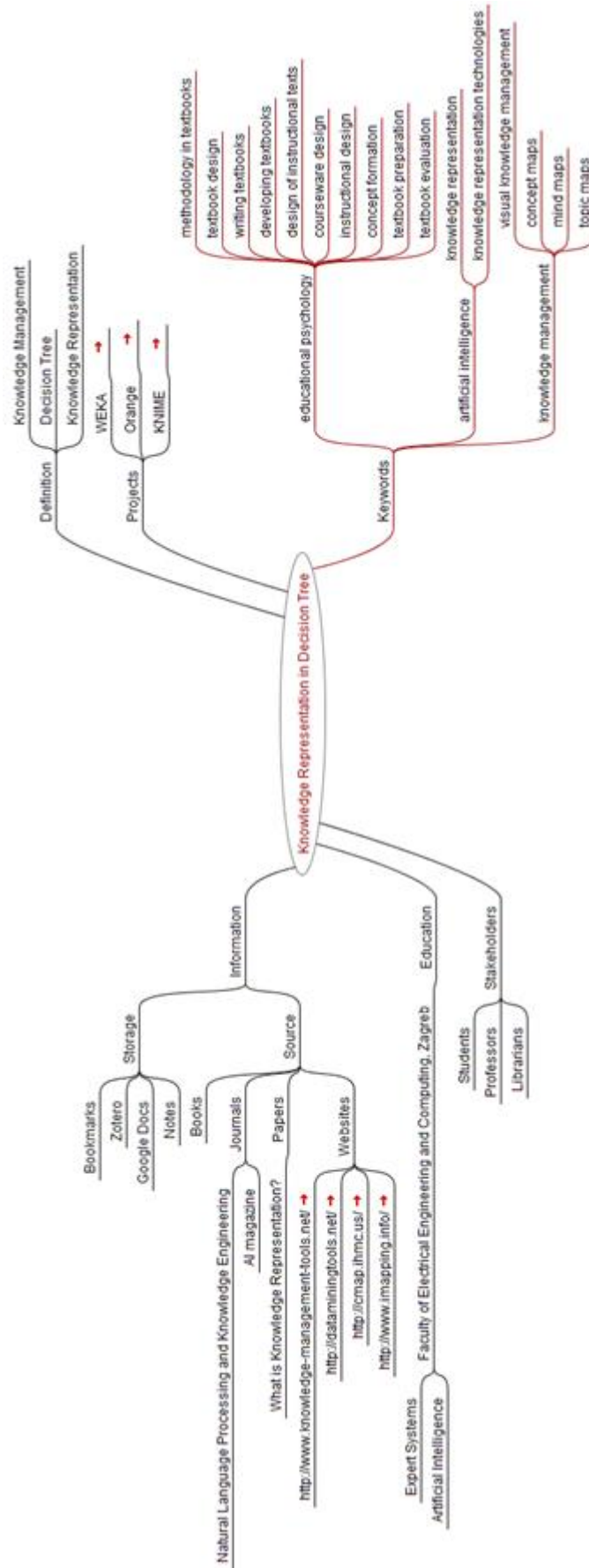
Mentalne mape ili mape uma su vrsta konceptualnih mapa sa sljedećim karakteristikama (Informationtainers, 2009):

- glavni pojam, ideja ili fokus je prikazan u centru slike
- glavne teme izlaze i granaju se kao ogranci iz centralne slike
- ogranci sadrže ključnu sliku ili ključnu riječ koja je napisana na pridruženoj liniji
- teme od manje važnosti reprezentiraju se kao izdanci važnijih grana
- grane formiraju povezanu strukturu stabla

Tvorcem modernih mentalnih mapa smatra se britanski psiholog Tony Buzan. U svom vodiču, u kojem govori kako načiniti mentalnu mapu koristeći neka osnovna pravila, daje sljedeće savjete:

1. počnite od sredine nekim crtežom koji karakterizira temu na kojoj radite
2. koristite razne simbole i različite veličine slova u cijelom dijagramu
3. odredite i upišite ključne riječi
4. svaka ključna riječ mora biti samostalno zapisana
5. značenje na linijama koje izlaze iz ključne riječi mora se odnositi prema značenju ključne riječi
6. dužina linije, iznad koje pišete riječi, neka bude jednaka dužini riječi
7. koristite različite boje
8. razvijte svoj stil izrade mentalne mape
9. naglasite najvažnije dijelove i koristite asocijacije
10. nacrt mentalne mape napravite jasno u zrakastoj strukturi

Sljedeća slika (Slika 4.3) prikazuje primjer mentalne mape, napravljene prilikom istraživanja područja kroz ovaj rad.



Slika 4.3 Mentalna mapa Prezentacija znanja

5. Aplikacija Decision Tree Resolver

U ovom radu naglasak je stavljen na stabla odlučivanja i konceptualne mape kao relativno intuitivne alate za prikaz znanja. Istraživanjem dostupnih alata za izradu stabla odlučivanja uočen je nedostatak jednostavnih *online* alata. Alat za izradu konceptualnih mapa IHMC CmapTools, opisan u nastavku, izdvojen je kao kvalitetan i intuitivan alat koji omogućuje pohranu izrađenih mapa na zajedničke servere i na taj način zajedničku suradnju više ljudi na izradi jedne mape. Izrađena aplikacija Decision Tree Resolver pretvara konceptualne mape izrađene alatom IHMC CmapTools u *online* stabla odlučivanja koja omogućuju korisniku jednostavno kretanje po stablu i na kraju, donošenje odluke.

U nastavku su ukratko opisane tehnologije korištene pri izradi aplikacije (5.1), realizacija same aplikacije i njene funkcionalnosti (5.2) te neke od mogućnosti nadogradnje (5.3).

5.1. Odabrane tehnologije

Za izradu konceptualnih mapa odabran je alat IHMC CmapTools (5.1.1), za izradu same web aplikacije programski jezika Java na platformi Java EE za izradu web aplikacija (5.1.2) te HTML i CSS za izradu i dizajn web stranica (5.1.3 i 5.1.4).

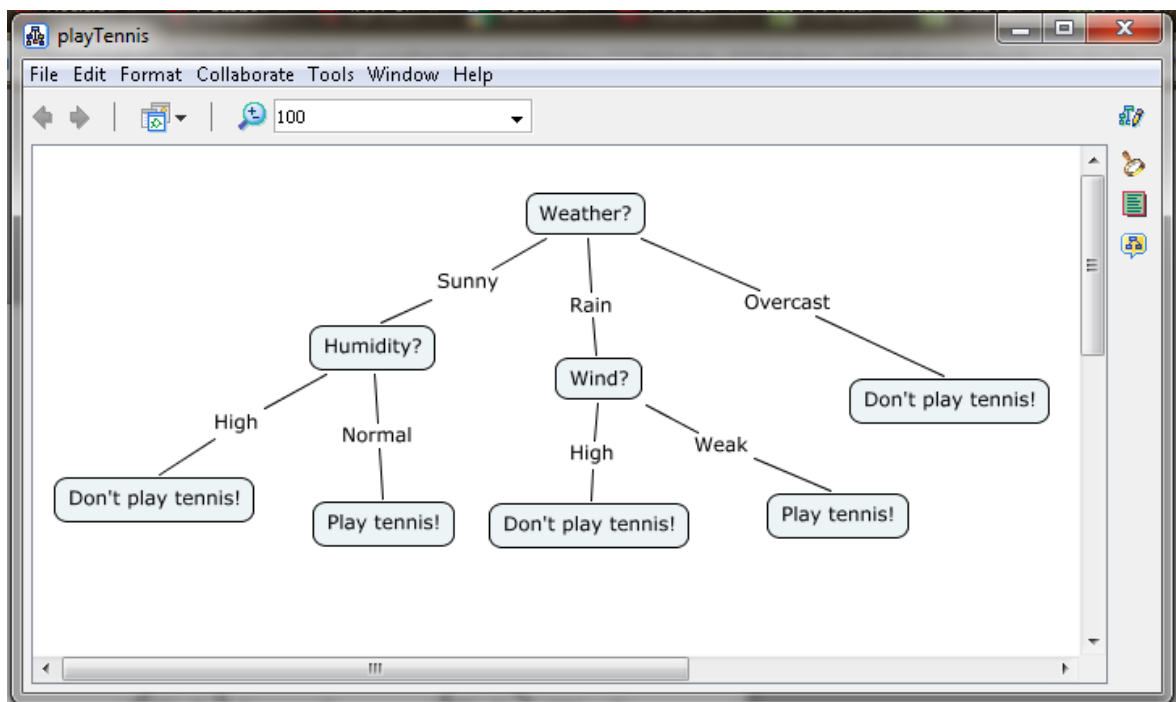
5.1.1. IHMC CmapTools

IHMC CmapTools je alat razvijen na IHMC institutu (Florida Institute for Human and Machine Cognition) čija područja djelatnosti obuhvaćaju prezentaciju znanja u formi pogodnoj za računalnu obradu, istraživanje i razvijanje standarada za razvoj semantičkog Interneta - Interneta koji pruža bogatiji i bolje povezani sadržaj.

IHMC CmapTools služi za izradu sadržajno bogatih konceptualnih mapa jer je svaki entitet u izrađenoj konceptualnoj mapi moguće dodatno pojasniti podatkovnom poveznicom na novu konceptualnu mapu, sliku, video ili dokument. Korisnici mogu spremati svoje mape na vlastitim računalima ili na Cmap serverima te tako dijeliti svoje mape s ostalim korisnicima. Pri svakom spremanju mape automatski se generira njena web stranica na nekom od servera. Dostupna je i stranica koja omogućava pretraživanje svih javno dostupnih konceptualnih mapa i sadržaja o njima na webu: <http://www.cmappers.net/>.

Alat nudi mogućnost izvođenja podataka u različite formate poput PNG, PDF, HTML, CXL, XTM/XCM, IVML te *Propositions as Text* formata. Upravo je *Propositions as Text* format koji će aplikacija Decision Tree Resolver koristiti pri pretvaranju konceptualne mape u stablo odlučivanja.

Na slici je prikazana konceptualna mapa koja odlučuje treba li igrati tenis ili ne, a na sljedećoj slici je prikaz iste mape u *Propositions as Text* formatu. Svaki red tog formata prikazuje jednu propoziciju (engl. *proposition*) koju sačinjava trojka koncept, veza, koncept.



Slika 5.1 Konceptualna mapa Play tennis

Tablica 5.1 *Propositions as Text* format mape Play tenis

Weather?	Rain	Wind?
Wind?	Weak	Play tennis!
Humidity?	High	Don't play tennis!
Weather?	Sunny	Humidity?
Weather?	Overcast	Don't play tennis!
Humidity?	Normal	Play tennis!
Wind?	High	Don't play tennis!

5.1.2. Java

Java je programski jezik kojeg je razvio James Gosling za Sun Microsystems, a koji je službeno pušten u korištenje 1995. godine. Sintaksu nasljeđuje od programskih jezika C i C++. Za izvođenje Java programa na računalu je potrebno imati instaliran Java virtualni stroj (engl. *Java Virtual Machine*). Java je objektno-orientirani jezik namijenjen za općenitu uporabu. Napisani kôd koji se izvršava na jednoj platformi, ne mora biti ponovno preveden (engl. *compile*) da bi se uspješno izvršavao na drugoj platformi pa vrijedi sintagma „napiši jednom, izvodi bilo gdje“ (engl. *write once, run anywhere*). Java je trenutno jedan od najpopularnijih programskih jezika (Wikipedija).

Za izradu ove aplikacije korištena je Java na platformi Java EE (engl. *Java Enterprise Edition*), što je platforma specijalizirana za izradu web aplikacija. Korištena je tehnologija JSP (engl. *Java Server Pages*) koja omogućuje izradu dinamički generiranih web stranica temeljenih na HTML-u, XML-u ili drugim tehnologijama.

5.1.3. HTML

HTML je prezentacijski jezik za izradu web stranica. Hipertekst dokument stvara se pomoću HTML jezika. HTML jezikom oblikuje se sadržaj i stvaraju se hiperveze hipertekst dokumenta. HTML je jednostavan za uporabu i lako se uči, što je jedan od razloga njegove opće prihvaćenosti i popularnosti. Svoju raširenost zahvaljuje jednostavnosti i tome što je od početka bio zamišljen kao besplatan i tako dostupan svima. Prikaz hipertekst dokumenta omogućuje web preglednik. Temeljna zadaća HTML jezika jest uputiti web preglednik kako prikazati hipertekst dokument. Pri tome se nastoji da taj dokument izgleda jednako bez obzira o kojemu je web pregledniku, računalu i operacijskom sustavu riječ. Osnovni građevni element svake stranice su znakovi (engl. *tags*) koji opisuju kako će se nešto prikazati u web pregledniku. Povezice unutar HTML dokumenata povezuju dokumente u uređenu hijerarhijsku strukturu i time određuju način na koji posjetitelj doživljava sadržaj stranica (Wikipedija).

HTML je u aplikaciji korišten za izradu statičkih web stranica, dok se dinamičnost stranica ostvaruje Java dijelovima koda. CSS tehnologija služi za dizajn web stranica i opisan je u nastavku.

5.1.4. CSS

CSS (engl. *Cascading Style Sheets*) je stilski jezik koji rabimo za opis prezentacije dokumenta napisanog pomoću jezika za označavanje (HTML). CSS je stvoren kako bi se omogućilo odvajanje dizajna stranice (oblik i veličina slova, boje, pozicioniranje, izgled slika i tablica, pozadina) od sadržaja same stranice napisanog u nekom od jezika za označavanje (npr. HTML). Ovakva odvojenost pruža veću fleksibilnost i kontrolu nad karakteristikama prikaza te omogućuje većem broju stranica da imaju isti dizajn bez da se svaka posebno oblikuje (Wikipedija).

5.2. Realizacija

Početna stranica aplikacije Decision Tree Resolver izgleda kao na sljedećoj slici (Slika 5.2):

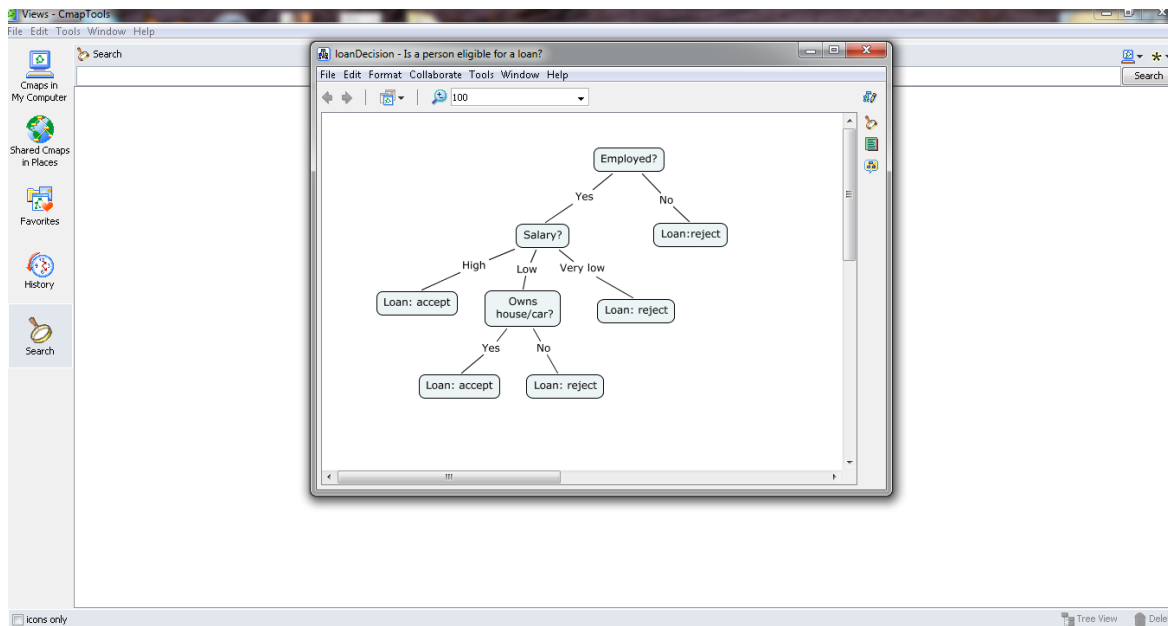


Slika 5.2 Decision Tree Resolver početna stranica

Početna stranica nudi kratak opis aplikacije u prvom odlomku, drugi odlomak sadrži podatkovne poveznice na upute za korištenje i primjere, dok je zadnji odlomak forma za dohvaćanje konceptualne mape u *Propositions as Text* formatu.

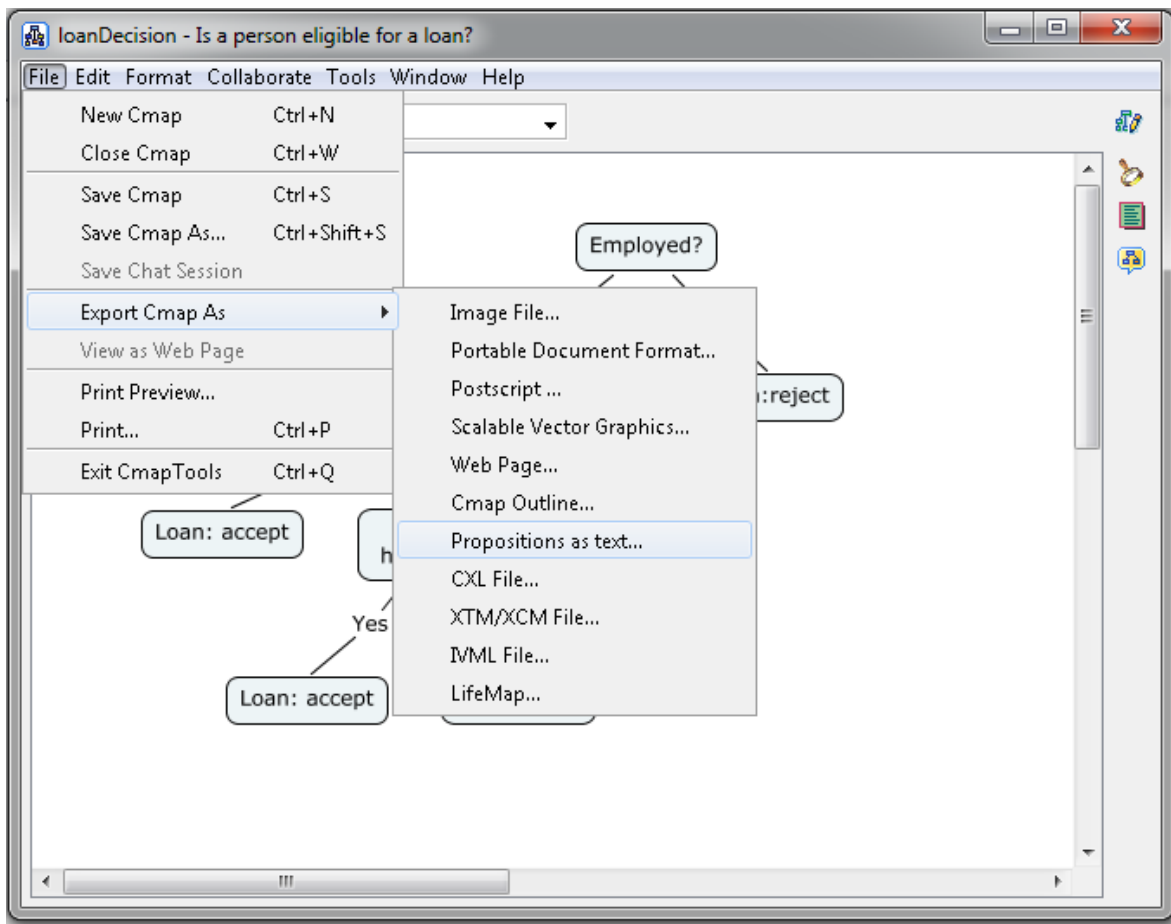
Podatkovna poveznica *Instructions* vodi na stranicu s uputama za korištenje koje su opisane u četiri koraka:

1. Potrebno je sagraditi konceptualnu mapu koristeći alat IHMC CmapTools (Slika 5.3)



Slika 5.3 Korak 1

2. Izgrađenu mapu treba izvesti u *Propositions as Text* format (Slika 5.4)



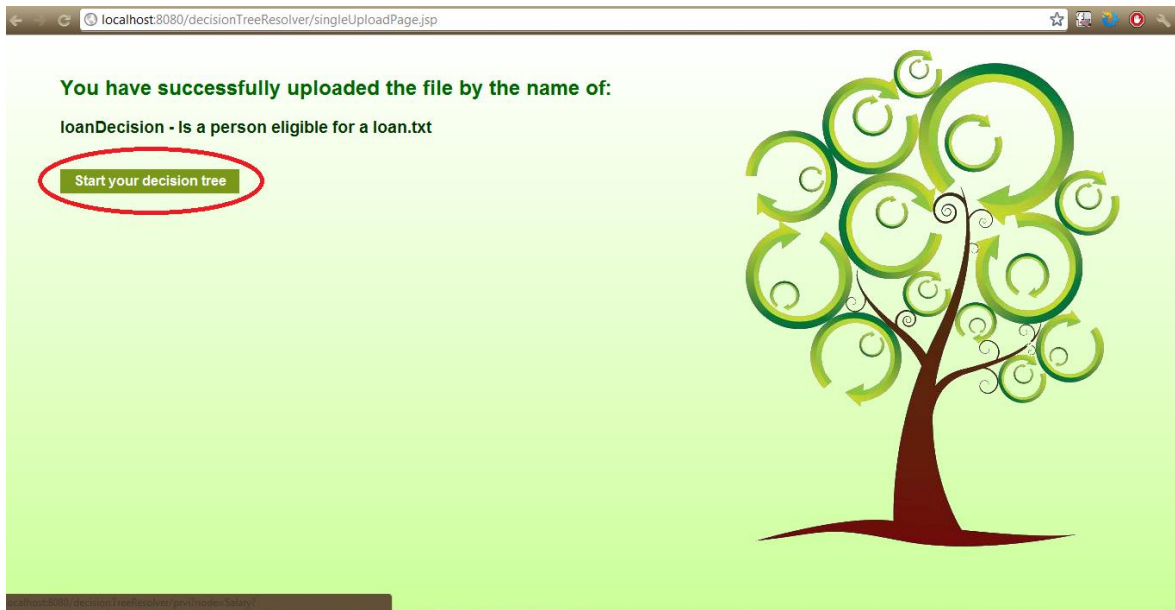
Slika 5.4 Korak 2

3. Zatim korisnik otvara url Decision Tree Resolver aplikacije te joj predaje svoje konceptualnu mapu (Slika 5.5)



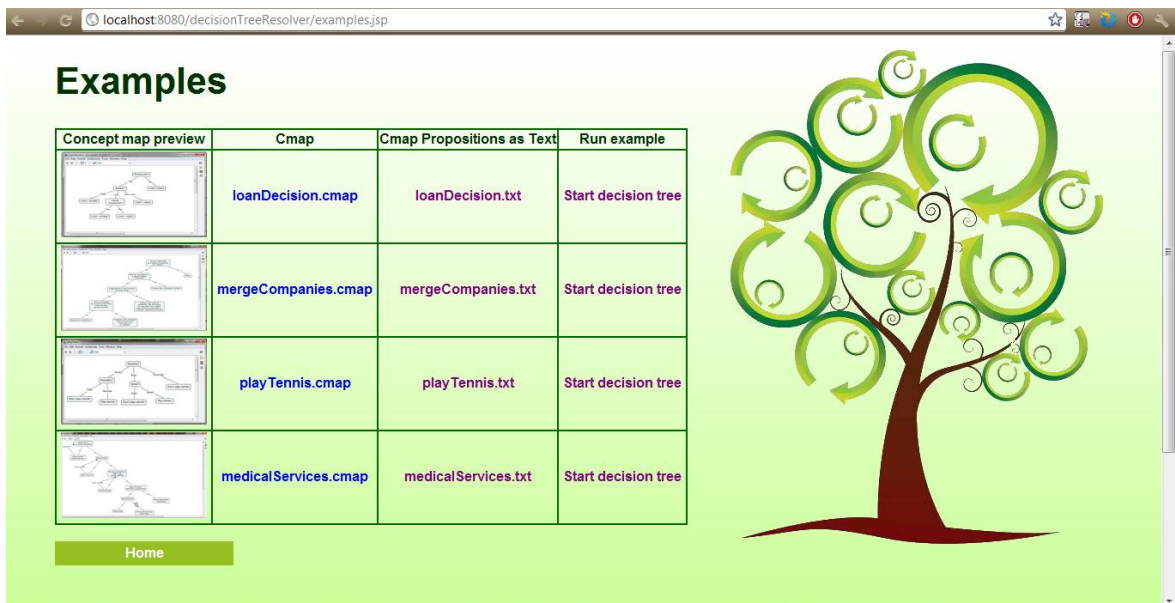
Slika 5.5 Korak 3

4. Nakon što je korisnikova mapa dohvaćena, korisnik slijedi aplikaciju koja mu pokazuje njegovo stablo (Slika 5.6)



Slika 5.6 Korak 4

Podatkovna stranica *Examples* (Slika 5.7) vodi na stranicu gdje su tabelarno prikazani primjeri na način da je dostupna slika konceptualne mape, sama mapa, mapa u *Propositions as Text* formatu i mogućnost isprobavanja izgrađenog stabla.



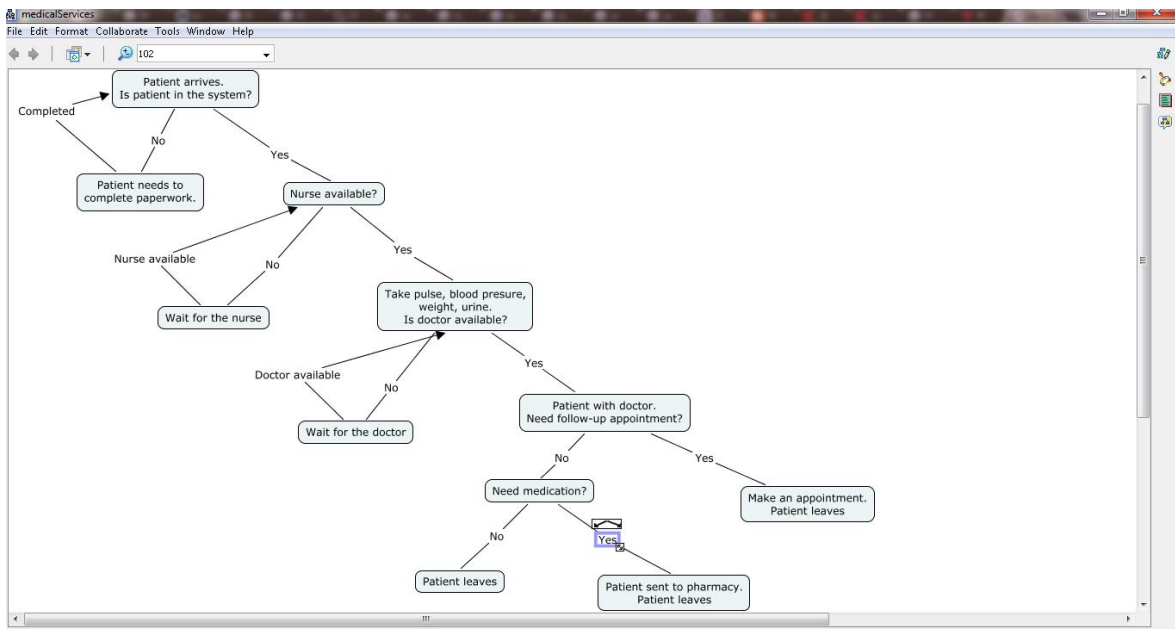
Slika 5.7 *Examples* stranica

Izgrađeno stablo odlučivanja se sastoji od pitanja (uvjeta) te odgovora koji su dani u obliku podatkovnih poveznica koje onda vode na novo pitanje ili na stranicu s krajnjom odlukom. Primjer dijela stabla odlučivanja dan je sljedećom slikom (Slika 5.8).



Slika 5.8 Stablo odlučivanja

Valja napomenuti da aplikacija podržava, osim gradnje jednostavnih stabala odlučivanja, i gradnju stabala odlučivanja s povratnim vezama. Takva se stabla onda nazivaju dijagramima toka (engl. *flow charts*). Primjer takvog dijagrama prikazan je na sljedećoj slici (Slika 5.9), a simulira posjet pacijenta liječniku.



Slika 5.9 Primjer stabla s povratnim vezama

5.3. Mogućnosti nadogradnje

Aplikacija Decision Tree Resolver pruža osnovne funkcionalnosti pretvorbe konceptualne mape u *online* stablo odlučivanja. No, bilo bi poželjno da se aplikacija dalje razvija i nadograđuje. Neki od prijedloga za nadogradnju su sljedeći:

- Korištenje aplikacije za provjeru znanja na način da se testira valjanost izgrađene mape kroz testne slučajeve koji bi provjerili dobije li se očekivani rezultat
- Korištenje nekog drugog, sadržajem bogatijeg, izvedenog formata za izradu stabla odlučivanja
- Izrada stranice koja će sadržavati sve do tog trenutka dohvaćene konceptualne mape; time bi se osigurala velika baza primjera
- Analiziranje dokumenta (engl. *log file*) u kojem su spremljeni svi korisnikovi koraci pri odlučivanju; time bi se omogućilo prepravljanje i poboljšanje izgrađenih stabala

Zaključak

Problem kvalitetnog i intuitivnog formaliziranja i ekspliciranja znanja oduvijek je zanimao različite znanstvene discipline, od logike i filozofije do psihologije i pedagogije. Razvojem računalne znanosti, prezentacija znanja postaje i jedno od područja istraživanja u umjetnoj inteligenciji. Umjetna inteligencija razlikuje četiri načina prezentacije znanja: prezentacija znanja logikom, proceduralna prezentacija znanja, mrežna i strukturalna prezentacija.

Stabla odlučivanja se koriste za grafički prikaz rada ekspertnih sustava koji su predstavnici proceduralne prezentacije znanja. Stabla odlučivanja predstavljaju intuitivan prikaz određenog dijela znanja, a sastoje se od čvorova odluka, čvorova vjerojatnosti određenih događaja i konačnih ishoda ili odluka. S druge strane, konceptualne mape kao predstavnici mrežne prezentacije znanja, prikazuju znanje kao mrežu međusobno povezanih pojmova. U posebnim slučajevima, alati za izradu konceptualnih mapa mogu se koristiti i za izradu stabla odlučivanja.

Uviđajući jednostavnost korištenja konceptualnih mapa, ali i nedostatak *online* alata za izradu stabla odlučivanja, izrađena je jednostavna aplikacija Decision Tree Resolver. Ova aplikacija prevodi konceptualnu mapu izrađenu alatom IHMC CmapTools u stablo odlučivanja čiji čvorovi odluke postaju pitanja, a čvorovi vjerojatnosti podatkovne poveznice. Ovakva aplikacija bi trebala biti intuitivna za korištenje i korisnicima s minimalnim informatičkim vještinama. Brojne su mogućnosti nadogradnje aplikacije, od implementiranja podrške za rad s drugim vrstama dokumenata i alata do korištenja aplikacije za provjeru znanja.

Literatura

- [1] PLESLIĆ S., Uvod u Upravljanje Znanjem, *Predavanje 1*. Fakultet Elektrotehnike i Računarstva, Sveučilište u Zagrebu, 2012.
- [2] FROST R.A., Introduction to Knowledge Base Systems. Collins, 1986.
- [3] ZELENY, M., Management Support Systems: Towards Integrated Knowledge Management. *Human Systems Mangement*, 1987, 7, 1, 59-70.
- [4] CLEVELAND H., Information as Resource. *The Futurist*, December 1982, 34-39.
- [5] PRGOMET M., Neformalni Oblici Prikaza Znanja U Internet Okruženju. *Diplomski rad*. Prirodoslovno Matematički Fakultet, Sveučilište u Splitu, 2011.
- [6] NONAKA, I; TAKEUCHI, H., The knowledge creating company: how Japanese companies create the dynamics of innovation. New York: Oxford University Press, 1995, 284.
- [7] WILDE, S., Customer Knowledge Management. Berlin:Springer, 2011, 20-22.
- [8] GARCÍA R., A Semantic Web Approach to Digital Rights Management. Doctoral thesis. PhD Thesis, Universitat Pompeu Fabra, Barcelona, Španjolska, 2006.
- [9] SOWA, J.F., Knowledge Representation. Logical, philosophical and computational foundations. Brooks Cole Publishing Co., 2000.
- [10] DAVIS R., SHROBE H., SZOLOVITS P., What is a Knowledge Representation? *AI Magazine*, 14(1):17-33, 1993.
- [11] ARMBRUSTER BONNIE B., Schema Theory and the Design of Content-Area Textbooks. *Educational Psychologist*, 21, 4, 1986, 253-267.
- [12] JONASSEN DAVID H., Hypertext Principles for text and Courseware Design. *Educational Psychologist*, 21, 4, 1986, 269-292.
- [13] NAZAROVA, T. S., GOSPODARIK, IU. P., Strategy for the Development of the Textbook. *Russian Education and Society*, 48 , 6, Jun 2006, 86-102 .
- [14] KERBER M., Intro to AI *Lecture*, 2004/05
- [15] REICHGELT H., Knowledge representation: an AI perspective. Ablex Pub. Corp., 1991
- [16] PFENNING F., Logic Programming. *Lecture 1*, Carnegie Mellon University, 2006.
- [17] DALBELO-BAŠIĆ B., Umjetna Inteligencija, *Predavanje 8*, Fakultet Elektrotehnike i Računarstva, Sveučilište u Zagrebu, 2012.
- [18] CLIPS, <http://clipsrules.sourceforge.net/>, dokumentacija alata, 10. lipnja 2012
- [19] MINSKY M., A Framework for Representing Knowledge. MIT-AI Laboratory Memo 306, June, 1974.
- [20] LUGER, GEORGE F., STUBBLEFIELD, WILLIAM A. Artificial intelligence: Structures and strategies for complex problem solving. Addison-Wesley. 1998.

- [21] SCHANK ROGER C., ABELSON ROBERT P. Scripts, Plans, Goals and Understanding: an Inquiry into Human Knowledge Structures. L. Erlbaum, Hillsdale, NJ, 1977.
- [22] GAMBERGER D., ŠMUC T., MARIĆ I. Data Mining Server Project. Institut R. Bošković, 2001.
- [23] RAHMAN T. An efficient visual classification based approach to Decision Tree construction. 14th International Conference on Advanced Communication Technology (ICACT), 2012 .
- [24] MIHALIĆ T., Organizacija znanja: tematske mape i glazba. Zbirka muzikalija i audiomaterijala, Nacionalna i sveučilišna knjižnica u Zagrebu, 2008.
- [25] NOVAK JOSEPH D., CAÑAS ALBERTO J. The Theory Underlying Concept Maps and How to Construct and Use Them. Technical Report IHMC CmapTools 2006-01 Rev 2008-01
- [26] INFORMATIONTAMERS,
http://www.informationtamers.com/WikIT/index.php?title=Mind_maps, 2009, 10. lipnja 2012
- [27] WIKIPEDIJA, Java (programming language),
[http://en.wikipedia.org/wiki/Java_\(programming_language\)](http://en.wikipedia.org/wiki/Java_(programming_language)), 18. lipnja 2012.
- [28] WIKIPEDIJA, HTML, <http://hr.wikipedia.org/wiki/HTML>, 18. lipnja 2012.
- [29] WIKIPEDIJA, CSS, http://en.wikipedia.org/wiki/Cascading_Style_Sheets, 18. lipnja 2012.

Sažetak

Znanje je potrebno formalizirati kako bi se omogućilo njegovo lakše dijeljenje i korištenje u računalnim sustavima. Različite discipline se bave problemom prezentacije znanja. Umjetna inteligencija nudi više načina prezentacije znanja koji su detaljno opisani u radu. Rad se usmjerio na koncepte stabla odlučivanja i konceptualne mape, kao jedne od najintuitivnijih i najjednostavnijih prikaza. U sklopu rada je napravljena web aplikacija Decision Tree Resolver koja prihvaća konceptualnu mapu izrađenu IHMC CmapTools alatom i pretvara ju u online stablo odlučivanja. Dan je pregled aplikacije i njenih funkcionalnosti kao i prijedlozi za daljnju nadogradnju.

Ključne riječi: znanja, prezentacija znanja, stablo odlučivanja, konceptualne mape

Summary

Knowledge needs to be formalised in order to allow easier sharing and to enable its efficient use in computer systems. Different disciplines address the problem of knowledge representation. Artificial intelligence describes several knowledge representation techniques, which are discussed in this thesis. This thesis focuses on the concepts of decision trees and conceptual maps as one of the easiest and most intuitive knowledge representation techniques. Decision Tree Resolver is a web application developed as a part of this thesis. It takes a concept map made with IHMC CmapTools tool as input and transforms it into the online decision tree. This thesis provides an overview of the application and its functionality as well as instructions for further development.

Key words: knowledge, knowledge representation, decision tree, concept map

Skraćenice

JVM *Java Virtual Machine*

HTML *HyperText Markup Language*

CSS *Cascading Style Sheets*

Java virtualni stroj

hipertekstualni jezik za označavanje

kaskadni dizajn stranica

Privitak

U privitku će ukratko biti opisano kako instalirati i koristiti izgrađenu Decision Tree Resolver aplikaciju.

Instalacija programske podrške

Potrebno je:

1. Instalirati Apache-tomcat-7.0.27 server
2. Instalirati Java virtualni stroj verzije JVM 1.7.0_01-b08 ili novije
3. Na stranici <http://localhost:8080/manager/> odabrati WAR datoteku u kojoj je aplikacija te ju pokrenuti

Upute za korištenje programske podrške

Potrebno je:

1. Otići na web stranicu aplikacije <http://localhost:8080/decisionTreeResolver/>
2. Učitati svoju konceptualnu mapu u aplikaciju te slijediti izgrađeno stablo odlučivanja do konačne odluke