

UNIVERZITET CRNE GORE
Elektrotehnički fakultet, Podgorica

Materijal sa jedanaestog termina predavanja iz

EKSPERTNIH SISTEMA

PREDSTAVLJANJE NEFORMALNOG ZNANJA
SEMANTIČKE MREŽE I OKVIRI

Prof. dr Vesna Popović-Bugarin

Podgorica, 2013.

PREDSTAVLJANJE NEFORMALNOG ZNANJA

Alternativan pristup formalnom predavljanju znanja, jeste neformalno predavljanje znanja. Neformalno predavljanje znanja je strukturano, odnosno modularno. Ovakav model predavljanja znanja ima veliki broj prednosti u poređenju sa formalnim, [4]:

1. Baza znanja se može na jednostavan način predstaviti u modularnom obliku, čime se obezbijuje sakupljanje povezanih grupa znanja radi lakšeg pristupa.
2. Praćenje toka zaključivanja (u cilju dobijanja objašnjenja, odnosno obrazloženja donijetih zaključaka) postaje mnogo lakše u BZ sa strukturanim modelom.
3. Strukturani model čini mogućim predavljanje monotone, nemonotone i podrazumijevane logike.
4. Fragmentiranim djelovima znanja se može konkurentno pristupiti iz više modula, čime se modulu za zaključivanje obezbijuje okvir za dijeljenje resursa. Ovakav način dijeljenja resursa zahtijeva minimalnu strukturu i samim tim manje hardverskih/softverskih uređaja za realizaciju baze znanja.
5. Više modula strukturanih modela može biti aktivno istovremeno, čime se obezbijuje prostor za masovni paralelizam u procesu zaključivanja. Samim tim se vremenska efikasnost modula za zaključivanje može poboljšati sa strukturanim modelima.

U ovom poglavlju će biti proučavane semantičke mreže i okviri, kao modeli strukturiranog predavljanja znanja.

5.1 Semantičke mreže

Semantičke mreže su se prvi put primijenile u cilju modelovanja ljudske memorije i za razumijevanje prirodnih jezika, gdje je semantika (značenje) riječi povezanih u rečenici izvlačena korišćenjem ovakvih mreža. Stoga je motivacija za upotrebu semantičkih mreža zapravo želja da se nađe forma predavljanja znanja kojom bi se omogućilo smještanje i „značenja“ riječi, tako da se upotreba ovih značenja vrši na način kako to čine ljudska bića. Vremenom su semantičke mreže našle širu primjenu u zaključivanju u sistemima zasnovanim na znanju [4]. **Semantičke mreže se koriste za predavljanje propozicionih tvrdnji, tj. tvrdnji koje mogu biti ili tačne ili netačne.**

Postoji mnogo varijanti semantičkih mreža, ali sve one mogu predstaviti individualne objekte, kategorije (klase) objekata i relacije među objektima i/ili klasama. U matematičkom smislu, semantičke mreže su označeni usmjereni grafovi. Tipična grafička notacija prikazuje imena objekata ili kategorija u **ovalnim oblicima ili box-ovima** (čvorovima), i povezuje ih označenim lukovima koji predstavljaju relacije među njima. Dakle, semantičke mreže se sastoje od elementarnih parova: **objekata (dogadaja, koncepata) predavljenih čvorovima i relacija među njima, predavljenih označenim lukovima, strelicama.**

Jednostavna semantička mreža predavlja predikat: $SVIDJA(x,y)$:



Semantičkim mrežama se mogu predstaviti, ne samo binarni predikati (predikati koji imaju dva argumenta), već i nebinarni predikati. Posmatrajmo tako unarni predikat $DJECAK(MARKO)$. Ovo može biti predstavljeno binarnim predikatom $IS_AN_INSTANCE_OF(MARKO, DJECAK)$, što odgovara logičkom tvrđenju $MARKO \in DJECAK$.



Ovdje predikat postaje klasa-kategorija povezana sa svojom instancom-objektom relacijom **IS_AN_INSTANCE_OF**.

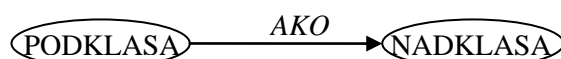
Određena vrsta relacija, kao što je prethodno ilustrovana, se pokazala kao veoma korisna u predstavljanju širokog spektra znanja. S toga se u semantičkim mrežama uobičajeno koriste ovi standardni tipovi veza, a rjeđe se definišu nove relacije. Korišćenjem standardnih tipova relacija omogućava se lakše razumijevanje nepoznatih semantičkih mreža od strane velikog broja ljudi.

Dvije vrste najčešće korišćenih relacija (linkova) su **IS-A** i **A-KIND-OF**, koje se uobičajeno pišu kao **ISA** i **AKO**.

- **ISA** je akronim od *is an instance of*. Odnosi se na vezu specifičnog primjerka (objekta) neke klase sa klasom kojoj taj objekat pripada, kao u prethodnom primjeru. Ovdje je klasa povezana sa matematičkim konceptom skupa, s obzirom da se odnosi na skup objekata. Iako skup može imati elemente bilo kojeg tipa, objekti u skupu imaju neku međusobnu vezu, odnosno zajedničke karakteristike kojima se jasno definiše njihova pripadnost datom skupu (kao u objektu orijentisanom programiranju). Koncept klase i **ISA** veza se može iskoristiti da se predstavi i situacija, akcija i/ili događaj. Link **ISA** je usmjeren od objekta ka klasi kojoj taj objekat pripada.



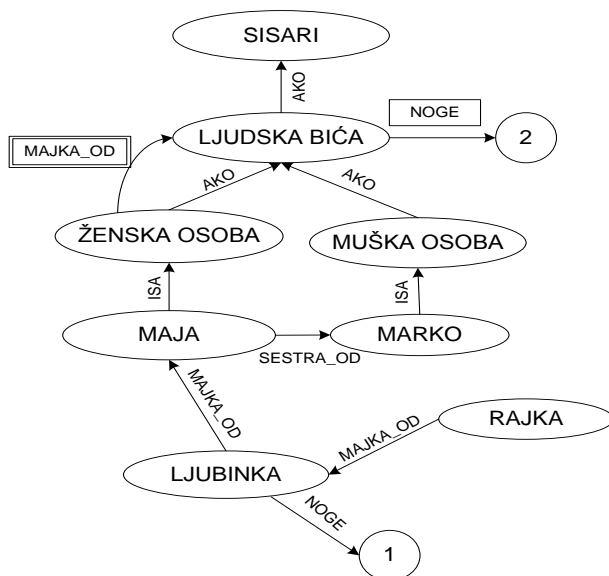
- Link **AKO** je akronim od *a kind of*, i koristi se za povezivanje jedne klase sa drugom. **AKO** se ne koristi za povezivanje specifičnih instanci (objekata) klase sa klasom, jer je to funkcija relacije **ISA**. Dakle, **AKO** povezuje pojedinačnu klasu sa roditeljskom klasom iz koje je posmatrana klasa izvedena. Može se reći i da **AKO** povezuje generički čvor (klasu), sa drugim generičkim čvorovima (klasama), dok **ISA** povezuje instancu (objekat) sa generičkim čvorom (klasom). **AKO** je usmjeren od podklase ka nadklasi (od izvedene klase ka roditeljskoj klasi).



I druge vrste linkova se mogu naći u semantičkim mrežama. Na primjer, **PARTOF** se koristi za označavanje da je neki čvor (klasa, objekat, događaj, ...) dio nekog drugog čvora. Naravno, i kreator klase može definisati nove linkove koji predstavljaju relaciju između dva objekta, dvije klase ili objekta i klase. Ovakvi

linkovi su predstavljeni strelicom sa imenom relacije. Ipak, na ovaj način se smanjuje mogućnost razumijevanja semantičke mreže od strane drugih ljudi. Na slici 1 je prikazan primjer semantičke mreže, kojom je predstavljena relacija članova unutar jedne porodice, kao i pripadnost pojedinih objekata opštim klasama.

Linkovi, odnosno relacije, predstavljene označenim strelicama, od izuzetnog su značaja u semantičkim mrežama jer obezbjeđuju osnovnu strukturu za organizaciju znanja, slika 1. Bez relacija, znanje je obična kolekcija nepovezanih činjenica. Sa relacijama, znanje je kohezivna struktura na osnovu koje se može izvoditi novo znanje. Na primjer, na osnovu semantičke mreže prikazane na slici 1, može se zaključiti, iako nema eksplicitne relacije označene sa BABA_OD, da je Rajka Majina baba.



Slika 1 Primjer semantičke mreže sa standardizovanim i izvedenim linkovima

Prilikom uspostavljanja veza među klasama, neophodno je jasno razdvojiti klase i njihove osobine, od osobina instanci (objekata) – specifičnog primjerka klase. Na primjer, ukoliko znamo da ljudska bića imaju ženske osobe kao majke, postavlja se pitanje da li se smije nacrtati veza MAJKA_OD od ŽENSKA_OSOBA ka LJUDSKA_BIĆA, slika 1. Odgovor je negativan, jer je MAJKA_OD relacija između određene osobe i njene majke, a klase kao cjeline, nemaju majku. Dakle, za svaki objekat klase ljudsko biće važi relacija da mu je majka ženska osoba (objekat druge klase). Stoga se ova relacija mora posmatrati kao:

$$\forall x (x \in LJUDSKA_BIĆA \Rightarrow (\forall y MAJKA_OD(y,x) \Rightarrow y \in ŽENSKA_OSOBA))$$

Ovo se može zapisati i kao:

$$\forall x x \in LJUDSKA_BIĆA \Rightarrow \exists y (MAJKA_OD(y,x) \wedge y \in ŽENSKA_OSOBA)$$

Na isti način bi zaključak da osoba ima dvije noge pisali kao:

$$\forall x x \in LJUDSKA_BIĆA \Rightarrow NOGE(x,2)$$

Dakle, i ovdje moramo biti pažljivi da ne zaključimo da kategorija – klasa ima dvije noge, već da za svaki element klase ljudsko biće, važi relacija NOGE koja ga povezuje sa objektom 2.

Jasno je da je jako bitno da semantičke mreže budu kompatibilne u predstavljanju kvantifikatora. Kada je riječ o kvantifikatorima u semantičkim

mrežama, uobičajeno se definišu i sljedeća dva tipa veza [1], kako bi se zadržalo njihovo značenje, a mreža bila lako čitljiva:

- Relacija između svakog elementa klase A i objekta B (obilježava se uokvirenim imenom relacije);
- Relacija između svakog elementa klase A i nekog elementa klase B (ime relacije dvostruko uokvireno).

Ilustracija prethodne dvije relacije je prikazana na slici 1. Relacija NOGE povezuje svaki element klase LJUDSKA BIĆA sa objektom 2. Dok relacija MAJKA OD predstavlja relaciju svakog elementa klase LJUDSKA BIĆA sa nekim elementom klase ŽENSKA OSOBA.

5.1.1 Nasljeđivanje u semantičkim mrežama

Binarna relacija $x < y$ je djelimično uređena ako: i) $x < x$ (t.j. $x < x$ je bezuslovno neistinito) i ii) kada je $x < y$ i $y < z$, imamo $x < z$, t.j. tranzitivna veza važi. Ukoliko je koncept u semantičkim mrežama djelimično uređen, takvu mrežu nazivamo sistemom nasljeđivanja [4]. Ipak, prilikom crtanja mreža, uvijek se izostavlja luk koji predstavlja tranzitivno nasljeđivanje, t.j. postoji usmjereni link od čvora x ka čvoru y i od čvora y ka čvoru z , dok je link od čvora x ka čvoru z očigledan i može biti izostavljen.

Zahvaljujući notaciji koju koriste, semantičke mreže su veoma pogodne za **zaključivanje pomoću nasljeđivanja**, gdje nasljeđivanje omogućava pomjeranje opisa sa klase na instance (uzorke) ili podklase. Na primjer, činjenicom da je *ženska osoba*, koja pripada klasi *ljudskih bića*, Maja nasljeđuje svojstvo da ima dvije noge, slika 1. Stoga, da bi se utvrdilo koliko nogu ima Maja, algoritam nasljeđivanja prati *ISA* link od Maje ka kategoriji kojoj ona pripada, zatim prati *AKO* link hijerarhijski na gore, dok ne pronađe kategoriju za koju postoji uokviren link NOGE (karakteristika svakog objekta klase) – u ovom slučaju, kategorija LJUDSKA BIĆA. Ukoliko ne postoji specifičniji dokaz koji tvrdi suprotno, nasljeđivanjem svi članovi klasa nasljeđuju sva svojstva svojih roditeljskih klasa, odnosno svi objekti nasljeđuju svojstva klase čije su instance.

Procedura nasljeđivanja se može predstaviti algoritmom [3]. Neka je F dati čvor, a S data relacija (link) (ilustrovaćemo ga kroz $F=MAJA$, $S=NOGE$, slika 1). Interesuje nas vrijednost relacije S za objekat F . Vrijednost relacije je objekat ili klasa na koju ukazuje:

- Korak 1. Sačiniti red čekanja koji se sastoji od čvora F i svih ostalih čvorova klase koji su sa čvorom F povezani *ISA* relacijom; F treba da bude na čelu (red čekanja: *MAJA*, *ŽENSKA OSOBA*);
- Korak 2. Dok red čekanja ne postane prazan ili dok se ne nađe relacija S , odrediti da li prvi element reda čekanja ima relaciju S :
 - a. Ako prvi element ima relaciju S , tada je vrijednost pronađena;
 - b. Inače, ukloniti prvi element iz reda čekanja i dodati čvorove vezane za novodobijeni prvi element sa *AKO* relacijom na kraj reda čekanja (Na kraju prve iteracije, red će biti: *ŽENSKA OSOBA*, *LJUDSKA BIĆA*, na kraju druge *LJUDSKA BIĆA*, *SISARI*. U trećoj iteraciji će za čvor *LJUDSKA BIĆA* biti pronađena relacija $S=NOGE$).

Korak 3. Ako je vrijednost pronađena obavijestiti da je pronađena vrijednost relacije S čvora F, inače obavijestiti o neuspjehu (pronađena je vrijednost 2 relacije *NOGA* čvora *MAJA*).

Nedostatak navedenog algoritma nasljeđivanja se ogleda u nemogućnosti da se njime odredi vrijednost relacije za neki čvor koji nema direktan *ISA* link ili traženu relaciju, npr. relacija *NOGE* za čvorove Ljubinka i Rajka na slici 1. S toga se veoma često u semantičkim mrežama koriste **if-needed** procedure. If-needed procedurama se tražena vrijednost neke relacije izračunava po potrebi, a na osnovu vrijednosti drugih relacija. Proces nasljeđivanja, koji uključuje if-needed procedure se može predstaviti algoritmom:

Neka je F dati čvor, a S data relacija (link):

Korak 1. Sačiniti red čekanja koji se sastoji od čvora F i svih ostalih čvorova klase koji su sa čvorom F povezane *ISA* relacijom; čvor F treba da bude na čelu;

Korak 2. Dok red čekanja ne postane prazan ili je uspješna if-needed procedura pronađena, odnosno relacija pronađena, odrediti da li prvi element reda čekanja ima relaciju S ili odgovarajuću proceduru u if-needed relaciji:

- a. ako prvi element ima relaciju S, ili postoji procedura kojom se može odrediti vrijednost relacije S, tada je vrijednost pronađena,
- b. inače, ukloniti prvi element iz reda čekanja i dodati čvorove vezane za prvi element u ovom trenutku sa AKO na kraj reda čekanja;

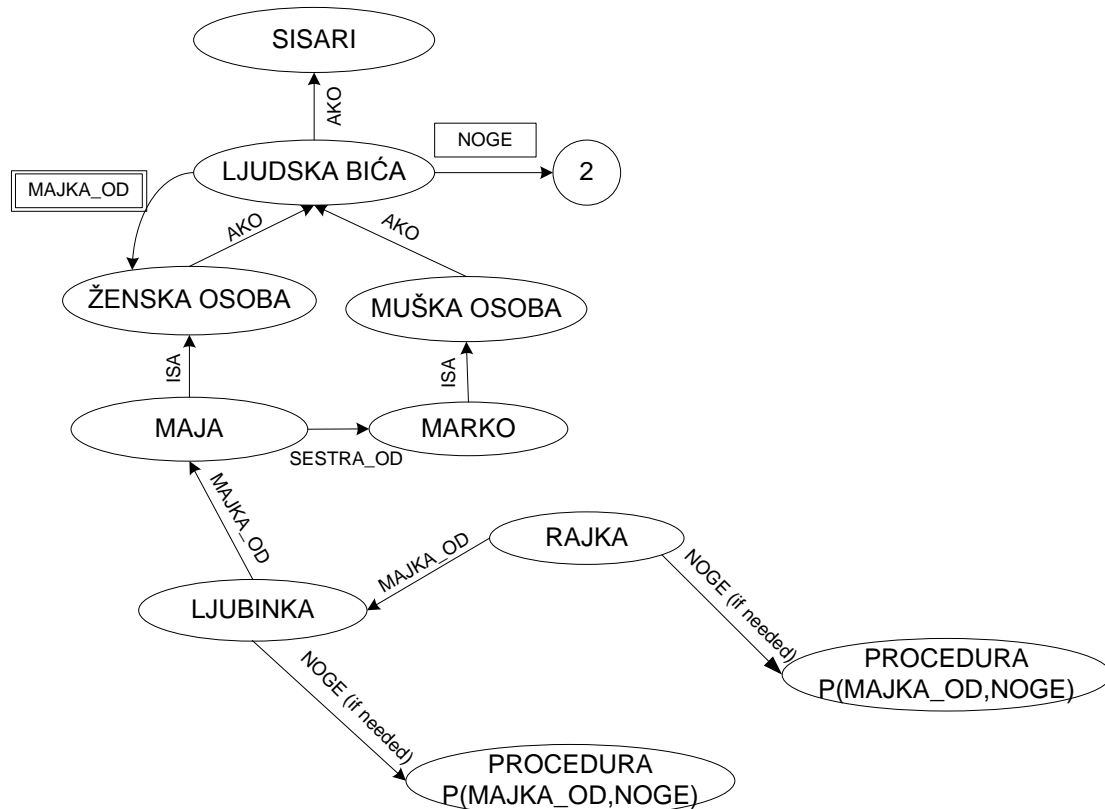
Korak 3. Ako je vrijednost pronađena obavijestiti da je pronađena vrijednost relacije S čvora F, inače obavijestiti o neuspjehu.

Primjer 1. Odrediti relaciju NOGE za čvorove LJUBINKA i RAJKA u semantičkoj mreži datoj na slici 2.

Ovdje smo definisali proceduru koja nam omogućava da odredimo vrijednost relacije NOGE za čvor koji nema direktnu relaciju ISA, ni NOGE, ali ima neku drugu relaciju koja ga povezuje sa nekim čvorom koji posjeduje željene relacije (ISA, NOGE). Argumente relacije određujemo tako da omogućimo navedeno povezivanje. Prvi argument je ime relacije koja će nas dovesti do najbližeg čvora sa željenom relacijom, a drugi ime relacije koju ćemo tražiti za taj čvor. Procedura će funkcionisati na sljedeći način:

- 1. Odrediti čvor koji je sa tekućim povezan povezan prvom navedenom relacijom u if-needed proceduri (MAJKA_OD).*
- 2. Odrediti vrijednost druge relacije (NOGE) za čvor nađen u koraku 1.*
- 3. Druga navedena relacija (NOGE) dobija vrijednost određenu u koraku 2.*

Ukoliko primijenimo algoritam nasljeđivanja koji uključuje i if_needed procedure, u prvoj iteraciji, u koraku 1 će se formirati red koji se sastojati od čvora LJUBINKA. U koraku 2 će biti ispunjen dio pod a. – aktivirana if_needed procedura, kojom se utvrđuje da relacija MAJKA_OD vodi ka čvoru MAJA. Nastavak se svodi na tražanje vrijednosti relacije NOGE za čvor MAJA (istim algoritmom), što smo ranije odradili, i dobili vrijednost 2. Ova vrijednost se dobija kao rezultat funkcije if_needed procedure i samim tim relaciji NOGE za čvor RAJKA.



Slika 2 Primjer semantičke mreže sa standardizovanim i izvedenim linkovima

Ukoliko bi tražili vrijednost relacije *NOGE* za čvor *RAJKA*, aktivirala bi se procedura *if_needed* koja bi ukazala na čvor *LJUBINKA* i tražila za njega relaciju *NOGE*. Proceduru za utvrđivanje vrijednosti relacije *NOGE* za čvor *LJUBINKA* smo već opisali. Pozivala bi se njegova *if_needed* procedura i rezultat proslijedio *if_needed* proceduri čvora *RAJKA*.

Može se reći da se rezonovanje/nasljeđivanje u semantičkim mrežama sastoji u praćenju veza u mreži. Zaključuje se na bazi veza i generišu se nove veze. Jednostavnost i efikasnost mehanizma nasljeđivanja, poredeći sa logičkom dokazivanjem teorema, je jedna od glavnih prednosti semantičkih mreža. Naime, nasljeđivanjem se eliminiše ponavljanje zajedničkih karakteristika. Osim toga, mnogo kompleksnih relacija se može predstaviti pomoću malog broja čvorova i linkova.

Nasljeđivanje postaje komplikovanije kada neki objekat pripada većem broju kategorija ili kada kategorija može biti podskup više od jedne kategorije; ovo se naziva višestrukim nasljeđivanjem. U ovim slučajevima algoritam nasljeđivanja može pronaći dvije ili više konfliktnih vrijednosti koje daju odgovor na postavljeni upit. Iz ovog razloga je višestruko nasljeđivanje zabranjeno u određenom broju objektno orijentisanih programskih jezika (kao što je Java) koji koriste nasljeđivanje u klasnoj hijerarhiji. Višestruko nasljeđivanje je obično dozvoljeno u semantičkim mrežama, ali neće biti pokriveno ovim kursom.

Jedan od najbitnijih aspekata semantičkih mreža jeste njihova sposobnost da predstave podrazumijevane vrijednosti za kategorije (objekte). Ukoliko bi se proučila slika 1 pažljivo, moglo bi se primijetiti da *Ljubinka* ima jednu nogu, uprkos činjenici da je ona ženska osoba i da sve ženske osobe, kao pripadnice ljudskih bića, imaju dvije noge. U striktno logičkoj BZ ovo bi predstavljalo kontradikciju, ali u semantičkim mrežama, zaključak (tvrdnja) da sve osobe imaju dvije noge ima

podrazumijevani status, to jeste, podrazumijeva se da osoba ima dvije noge ukoliko ovo nije u suprotnosti sa nekom konkretnijom (specifičnijom) informacijom. Podrazumijevana semantika je nametnuta prirodno samim algoritmom nasljeđivanja, jer ona prati link naviše od samog objekta i staje čim nađe željenu vrijednost čime podrazumijevana vrijednost se poništava konkretnijom vrijednošću.

Međutim, da bi se izbjeglo da se algoritmom nasljeđivanja za objekat RAJKA, aktiviranjem *if_needed* procedure date na slici 2, naslijedi specifična vrijednost za relaciju NOGE, objekta LJUBINKA, mora se promijeniti lista argumenata korišćene *if_needed* procedure. Ideja je da se premesti objekat koji ima specifičnu vrijednost relacije NOGE. To ćemo učiniti ako za objekat LJUBINKA, tražimo objekat na koji ukazuje relacija MAJKA_OD, pa za taj objekat vrijednost relacije MAJKA_OD i tek za dobijeni objekat vrijednost relacije NOGE. Sada bi nam *if_needed* procedura za čvor RAJKA bila *NOGE(if_needed) PROCEDURA P(MAJKA_OD, MAJKA_OD, NOGE)*.

Najveća prednost modela predstavljanja znanja semantičkim mrežama, je njihova fleksibilnost, koja se sastoji u tome da se novi čvorovi i veze mogu dodavati i brisati po potrebi. Osim navedene prednosti, veoma bitna prednost je mogućnost nasljeđivanja osobina. Svaki čvor semantičke mreže, koji predstavlja jedinku neke klase, posjeduje osobine opštijih klasa čiji je član. Dalje, semantičke mreže mogu biti izdijeljene u module, gdje je jedan ili više modula podskup ostalih. Ovakva modularna struktura pomaže hijerarhijskom zaključivanju i stoga je vremenski efikasna. Takođe, obzirom da su sve asocijacije između objekata eksplicitno povezane kao lukovi u mreži, i pretraživanje baze znanja je u principu vrlo efikasno. Ukoliko se mreža predstavi pomoću lista susjedstva (jedan od načina predstavljanja grafova u memoriji), a ne kao lista parova koji opisuju relacije u grafu, moguće je veoma brzo obaviti pretraživanje.

5.2. Okviri

Okviri su jedan od načina predstavljanja znanja kojim se hijerarhijski organizuje predstavljanje istinitih iskaza za neku opštu klasu elemenata. Okviri su u stvari struktura sačinjena od pregradaka (slotova) koji zajedno opisuju stereotipske objekte, događaje, akcije. **Okviri su posebno pogodni za simuliranje zdravorazumskog znanja**, što je veoma zahtjevan zadatak za VI. Osnovna karakteristika okvira jeste da oni predstavljaju povezane činjenice o nekom subjektu koji posjeduje mnogo podrazumijevanih karakteristika. Tako bi okvir bio dobar izbor za predstavljanje znanja o mehaničkim objektima, kao što je na primjer automobil. Komponente automobila, kao što je motor, šasija, kočnice, itd, bi okvirom bile povezane tako da se dobije opšta slika njihovih odnosa. Dalji detalji o pojedinim komponentama bi se mogli dobiti ispitivanjem strukture okvira [5]. Iako će detalji po jedinim marki automobila da variraju, većina automobila ima zajedničke karakteristike: točkove, motor, šasiju ...

Strukturno, svaki okvir znanja se sastoji od dva dijela i to: imena okvira, i liste parova atribut (pregradak – slot) – vrijednost (informacije o atributima datim objektima okvira). Na ovaj način je okvir povezan sa skupom činjenica, pravila, procedura, podrazumijevanih vrijednosti i uputstava koji ga u potpunosti određuju [1]. Svaki pregradak predstavlja standardno svojstvo ili atribut elementa koji je predstavljen okvirom. Obično je pregradak okvira popunjen vrijednostima, ali može

sadržati i druge okvire, listu drugih okvira, i/ili listu (opseg) osobina koje se vezuju za pregradak, tj. vrijednosti atributa koje se vezuju za atribut, pri čemu pregradak uvijek sadrži isti tip informacija. Pregradak obezbjeđuje način da se sistematski smjesti jedna komponenta stečenog iskustva u odnosu na predstavljenu klasu elemenata. Nekada se razvijaju mreže u kojima su čvorovi okviri.

Neki pregraci mogu poslužiti kao veza sa drugim okvirima, što znači da se okviri mogu vezivati u veće cjeline i na njima izvoditi logičke operacije. Kod okvira znanja se mogu, ne samo efikasno "upisivati/brisati" pregraci, nego je moguće i njihovo poređenje, sortiranje, pretraživanje, uvijek vezano za dva ili više okvira [1].

Opisi pomoću okvira u odnosu na mreže su jasniji, jer se semantička mreža najprije mora protumačiti. Okviri omogućavaju laku hijerarhijsku organizaciju našeg znanja.

Pridruživanjem procedura (demoni) postižu se neki sporedni efekti ili se postiže neka druga akcija u bazi znanja. Pregraci mogu sadržati procedure **if-needed**, **if-added** i **if-removal** [3].

- If needed procedura se izvršava kada je vrijednost pregratka koji sadrži ovu proceduru potrebna, a nijedna nije inicijalno prisutna u njemu, ili podrazumijevana vrijednost nije pogodna. U opštem slučaju if-needed metode se mogu formulirati tako da pregradak dobija vrijednost upitom korisnika, iz baze podataka, iz nekog algoritma, od drugog objekta ili od drugog ekspertnog sistema.
- If added procedura se aktivira pri dodjeli vrijednosti pregratka.
- If removal procedura se aktivira pri uklanjanju ili zastarijevanju vrijednosti pregratka.

Procedure se mogu napisati na nivou klase i naslijeđuju se prilikom izvođenja klasa.

Smatra se da strukture poput okvira organizuju naše vlastito znanje svijeta. Mi se prilagođavamo svakoj novoj situaciji pozivanjem informacija strukturiranih na osnovu starih iskustava. Tada mi modernizujemo ili obnavljamo detalje prošlih iskustava, da bi se prikazale individualne razlike nove situacije. Kao primjer neka posluži hotelska soba. Svaka hotelska soba je otprilike ista: očekujemo krevet, kupatilo, mjesto gdje ćemo smjestiti prtljag, telefon, brošuru s uputstvima. Detalji se mogu razlikovati. Mi ne trebamo graditi novo znanje hotelske sobe za svaki novi hotel u koji dolazimo. Svi dijelovi generičkog hotela su organizovani u konceptualne strukture kojima pristupamo kada ulazimo u hotelske sobe; pojedinosti nekih soba se dostavljaju ako je potrebno.

Okviri daju nove mogućnosti semantičkim mrežama omogućavajući da se složeni objekti prikazuju kao jednostruki okviri, a ne kao velike mrežne strukture. To je prirodan način za prikaz stereotipnih entiteta, klasa, nasljeđivanja i implicitnih vrijednosti. Okviri omogućavaju prikaz većih struktura znanja, hijerarhijskih struktura, stupnjeve apstrakcije, te kombinaciju deklarativnih i proceduralnih znanja. Okviri se koriste i u istraživanjima i u izgradnji praktičnih aplikacija.

Možemo reći da su okviri znanja mreže čvorova i relacija organizovanih hijerarhijski, gdje čvorovi na vrhu predstavljaju uopštene koncepte (objekte), a niži čvorovi specifične dijelove tih konceptata. Predstavljaju pogodno sredstvo za prikazivanje podataka i relacija.

Dakle, generički okvir (klasa) sadrži generičke informacije o konceptu, a to su: deskriptivno ime koncepta, skup osobina (pregradaka) koje su karakteristične za sve pridružene objekte, kao i vrijednost osobina za koje se smatra da su zajedničke objektima (podrazumijevane vrijednosti). Koncept automobila bi se mogao predstaviti kao:

```

Okvir (ime klase): Automobil
  pregradak      vrijednost
  proizvođač      ?
  model           ?
  godina         ?
  mjenjač        MANUELNI
                 if-needed: Procedure FIND_MJENJA
  motor          (BENZIN, NAFTA)
                 if-added: Procedure DODAJ_GORIVO
  točkova        4
                 if-needed: Procedure FIND_TOCKOV
  boja           ?

```

Slika 3 Primjer generičkog okvira – klase

Dakle, podrazumijevano je da svaki putnički automobil ima četiri točka, a statistika kaže da se u Crnoj Gori najviše voze automobili sa manuelnom mjenjačem, te se vrijednosti za odgovarajuće slotove uzimaju kao podrazumijevane. Dodata je procedura if-needed koja bi se aktivirala u slučaju da se pojavi automobil koji nema manuelni mjenjač. Dalje, vidimo da vrijednost predgratka može biti data kao opseg vrijednosti. Tako se smatra da će se vrijednost za vrstu motora naći u skupu (BENZIN, NAFTA). U slučaju da je potrebno dodati neku novu vrijednost (alternativnu vrstu goriva, npr. plin), if-added procedura bi bila aktivirana. S druge strane, ne može se predvidjeti proizvođač, model, godina proizvodnje i boja, te se ti pregraci ostavljaju prazni.

Pored generičkih klasa postoje i okviri koji predstavljaju konkretnu instancu, objekat klase. Koriste se za predstavljanje specifičnog znanja. Instanca ili primerak je specifičan objekat iz klase objekata koji sadrži sve karakteristike okvira klase, ali obično sadrži i informacije koje su specifične za taj objekat. Primjer instance klase automobil bi bio:

```

Okvir: AUTOMOBIL      (ime objekta MojAutomobil)
  pregradak      vrijednost
  proizvođač      RENAULT
  model           MEGANE COUPE
  godina         2010
  mjenjač        MANUELNI
  motor          BENZIN
  točkova        4
  boja           SIVA

```

Slika 4 Primjer instance – objekta generičkog okvira

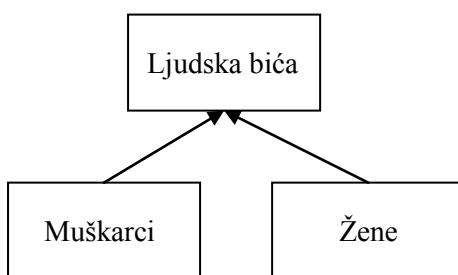
Vidimo da je ovdje potrebno navesti i ime objekta.

Jedna od mogućnosti okvira je nasljeđivanje klasa (generičkih okvira), nasljeđivanjem pregradaka i implicitnih vrijednosti preko klasa/podklasa i klasa/objekata. Pod nasljeđivanjem pregradaka podrazumeva se činjenica da izvedeni

okvir mora imati sve pregratke kao i okvir koji je generalniji od njega, bez obzira da li su oni popunjeni vrijednostima ili nijesu. Pod nasljeđivanjem implicitnih vrijednosti pregradaka se podrazumijeva da se vrijednost nekog pregratka u posmatranom okviru, ako nije data, može utvrditi na osnovu postojeće vrijednosti istoimenog pregratka nekog od generalnijih okvira. Kada se stvara instanca – objekat (MojAutomobil) generičkog okvira (Automobil), sistem nastoji popuniti slotove, bilo da se pita korisnik, ili preko implicitnih vrijednosti ili preko procedura. Na taj način se nasljeđuju svojstva preko hijerarhije. Izvedenom primjerku ili klasi se uobičajeno dodjeljuju i svojstva ili vrijednosti koje samo on posjeduje.

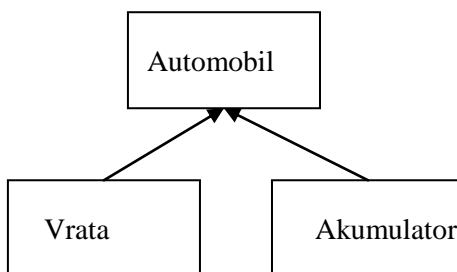
Postoje tri vrste veza između klasa u slučaju da se koriste podklase [3]:

- generalizacija (AKO): klasa/podklasa. Na primjer, muškarci i žene kao podklase imaju vezu generalizacija prema višoj klasi ljudska bića. Naravno, podklase muškarci i žene imaju i neka specifična svojstva u odnosu na klasu iz koje su izvedena:



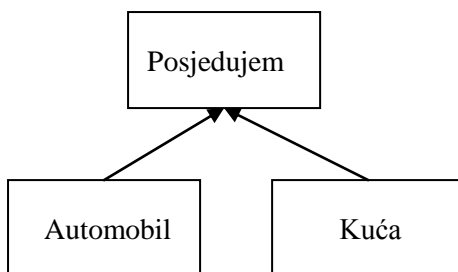
Slika 5 Generalizacija kao veza klasa/podklasa

- agregacija (A Part OF (APO)): ovo su kontejnerski okviri



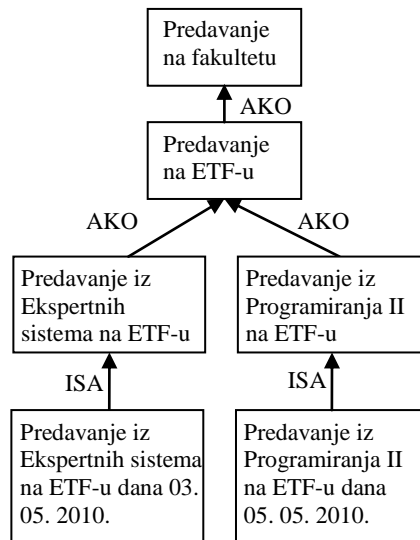
Slika 6 Agregacija kao veza klasa/podklasa

- asocijacija:



Slika 7 Asocijacija kao veza klasa/podklasa

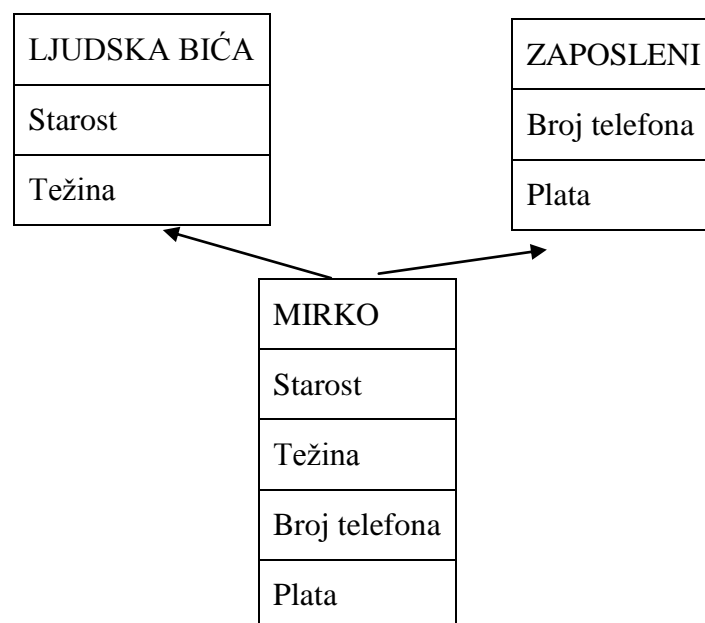
Pregraci takođe mogu sadržati i relacije tipa AKO koja predstavlja generičku relaciju – jedna klasa je specijalizacija druge klase, i ISA koja je specifična relacija – objekat je specijalna instanca neke klase:



Slika 8 Ilustracija relacija AKO i ISA kod okvira

Podklase nasleđuju ponašanje svojih roditelja ili nadklase. Ako je bilo koja procedura priključena aspektu nadklase (roditeljske klase), ona će biti naslijeđena od podklase (potomka). Dakle, prilikom nasleđivanja, okvir potomak će naslijediti sve vrijednosti roditelja, ako se ove vrijednosti namjerno ne izmijene. Na primer, MojAutomobil može da ima samo tri točka. Ovdje se mora voditi računa o tome da se svaki okvir koji je izuzetak od standardnog mora eksplicitno obraditi kada se projektuje sistem zasnovan na okvirima.

I kod okvira postoji višestruko nasleđivanje. Naime, veoma često se javlja potreba da se nova klasa dobije kombinacijom više postojećih. Na primjer, okvir Mirko se može predstaviti kao dio svijeta ljudskih bića i kao dio svijeta zaposlenih u nekoj kompaniji. Objekat Mirko nasleđuje informacije od dva prethodna okvira: ljudska bića i zaposlenih, slika dolje.



Slika 9 Ilustracija višestrukog nasleđivanja kod okvira

LITERATURA

- [1] Poliščuk, E.J.: *Ekspertni sistemi*, Informatička literatura JEP (vlastito izdanje), Podgorica, 2004. Joseph C. Giarratano, Gary D. Riley.: *Expert Systems: Principles and Programming*, Prentice Hall, 2nd ed., 2002.
- [2] Russell S., Norvig P.: *Artificial Intelligence: A Modern Approach*, Prentice Hall, NJ, 1995.
- [3] <http://ri4es.etf.rs/predavanja.html>, posljednji put pristupano, 08. 02. 2010. godine.
- [4] Konar A.: *Artificial Intelligence and Soft Computing: Behavioral and Cognitive Modeling of the Human Brain*, CRC Press, December 8, 1999.
- [5] Joseph C. Giarratano, Gary D. Riley.: *Expert Systems: Principles and Programming*, Prentice Hall, 2nd ed., 2002.