

Case-Based Reasoning

ZAKLJUČIVANJE NA OSNOVU SLUČAJEVA

V. Kurbalija, M. Ivanović, Z. Budimac
Institut za matematiku i informatiku
Prirodno-matematički fakultet
Univerzitet u Novom Sadu, Jugoslavija

Trg Dositeja Obradovića 4, 21000 Novi Sad, Jugoslavija
e-mail: {kurba, mira, zjb}@im.ns.ac.yu

Apstrakt: Zaključivanje na osnovu slučajeva je relativno novi pristup u inteligentnom pretraživanju (velikih) baza podataka. Svaka nova pretraga je bazirana na prethodnim sličnim slučajevima, tako da je iskustvo prisutno u svakoj novoj pretrazi. Ovom pristupu je posvećena velika pažnja kako u elektronskom poslovanju tako i u pretraživanju preko Interneta. U ovom radu prezentovani su osnovni koncepti zaključivanja na osnovu slučajeva i neke karakteristične primene.

1. Uvod

Uopšteno, zaključivanje na osnovu slučajeva (CBR – Case-Based Reasoning) je tehnika za rešavanje problema kod koje se pamte prethodni slučajevi (u obliku: *problem* – *rešenje*) i koriste za rešavanje novih problema, gde se staro rešenje može, u nekoj meri, adaptirati da bi odgovaralo novom problemu.

1.1. Znanje

Generalno govoreći, *znanje* predstavlja pojam koji opisuje nešto što čovek, formalni sistem ili mašina može da iskoristi za izvršenje određenog zadatka (za rešenje problema). Da bi neko upotrebio znanje mora da ima pristup znanju i da ume da ga iskoristi za rešenje problema.

Reprezentacija znanja se sastoji od određene strukture podataka i dodatnih operatora koji omogućavaju izmenu podataka u datoj strukturi. Najzastupljenija struktura podataka u CBR-u je *atribut-vrednost reprezentacija*.

Svaki atribut je određen sa:

- imenom A
- skupom $dom(A)$ – domen atributa A
- promenljivom x_A – koja može da uzima razne vrednosti iz domena.

Za konačan niz atributa A_i , $1 \leq i \leq n$, *atribut-vrednost vektor* je n -torka (a_1, \dots, a_n) tako da $a_i \in dom(A_i)$.

Međutim, ako nekada nisu poznate vrednosti svih atributa (što je čest slučaj u CBR-u), tada se mora dozvoliti da neke promenljive postoje i bez vrednosti (u tom slučaju je to nepoznata vrednost).

Ako bi, na primer, posmatrali atribut $A=Boja$, tada bi domen ovog atributa mogao biti $dom(Boja)=\{crvena, žuta, plava, zelena, ljubičasta, \dots\}$, dok bi promenljiva x_{Boja} dobijala neku vrednost iz skupa $dom(Boja)$.

Takođe, ako bi posmatrali niz atributa $Boja, Materijal, Veličina$, koji bi mogli da opisuju neki odevni predmet, tada bi neki atribut-vrednost vektori mogli biti: $(plava, pamuk, XL)$, $(žuta, pamuk, M)$, $(zelena, vuna, XL)$, $(žuta, sintetika,)$, $(crvena, , XXL)$.

U poslednja dva vektora nisu poznate vrednosti atributa $Veličina$ i $Materijal$ respektivno, tako da odgovarajuće promenljive nemaju vrednost.

1.2. Osnovni koncepti

CBR je tehnika veštačke inteligencije koja se ubraja u tehnike pogodne za rešavanje problema. Osnovni scenario i način funkcionisanja CBR-tehnike, pojednostavljeno, izgleda ovako:

Da bi pronašli rešenje aktuelnog problema, tražimo sličan problem u iskustvenoj bazi slučajeva, uzimamo rešenje problema iz prošlosti koji je najbliži aktuelnom problemu i koristimo njega kao početnu tačku u traženju konačnog rešenja za aktuelni problem.

Generalna težnja u svim sistemima baziranim na znanju je korišćenje nekih prošlih iskustava. Iskustvo može biti nešto što je tačno ili netačno, korektno ili nekorektno, manje ili više korisno. Možemo ga predstaviti pravilom, ograničenjem, nekim generalnim zakonom ili jednostavno samo čuvanjem događaja koji se desio.

Slučaj

Definicija 1. *Slučaj* je neka zapamćena situacija gde je problem delimično ili potpuno rešen.

Najjednostavnije, slučaj možemo predstaviti kao uređeni par: $(problem, rešenje)$ gde *problem* predstavlja opis problema a *rešenje* je opis manje ili više preciznog rešenja postavljenog problema.

Postojanje slučaja (u nekoj bazi slučajeva) znači da se odgovarajuća situacija desila u prošlosti. Ova situacija sadrži odluke za koje je neko smatrao da su korisne. Međutim, možda za nekog drugog te odluke nisu toliko korisne, a u nekim situacijama mogu biti čak i pogrešne.

Iz ovoga jasno proizilazi da slučajevi moraju biti pažljivo izabrani, tako da se izdvajaju različite kategorije slučajeva: dobar, tipičan, važan, dvosmislen, nepotreban itd.

U mnogim praktičnim aplikacijama se susreću problemi koji su nepotpuno opisani. Ponekad i problem i rešenje mogu biti nepotpuni i tada govorimo o *nepotpunim slučajevima*. *Kompletiranje slučaja* je još jedan zadatak koji može biti rešen CBR-om.

Sličnost

Sledeći veoma bitan koncept u CBR-u je *sličnost*. Dok se u klasičnim bazama podataka informacija može dobiti samo tačnim poređenjem slogova, u CBR-u se informacija može dobiti iako ne postoje slogovi koji apsolutno odgovaraju upitu (traže se slogovi koji su slični upitu). Pojam sličnosti je dualan matematičkom pojmu – *rastojanje*.

Definicija 2. *Sličnost* se može definisati kao funkcija: $sim : U \times CB \rightarrow [0, 1]$, gde je U skup svih objekata u nekom sistemu, a CB je baza slučajeva (samo oni slučajevi koji su rešeni i snimljeni u prošlosti). Logično, veća vrednost funkcije sličnosti ukazuje da su dva objekta više slična.

Granični slučaj je $sim(x, x) = 1$, što znači da je objekat najbliži samom sebi. Uglavnom se umesto sličnosti koristi opštiji pojam *prihvatljivost*.

U meri sličnosti se najčešće akumulira znanje u CBR sistemima. Ustanovljene su različite vrste sličnosti:

- *Površinska sličnost* razlikuje samo sintaksne osobine reprezentacije
- *Lokalna sličnost* razmatra samo sličnost pojedinih atributa i
- *Globalna sličnost* predstavlja potpun pogled na slučaj. Obično se izvodi iz lokalne sličnosti.

Adaptacija

Najjednostavniji način za rešavanje nekog problema je da se u uzme nepromenjeno rešenje slučaja koji je najbliži početnom problemu i iskoristiti se kao rešenje početnog problema. Međutim, u mnogim domenima i male razlike između problema zahtevaju značajne modifikacije na njihovim rešenjima.

Definicija 3. Pravljenje odgovarajućih modifikacija na rešenjima se naziva *adaptacija slučaja*.

Generalna zamisao u svim CBR sistemima je da se rešenje sličnog problema može lako adaptirati tako da predstavlja rešenje početnog problema.

Ideja čuvanja znanja u CBR-u je različita od tradicionalnog koncepta čuvanja znanja u programiranju. Osnovna razlika je u tome što se celokupno znanje koristi pri rešavanju čak i najmanjeg zadatka.

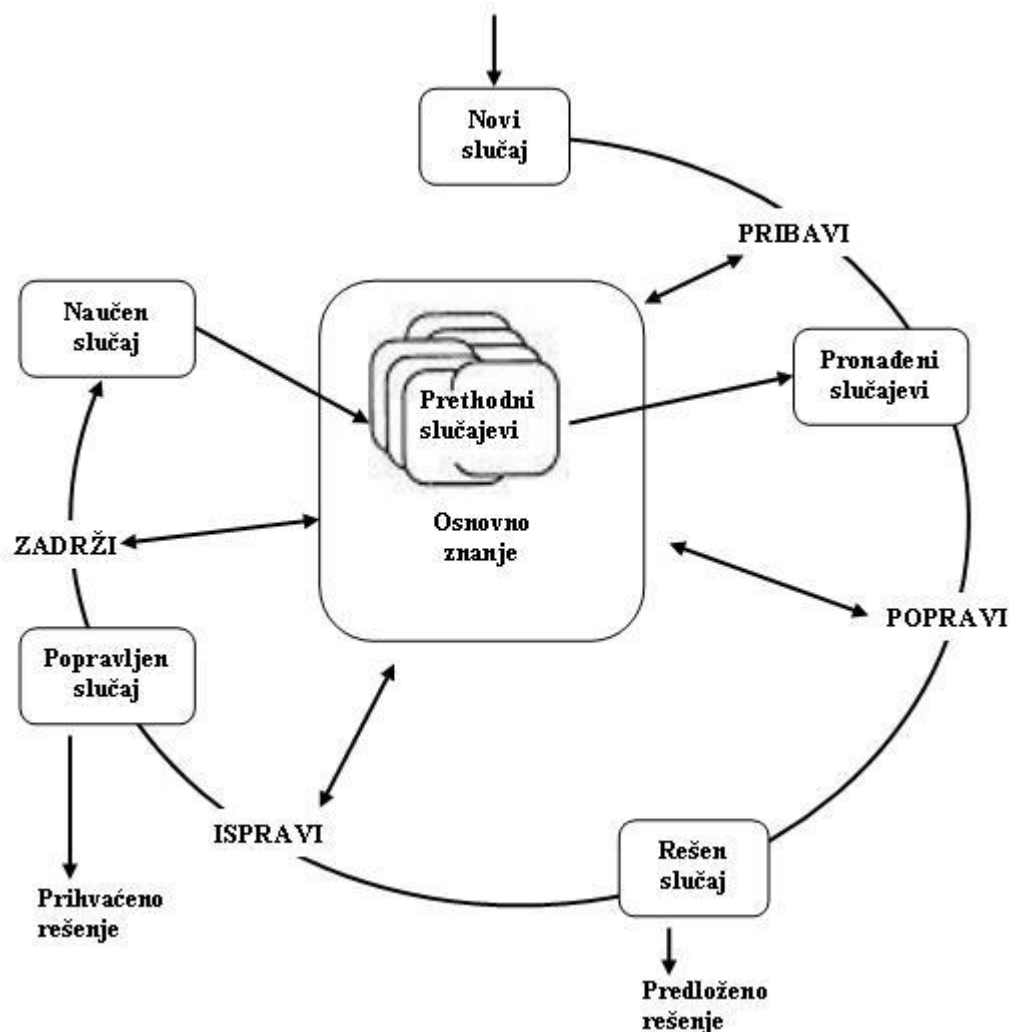
CBR sistem - način funkcionisanja

U CBR-u znanje se čuva korišćenjem sledećih koncepata:

1. Rečnik koji se koristi za predstavljanje slučajeva,
2. Mera sličnosti,
3. Baza slučajeva i
4. Transformacija rešenja.

Koncepti 1, 2 i 4 se formiraju u toku kompajliranja sistema (compile-time), dok koncept 3 dolazi do izražaja u vremenu izvršavanja (run-time). Ovo je velika prednost CBR-a jer je znanje moguće menjati i u toku izvršavanja programa.

U principu svaki koncept može da nosi svo znanje (a može da obuhvati i neke nebitne elemente). Moguće je menjati jedan koncept, a da to ne utiče na ostale. Zbog ove osobine održavanje CBR sistema je mnogo lakše nego održavanje klasičnih sistema znanja [2], [5].



Slika 1. Glavne faze u CBR sistemu po Aamodt and Plaza (1994)

CBR sistem pored pronalaženja rešenja mora da obavlja još niz aktivnosti. Glavne faze koje CBR sistem mora da obavi pri svakom rešavanju problema prikazane su na slici 1 [1].

U fazi *pribavi* se pronalaze najslićniji slučajevi početnom problemu. Ovi slučajevi se sastoje od opisa prethodnih problema koji su najslićniji početnom problemu i njihovih uspešnih rešenja. Pronađeni slučajevi a posebno njihova rešenja predstavljaju polaznu osnovu za rešavanje aktuelnog problema.

U fazi *popravi* vrše se neke modifikacije na pronađenim slučajevima da bi se došlo do što boljeg rešenja početnog problema (adaptacija slučaja). Ova faza ima različite rioritete u različitim aplikacijama. Dok u nekim aplikacijama nema potrebe za adaptacijom, u nekim aplikacijama je ova faza nezamenjiva.

Kako CBR sistem samo predlaže rešenje, postoji potreba za dokazom tačnosti ili za eksternom validacijom. To je zadatak faze *ispravi*. U ovoj fazi sistem očekuje od korisnika ili od eksperta potvrdu o tačnosti predloženog rešenja. Ovo je obavezna aktivnost koja omogućava sistemu da nauči ispravno rešenje.

U fazi *zadrži*, znanje naučeno iz ovog novog problema se integriše u sistem. Učenje se obično realizuje ubacivanjem novog slučaja i/ili modifikacijom mere sličnosti između sačuvanih i korišćenih slučajeva i aktuelnog slučaja.

2. Proširenje nekih koncepata CBR

U prethodnom poglavlju opisani su osnovni koncepti koji su neophodni za proučavanje i primenu CBR-tehnike. Međutim, za praktičnu primenu i implementaciju CBR-tehnike neophodno je proširiti ove koncepte i detaljnije ih opisati. U ovom poglavlju detaljnije su opisani: *kompletiranje zadatka* kao opšti postupak pronalaženja rešenja u svim CBR-sistemima, *entiteti informacija* kao pogodan način za predstavljanje slučajeva, *funkcija prihvatljivosti* kao matematički aparat koji se koristi za pronalaženje rešenja i *mreža za pronalaženje slučajeva* kao jedan od načina za pronalaženje rešenja.

2.1. Kompletiranje zadatka

CBR kao metoda za rešavanje problema ima za cilj da za početni problem pronađe adekvatno rešenje. Obzirom da je bazirana na analizi prethodnih slučajeva ova metoda nas vodi do zaključka da se slučaj deli na deo vezan za problem i deo vezan za rešenje. Za dati novi problem, potrebno je naći sličan problem u bazi slučajeva i adaptirati njegovo rešenje tako da se dobije rešenje početnog problema. Međutim, rešavanje problema obično ne počinje sa kompletnim opisom problema, pa je samim tim i identifikacija krajnjeg rešenja znatno otežana.

Novi slučaj se, na početku, obično sastoji od male količine informacija koje daju samo prvi utisak o početnom problemu. Ceo CBR proces se sastoji od kompletiranja zadatka do konačnog rešenja. Posledica ovoga je da rezultujući slučaj obično zavisi od niza odluka. Ove odluke su inicijalno otvorene i zavise od odluke korisnika. Kako kompletiranje zadatka napreduje, sve više informacija se sakuplja u ciju dostizanja uspešnog rešenja. Posle kompletiranja novo iskustvo je naučeno i može se iskoristiti za kasnije zadatke. Ovaj proces rešavanja zadatka prikupljanjem informacija naziva se *kompletiranje zadatka*.

2.2. Entiteti informacija

Slučaj predstavlja rezultat procesa kompletiranja zadatka. Svaki korak tog procesa dodaje neke nove entitete informacija u slučaj. Trenutna situacija u svakom koraku je opisana entitetima informacija poznatim u tom trenutku. Konačan slučaj koji će se pojaviti u bazi je kompletan skup entiteta informacija.

Skupljene informacije u toku procesa kompletiranja zadatka su iz realnog sveta (rezultat testa, odluka korisnika itd). CBR sistem, na osnovu raspoloživih informacija, samo predlaže sledeći korak. Osim toga broj entiteta informacija u slučaju može da varira. Na korisniku je da odluči kada je slučaj rešen (Npr. Popravku nekog uređaja je moguće uraditi i bez izračunavanja cele dijagnoze). Entiteti informacija, koji će se kasnije javiti u bazi slučajeva, mogu biti samo podskup entiteta informacija skupljenih tokom kompletiranja slučaja.

Definicija 4. *Entitet informacija* je atomični (jedinični) deo slučaja ili upita. Skup svih potencijalnih entiteta informacija u nekom domenu označava se sa E .

- Slučaj c je skup entiteta informacija, $c \subseteq E$.
- Upit q je takođe skup entiteta informacija, $q \subseteq E$.
- Skup slučajeva (u bazi) se označava sa C , $C \subseteq P(E)$.

U mnogim aplikacijama entiteti informacija su jednostavno realizovani kao atribut-vrednost parovi.

Na primer: $\langle \text{cena}, 1000 \rangle$, $\langle \text{cena}, 324 \rangle$, $\langle \text{boja}, \text{plavo} \rangle$, $\langle \text{masa}, 54 \text{ kg} \rangle$.

U prethodnom primeru su prva dva entiteta informacija uporediva, jer imaju isto ime atributa, dok su ostali entiteti informacija neuporedivi. U ovom slučaju skup E se deli u disjunktne podsupove E_A , gde E_A sadrži vrednosti iz E za atribut A . Dakle, za ovaj konkretan primer ovi skupovi bi bili:

$E_{\text{cena}} = R$, gde je R skup realnih brojeva;
 $E_{\text{boja}} = \{\text{crvena}, \text{žuta}, \text{plava}, \text{zelena}, \text{ljubičasta}, \dots\}$;
 $E_{\text{masa}} = R^+$, gde je R^+ skup pozitivnih realnih brojeva.

I upiti i slučajevi mogu se posmatrati i kao atribut-vrednost vektori (a_1, \dots, a_n) , gde a_i predstavlja vrednost i -tog atributa A_i .

Formalno gledano nema razlike između upita $q = (q_1, \dots, q_n)$ i slučaja $c = (c_1, \dots, c_n)$. Ako se slučajevi i upiti posmatraju kao atribut-vrednost vektori za konačan skup atributa A_1, \dots, A_n , tada svaki slučaj ili upit mogu da sadrže najviše jedan entitet informacija iz svakog E_{A_i} .

2.3. Funkcija prihvatljivosti

U CBR sistemima potrebno je iskoristiti sličnost između pojedinih entiteta informacija u cilju pronalaženja slučajeva pod pretpostavkom da ti slučajevi mogu poslužiti za rešavanje početnog problema. Korisnost izabranog slučaja zavisi od niza okolnosti iz realnog sveta koje nisu poznate sistemu u procesu pronalaženja tog slučaja. To znači da je korisnost slučaja aposteriorni kriterijum koji se ne može izračunati pri pronalaženju nego tek kada su slučajevi pronađeni. Stoga se za izračunavanje korisnosti slučaja uzima apriorna aproksimacija koja se naziva – *prihvatljivost*.

Definicija 5. Funkcija kojom se izračunava prihvatljivost slučaja naziva se *funkcija prihvatljivosti*.

Da bi funkcija prihvatljivosti bila dobra aproksimacija za korisnost potrebno je da budu zadovoljene sledeće osobine[4]:

O1: Slučaj može biti prihvatljiv za dati upit i ako postoje entiteti informacije koji nisu odgovarajući.

O2: Slučaj može biti neprihvatljiv za upit ako postoji neprihvatljiv entitet informacije (npr. fiksni budžet može da spreči skupe ponude).

O3: Isti entitet informacije može imati različitu važnost za različite slučajeve (npr. entitet informacija <pol, muški> ima različitu važnost u testiranju trudnoće i u testiranju na virus gripa).

O4: Isti entitet informacije može imati različitu važnost za različite upite ako se uzmu u obzir namere korisnika (npr. materijal koji se koristi za izradu nekog dela ima različiti prioritet u različitim dizajnerskim upitima, jer je pod nekim okolnostima materijal jako bitan, dok je pod nekim drugim totalno nebitan).

O5: Entiteti informacije ne moraju biti nezavisni jedni od drugih (npr. kratak spoj i nestanak struje su različiti entiteti informacija koji su u međusobnoj vezi).

Već je rečeno da su upiti definisani kao skupovi entiteta informacija. Međutim, postoji jedna posebna vrsta upita - težinski upit koji predstavlja generalizacija ovog koncepta.

Definicija 6. *Težinski upit* dodeljuje važnost svakom entitetu informacije sledećom funkcijom:

$$\alpha_q : E \rightarrow R$$

gde R predstavlja skup realnih brojeva, a $\alpha_q(e)$ ukazuje na važnost entiteta informacije e u upitu q .

Visoke vrednosti težinskog upita ukazuju na veću važnost entiteta informacije za upit dok negativne vrednosti ukazuju na odbacivanje slučaja koji sadrže taj entitet informacije. Vrednost 0 se koristi kao neutralni element tj. $\alpha_q(e) = 0$ znači da pojavljivanje entiteta informacije e nije bitno u upitu q .

Definicija 7. *Lokalna funkcija prihvatljivosti* σ za atribut A je definisana na skupu $dom(A)$ na sledeći način:

$$\sigma : dom(A) \times dom(A) \rightarrow R,$$

gde $dom(A) \times dom(A)$ predstavlja skup uređenih parova entiteta informacija gde je prvi entitet informacija iz upita, a drugi iz slučaja. Veća vrednost $\sigma(e, e')$ ukazuje na veću prihvatljivost entiteta informacije e (iz upita q) za entitet informacije e' (iz slučaja c).

Koristeći funkciju σ moguće je izračunati prihvatljivost jednog entiteta informacije e' iz slučaja za jedan entitet informacije e iz upita. Međutim, upit može da sadrži više entiteta informacije e tako da je $\sigma(e, e')$ definisano za različite entitete informacija iz upita i jedan

entitet informacije e' iz slučaja. Međutim postavlja se pitanje: kako ove vrednosti treba akumulirati u jednu vrednost za e' koja izražava rezultujuću prihvatljivost za dati upit.

Definicija 8. Neka $E_e = \{e_1, \dots, e_n\}$ označava skup svih entiteta informacije koji su uporedivi sa entitetom informacije e iz slučaja c tj. $E_e := \{e' \mid \sigma(e', e) \text{ definisano}\}$. Lokalna funkcija akumulacije π_e za e je:

$$\pi_e : \underbrace{R \times \dots \times R}_{n \text{ puta}} \rightarrow R,$$

tako da $\pi_e(a_1, \dots, a_n)$ označava akumuliranu prihvatljivost u e . Vrednosti a_i označavaju doprinose entiteta informacije $e_i \in E_e$ u zavisnosti od njihove pojave u upitu q i od njihove lokalne prihvatljivosti $\sigma(e_i, e)$.

Doprinosi se računaju funkcijom:

$$f : R \times R \rightarrow R$$

tako da je $a_i = f(\alpha_q(e_i), \sigma(e_i, e))$.

Najjednostavniji slučaj za funkciju f je proizvod, tako da se doprinosi računaju na sledeći način $a_i = \alpha_q(e_i) \cdot \sigma(e_i, e)$. Ova veza ima efekat da je $a_i = 0$ ako je $\alpha_q(e_i) = 0$ ili ako je $\sigma(e_i, e) = 0$. To znači da odsustvo entiteta informacije e_i ima isti efekat na e kao da je vrednost funkcije prihvatljivosti $\sigma(e_i, e) = 0$.

Pronalaženje slučaja iz baze slučajeva posmatramo kao proces prisećanja. Prisećanje može biti različite snage, tako da se slučajevi "takmiče" za određeni upit. Moguće je definisati za svaki entitet informacije $e \in E$ vrednost prisećanja za svaki slučaj. Slučajevi tada mogu da se porede u odnosu na akumulaciju vrednosti prisećanja iz entiteta informacije datog upita. Međutim, dodatni problem je kako računati vrednost prisećanja. Pojedinačno dodeljivanje za svaki par (e, c) gde $e \in E$, $c \in C$ nije adekvatno. Stoga se snaga prisećanja za entitet informacije $e_i \in c$ zamenjuje novim pojmom - važnost, a ova veličina se računa funkcijom važnosti.

Definicija 9. Važnost između entiteta informacije i slučaja je data funkcijom važnosti:

$$\rho : E \times C \rightarrow R.$$

Vrednost $\rho(e, c)$ se posmatra kao mera važnosti entiteta informacije e za pronalaženje slučaja c . Iz ovoga sledi da je $\rho(e, c)$ definisano ako i samo ako je $e \in c$.

Negativne vrednosti $\rho(e, c)$ se mogu koristiti sa značenjem "nemoj pronaći slučaj c ako je relevantan entitet informacije e ".

Prihvatljivost slučaja c za upit q se akumulira iz doprinosa entiteta informacije $e \in c$ i njegove važnosti $\rho(e, c)$. Doprinosi p_e entiteta informacije e se računaju korišćenjem lokalne funkcije akumulacije π_e kao što je rečeno u definiciji 8. Sakupljena akumulacija u slučajevima se računa globalnom funkcijom akumulacije.

Definicija 10. Globalna funkcija akumulacije π_c se definiše na sledeći način:

$$\pi_c : \underbrace{R \times \dots \times R}_{k \text{ puta}} \rightarrow R,$$

za slučaj $c = \{ e_1, \dots, e_k \}$. Akumulirana prihvatljivost slučaja c se računa sa $\pi_c(p_1, \dots, p_k)$ gde p_i predstavlja doprinos entiteta informacije $e_i \in c$. Ovaj doprinos zavisi od važnosti $\rho(e_i, c)$ i od realne vrednosti x_i . x_i predstavlja akumuliranu lokalnu prihvatljivost $\pi_{e_i}(a_1, \dots, a_n)$.

Doprinosi p_i se računaju funkcijom g :

$$g : R \times R \rightarrow R$$

tako da $p_i = g(x_i, \rho(e_i, c))$.

Globalna funkcija prihvatljivosti za slučajeve u odnosu na određeni upit, koja zadovoljava osobine O1, ... ,O5, se računa pomoću proširene funkcije prihvatljivosti koja je data sledećom definicijom.

Definicija 11. Prihvatljivost između težinskih upita i slučajeva je izražena proširenom funkcijom prihvatljivosti:

$$acc : R^E \times P(E) \rightarrow R$$

Prihvatljivost $acc(\alpha_q, c)$ slučaja c za težinski upit α_q izražena preko prethodnih funkcija sada izgleda ovako:

$$acc(\alpha_q, c) = \pi_c(g(\pi_{e_1}(f(\alpha_q(e_{11}), \sigma(e_{11}, e'_1))), \dots, f(\alpha_q(e_{1n1}), \sigma(e_{1n1}, e'_1))), \rho(e_1', c)),$$

...

$$g(\pi_{e_k}(f(\alpha_q(e_{k1}), \sigma(e_{k1}, e'_k))), \dots, f(\alpha_q(e_{knk}), \sigma(e_{knk}, e'_k))), \rho(e_k', c))$$

gde je: $c = \{e'_1, \dots, e'_k\}$ i $E_{e_i} = \{e_{i1}, \dots, e_{ini}\}$ za $i=1, \dots, k$.

Jedna od mogućnosti je da se funkcije f i g posmatraju kao proizvod a π_e i π_c kao sume, tada se dobija sledeći oblik acc funkcije:

$$acc(\alpha_q, c) = \sum_{e' \in c} \rho(e', c) \sum_{e \in E_{e'}} \sigma(e, e') \cdot \alpha_q(e)$$

Za navedenu formu funkcije acc ispunjene su osobine O1, ... ,O4. Osobine O1 i O2 su ispunjene zbog korišćenja težinskih suma, a osobine O3 i O4 su ispunjene na sledeći način:

- O3: Različita važnost entiteta informacija za različite slučajeve je izražena različitim vrednostima funkcije važnosti $\rho(e', c)$
- O4: Različita važnost entiteta informacija u upitu je izražena sa $\alpha_q(e)$.

Zadovoljenje osobine O5 (zavisnost entiteta informacija) ipak zahteva pažljiviji izbor funkcija f , g , π_e i π_c .

2.4. Mreža za pronalaženje slučajeva

Mreža za pronalaženje slučajeva (eng. Case Retrieval Net – CRN) je specijalna struktura podataka koja je razvijena posebno za smeštanje i pretraživanje velikih baza slučajeva. Ove mreže su u stanju da manipulišu delimično definisanim i dvosmislenim upitima, podržavaju

koncept kompletiranja slučaja i u stanju su da efikasno računaju funkciju prihvatljivosti za veliki broj slučajeva.

U tabeli 1. prikazano je pet slučajeva koji su deo baze slučajeva koji se koriste pri dijagnozi multipleks-skleroze u domenu neurologije [3]. Svaki slučaj se sastoji od 16 atributa koji predstavljaju najznačajnija opažanja u procesu dijagnoze.

Tabela 1. Neki slučajevi koji se koriste za dijagnozu multipleks-skleroze

	Case #1	Case #2	Case #3	Case #4	Case #5
I Natural History and neurological findings					
Course of disease	Relapsing /remitting	Secondary-progressive	Primary-progressive	Relapsing /remitting	Transient retrobulbar neuritis
Duration of the disease (years)	10	15	10	20	3
Motor symptoms	Mild hemiparesis	Paraparesis	Quadriparesis	Mild hemiparesis	-
Spastically	+	+	+	-	-
Cerebella symptoms	Increasing intention tremor	Ataxia	Ataxia of lower limbs, dysmetria, dysarthria	-	-
Sensory symptoms	Paresthesiae dysaesthesiae	Paresthesiae, Trigeminal neuralgia	Paresthesiae	Paresthesiae dysaesthesiae	-
Gait abnormalities	+	+	+	+	-
Bladder dysfunction	Micturition urgency	Retention	Incontinence	Micturition urgency	-
Bowel dysfunction	-	-	+	-	-
Visual symptoms	Transient optic neuritis	Diplopia	Permanent loss of visual acuity	-	Isolated retrobulbar neuritis
Cognitive dysfunction	Memory disturbance difficulty in sustaining attention	Memory disturbance	Dementia	Memory disturbance,	-
Psychiatric symptoms	Depression, Drug induced psychosis	Euphoria Pathological laughing and crying	-	Euphoria	-
II Laboratory tests in diagnosis CSF (Cerebrospinal fluid) Oligoclonal IgG bands	+	+	+	-	-
III Visual evoked potentials P100	130/121	171/176	170/165	125/121	137/130
IV Magnetic resonance imaging (TVLL total vol. lesion load) mm ³	170	920	1600	250	0 (optic nerve demil.)
Diagnosis	Definite multiple sclerosis	Definite multiple sclerosis	Definite multiple sclerosis	Probable multiple sclerosis	Possible multiple sclerosis

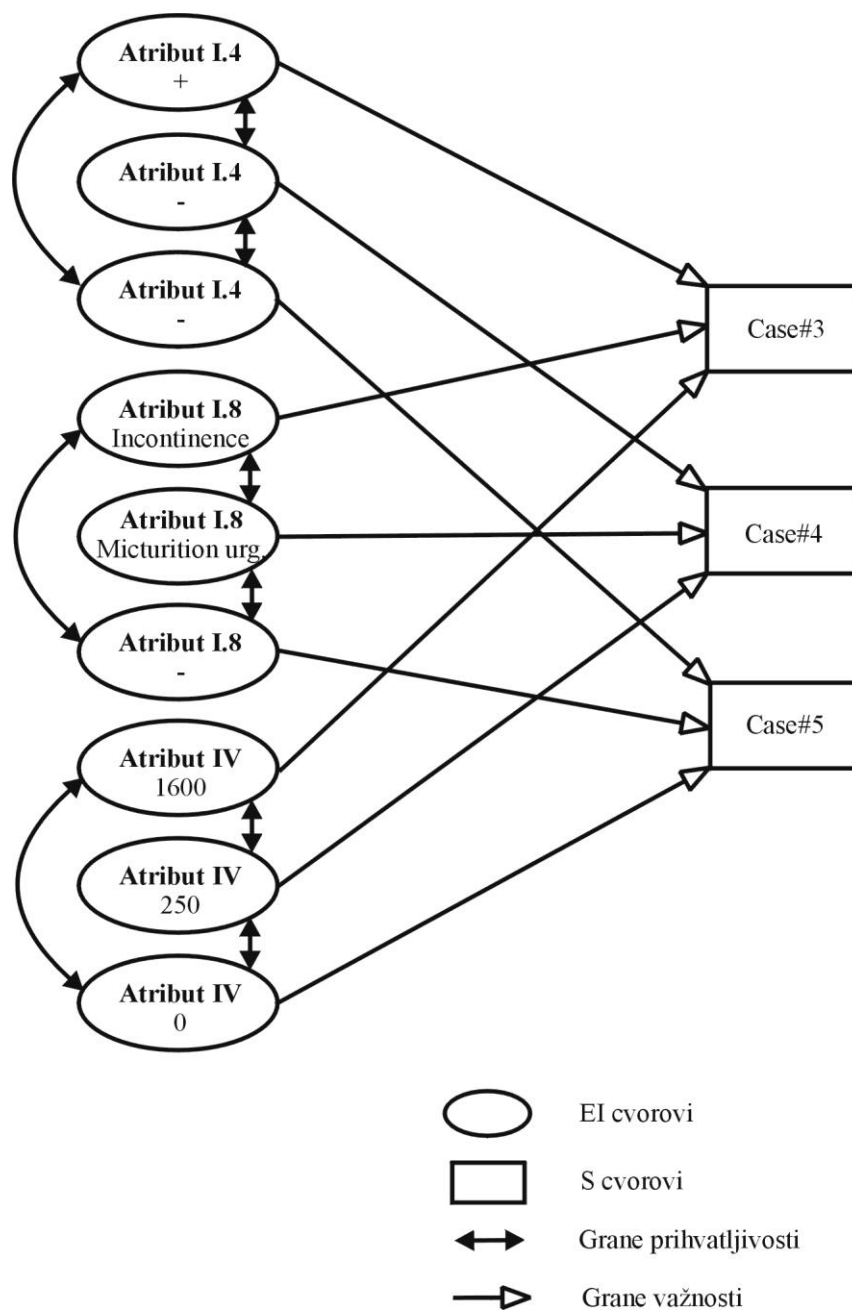
Na slici 2. je prikazan deo mreže za pronalaženje slučajeva koji je izgenerisan na osnovu tabele 1. Zbog veličine mreže, na slici se vide samo tri slučaja (*Case#3, Case#4, Case#5*) i

samo tri atributa sa po tri vrednosti (ukupno devet entiteta informacija) kao i njihov međusobni odnos.

Mreža za pronalaženje slučajeva je mrežna struktura sa:

- *entitet informacija čvorovima (EI čvor)* za svaki entitet informacije iz posmatranog domena i
- *slučaj čvorovima (S čvor)* za svaki slučaj iz domena.

Takođe, postoje *grane prihvatljivosti* od EI čvora e do EI čvora e' ako je $\sigma(e,e')$ definisano i *grane važnosti* od EI čvora e do S čvora c ako je $\rho(e,c)$ definisano. Sve grane u mreži imaju svoje težine koje odgovaraju vrednostima funkcija $\sigma(e,e')$ i $\rho(e,c)$.



Slika 2. Deo mreže za pronalaženje slučajeva.

U praksi, često je nemoguće uključiti sve entitete informacija iz domena u mrežu, jer ih obično ima beskonačno. Tada je dovoljno napraviti mrežu samo od onih entiteta informacija koji se pojavljuju u bazi slučajeva (njih ima konačan broj).

Prihvatljivost svih slučajeva za neki dati upit se računa procesom širenja aktivacije u mreži na sledeći način:

EI čvorovi su inicijalno aktivirani sa $\alpha_q(e)$ za dati težinski upit. Proces se nastavlja širenjem aktivacije kroz grane prihvatljivosti do sledećih EI čvorova, a od ovih čvorova kroz grane važnosti do S čvorova. Ovaj proces širenja je sličan procesima u neuralnim mrežama. Funkcije f i π_e su odgovorne za akumulaciju u EI čvorovima, a g i π_c za akumulaciju u S čvorovima. Konačna aktivacija u S čvorovima predstavlja vrednost prihvatljivosti slučajeva koji odgovaraju tim čvorovima za upit α_q . Tako izračunata prihvatljivost je ekvivalentna prihvatljivosti koja je izračunata korišćenjem formula iz prethodnih definicija, a dobijena je navedenim procesom u mreži koji se sastoji samo iz dve iteracije.

3. Tipovi aplikacija i primene

Zaključivanje na osnovu slučajeva je oblast veštačke inteligencije kome se u poslednje vreme pridaje velika pažnja i gde je za kratko vreme realizovan velik broj aplikacija u najrazličitijim oblastima. U ovom odeljku će biti objašnjene osnovne karakteristike za svaki tip aplikacije i biće navedeno nekoliko realizovanih aplikacija sa linkovima koji pružaju više informacija o njima [6].

Pre nego što damo prikaz oblasti najpogodnije primene ove tehnike potrebno je napraviti razliku između pojma *domena* i pojma *tipa zadatka*.

Domen primene ove tehnike predstavlja oblast ili disciplinu u kojoj se tehnika koristi za rešavanje problema. Tako su domeni na primer: mehaničko inženjerstvo, medicina, poslovna administracija itd. Svaki domen sadrži neke svoje posebne karakteristike. Domen strogo određuje izbor tipova podataka u aplikaciji.

Tip zadatka predstavlja vrstu problema koji je potrebno rešiti u nekom domenu. Tako su tipovi zadatka na primer: klasifikacija, dijagnoza, planiranje itd. On određuje tip problema i rešenja kao i aktivnosti potrebne za rešavanje. Ne postoji obostrano jednoznačno preslikavanje između domena i tipa zadatka, nego je moguć svaki uređeni par (domen, tip zadatka) i on zahteva svoju posebnu ekspertizu.

Potencijalne oblasti primene CBR tehnike su: *klasifikacija, dijagnoza, konfiguracija i dizajn, planiranje, donošenje odluka i pretraživanje i pronalaženje informacija*.

3.1. Klasifikacija

Klasifikacija obuhvata postupak određivanja klase za pojedine elemente iz nekog univerzalnog skupa, korišćenjem odgovarajućih funkcija.

U procesu klasifikacije se razmatra univerzalni skup U i podskupovi $K_i \subseteq U$, $i \in I$, koji se nazivaju klase. Klasifikator je funkcija

$$f: U \rightarrow I$$

tako da $f(x) = i$ implicira $x \in K_i$

U CBR tehnici klasifikator se definiše kao uređeni par (CB, sim) , gde je $CB \subseteq U$, a sim predstavlja funkciju sličnosti definisanu na $U \times CB$, a CB bazu slučajeva. Ako su klase elemenata u bazi slučajeva CB poznate, tada se klasa za bilo koji element iz univerzalnog skupa $x \in U$ izračunava na sledeći način:

$$x \in K_i \Leftrightarrow NN(x) \in K_i$$

gde je $NN(x)$ najbliži sused od x , ali u skupu CB .

Proces klasifikacije je u velikoj meri deterministički, pa samim tim nije privlačan za realizaciju u CBR-u. Stoga u ovoj oblasti nisu realizovane neke značajnije aplikacije.

3.2. Dijagnoza

Moglo bi se reći da je dijagnoza klasifikacija sa ocenom i nepotpunom informacijom. Ovo znači da je postavljanje dijagnoze samo završni korak u procesu u kojem je potrebno izvršiti još niz manjih podzadataka. Podaci za dijagnozu su vrednosti nekih atributa koji su dobijeni posmatranjem, postavljanjem upita i analizom testova. U procesu dijagnoze mogu se tražiti i neki dodatni testovi da bi se upotpunile informacije i da bi se dobio kvalitetniji rezultat.

Neke komercijalne aplikacije za dijagnozu, koje su realizovane CBR tehnikom su navedene u nastavku:

- **Case Advisor 4 / Webserver** – dijagnoza kvarova vezanih za PC računare. Case advisor 4 je alat koji omogućava dijagnozu i rešavanje problema vezanih za PC računare. Ovaj alat koristi statičku bazu znanja. Autori ovog alata dozvoljavaju njegovu besplatnu nekomercijalnu upotrebu. Case Advisor Webserver je alat koji omogućava online podršku korisnicima nekih uređaja. Ovde je navedena pomoć korisnicima pri upotrebi kablovske televizije. Detaljnije informacije se mogu naći na adresi:
<http://www.cs.sfu.ca/~isa/isaresearch.html#systems>
- **Case-Based Reasoning in Cardiovascular Disease** – medicinska dijagnoza. Ovaj sistem pokušava da pronađe dijagnozu (nekih kardiovaskularnih bolesti) na osnovu fizioloških simptoma. Dijagnoza se traži na osnovu baze slučajeva koja se sastoji od 240 karakterističnih slučajeva. Takođe, ovaj sistem podržava učenje iz iskustva. Detaljnije informacije se mogu naći na adresi:
<http://medg.lcs.mit.edu/projects/cbr.html>
- **MoCas** – dijagnoza u tehničkim domenima. Kod ovog sistema je karakteristično da se znanje iz nekog tehničkog domena integriše u sistem. Integracija znanja omogućava adaptaciju i transformaciju slučajeva. Na ovaj način se baza slučajeva

koristi značajno efikasnije. Detaljnije informacije se mogu naći na adresi:

http://www.wagr.informatik.uni-kl.de/~lsa/LSABook-English/LSABook-E_20.html

- **SpectroRx Resolution Expert** – alat za dijagnozu problema sa kompjuterskom mrežom. Kompanija «ENTERASYS», koja se bavi proizvodnjom mrežne opreme, upotrebila je CBR tehniku da bi obezbedila svojim klijentima online sistem za pomoć pri različitim problemima. Ovaj sistem značajno štedi vreme eksperata, jer eksperti pomažu klijentima samo kad sistem ne može, a ostalo vreme troše na bitnije stvari. Detaljnije informacije se mogu naći na adresi:
<http://www.cabletron.com/products/items/SA-CSII016/>

3.3. Konfiguracija i dizajn

Pod *konfiguracijom* se podrazumeva konstrukcija nekog predmeta iz datog skupa komponenata, ali tako da su zadovoljeni određeni uslovi. *Dizajn* unosi određeni stepen kreativnosti jer je u procesu dizajniranja moguće da neke komponente nisu eksplicitno zadate. U zavisnosti od stepena kreativnosti možemo razlikovati tri vrste dizajna: *rutinski*, *inovativni* i *kreativni*. Kod dizajna iskustvo igra značajnu ulogu. U realnim aplikacijama rešenje dobijeno CBR konfiguracijom ili dizajnom skoro uvek mora biti modifikovano od strane eksperta za konkretan problem.

Neke komercijalne aplikacije za konfiguraciju i dizajn, koje su realizovane CBR tehnikom su navedene u nastavku:

- **AIDA** – alat za pomoć u dizajnu aviona. AIDA (Artificial Intelligence supported Design of Aircraft) je alat za pomoć u prvoj fazi dizajniranja aviona – faza konceptualnog dizajniranja. Uz pomoć ovog alata dizajner može da posveti više pažnje kreativnosti, dok bi alat obavljao neke manje kreativne zadatke. Za konstrukciju ovog alata korišćene su različite tehnike veštačke inteligencije: Constraint-Based Reasoning, Case-Based Reasoning i Rule-Based Reasoning. Namera je da se koncept AIDA-e iskoristi i u drugim domenima kao što su dizajn brodova ili automobila. Detaljnije informacije se mogu naći na adresi:
<http://www.kbs.twi.tudelft.nl/Research/Projects/AIDA/>
- **Archie** – alat za pomoć u konceptualnom dizajnu građevina. Cilj ovog alata je da se iskoristi iskustveno znanje u dizajniranju javnih ustanova tako da buduće arhitektae izbegnu ponavljanje grešaka, ali i da imaju koristi od uspešnih inovacija. Archie počinje proces dizajna sa dizajniranjem temelja na osnovu funkcionalnosti koje treba da poseduje zgrada. Detaljnije informacije se mogu naći na adresi:
<http://www.cc.gatech.edu/aimosaic/faculty/kolodner/archie.html>
- **BRUSH** – alat za dizajn kupatila za invalide. BRUSH je alat za selektovanje i manipulisanje dizajnerskim slučajevima. Slučajevi u ovom alatu predstavljaju epizode u dizajnu kupatila za invalide. Sistem na osnovu ovih epizoda dizajnira novo kupatilo koje ima određene funkcionalnosti. Detaljnije informacije se mogu naći na adresi:
<http://www.arch.su.edu.au/~kate/BathRedesign/Guestbookii/Welcome.html>
- **CADET** – konceptualni dizajn elektro-mašinskih delova. CADET (**C**ase-based **D**esign **T**ool) je sistem koji pomaže u početnoj fazi dizajniranja (konceptualni dizajn) elektro-mašinskih delova. CADET se sastoji od više podsistema koji se nazivaju CARD (Case-based Retrieval for Design). U ovom sistemu su integrisani

CBR i MBR (Model-Based Reasoning). Detaljnije informacije se mogu naći na adresi: <http://www.cs.cmu.edu/afs/cs.cmu.edu/project/cadet/ftp/docs/CADET.html>

- **CBRTeam** – multiagentni sistem za dizajn. CBRTeam je sistem za dizajn koji koristi skup kooperativnih agenata za dizajniranje kondenzatora pare. Konkretno, realizovan sistem se sastoji od tri agenta *motor-agent*, *pump-agent* i *vbelt-agent* koji su zaduženi za dizajniranje motora, pumpe i kaiševa respektivno. Kada korisnik zada specifikaciju ovi agenti u saradnji i na osnovu raspoloživih delova pokušavaju da konstruišu odgovarajući kondenzator. Detaljnije informacije se mogu naći na adresi:
http://ksi.cpsc.ucalgary.ca/KAW/KAW96/prasad/subsubsection3_4_2_3.html

3.4. Planiranje

Pod terminom *planiranje* podrazumeva se širok pojam najrazličitijih zadataka. Ipak, najčešće se pod planiranjem podrazumeva *planiranje akcija*. Osnovni problem pri tome je naći niz akcija koje transformišu početnu situaciju u neku željenu tj. ciljnu situaciju. Veličina niza akcija nije ograničena, ali je skup akcija konačan.

Neke komercijalne aplikacije za planiranje, koje su realizovane CBR tehnikom su navedene u nastavku:

- **Bioplan** – planiranje bioprocesa u proizvodnji lekova. Detaljnije informacije se mogu naći na adresi: <http://www.vtt.fi/bel/bio/process/bioplan.htm>
- **Knowledge Based Mashing** – planiranje procesa proizvodnje piva. Detaljnije informacije se mogu naći na adresi: <http://www.vtt.fi/bel/bio/process/mashplan.htm>
- **Prodigy** – planiranje akcija nezavisno od domena. Prodigy je arhitektura za planiranje i učenje. Trenutno rad na Prodigy-u podržava razne koncepte među kojima su: učenje na primerima, parcijalno izračunavanje, grafičko učenje, automatsko apstrahovanje, inicijalno planiranje, CBR kao i velik broj domena. Detaljnije informacije se mogu naći na adresi:
<http://www.cs.cmu.edu/afs/cs.cmu.edu/project/prodigy/Web/prodigy-home.html>
- **CHARADE** – planiranje ljudskih i materijalnih resursa u vanrednim situacijama (požar) . Detaljnije informacije se mogu naći na adresi:
<http://mnemosyne.itc.it:1024/avesani/html/charade.html>

3.5. Donošenje odluka

Uloga aplikacije za donošenje odluka je da pomogne onome ko donosi odluke, a ne da ga u potpunosti zameni. Odgovor sistema obično nije rešenje nego savet ili neka korisna informacija.

Neke komercijalne aplikacije za donošenje odluka (decision support), koje su realizovane CBR-om su:

- **Aircraft Conflict Resolution** – aplikacija za pomoć pri kontroli letova. Ovaj alat omogućava pomoć pri rešavanju konflikata nastalih u avionskom saobraćaju. Poseban značaj ove aplikacije je u tome što su autori posvetili veliku pažnju

proučavanju raznih vrsta reprezentovanja slučajeva. Detaljnije informacije se mogu naći na adresi:

http://www.cs.tcd.ie/research_groups/aig/cbr.html

- **ALSTOM** – alat za povećanje iskorišćenosti vozova radi smanjenja troškova. Detaljnije informacije se mogu naći na adresi:
<http://www.acknosoft.com/alstom.html>
- **ANSALDO** – održavanje metroa u Napulju. Detaljnije informacije se mogu naći na adresi:
<http://www.acknosoft.com/ansaldo.html>
- **CBR Job Agent** – pronalaženje optimalnog posla na osnovu unetih sposobnosti. Detaljnije informacije se mogu naći na adresi:
<http://minsk.informatik.uni-kl.de:8100/launch/JobbQueryInterface>
- **DESSERT** – donošenje odluka u uslužnom menadžmentu. Detaljnije informacije se mogu naći na adresi:
<http://www.broadcom.ie/partners/acts/race/dessert/dessert.html>

3.6. Pretraživanje i pronalaženje informacija

Nekada se pod pronalaženjem informacija podrazumevala pretraga za određenim dokumentom u nekom skupu dokumenata. Danas je pronalaženje konkretne informacije centralna stvar dok je pronalaženje dokumenata samo koristan korak u rešavanju problema. Izuzetno značajan i skoro nezaobilazan primer primene je pretraga na Internetu.

Netačni pogoci (pronalaženje sličnih dokumenata ako ne postoji baš traženi) su ključni u ovoj oblasti, pa je CBR prirodna tehnika za pronalaženje informacija. Zbog eksponencijalnog rasta informacija ovo je obećavajuće polje za CBR, jer CBR tehnika poseduje mogućnost konstantnog učenja i dobar mehanizam za pretraživanje velikih baza podataka.

Neke realizovane komercijalne aplikacije za pronalaženje informacija, koje su realizovane CBR tehnikom su navedene u nastavku:

- **Broadway** – inteligentan pretraživač za Internet. Broadway je inteligentan pretraživač za Internet. On pokušava da iskoristi znanje naučeno tokom prošlih pretraga jedne grupe korisnika. Ovaj pretraživač prati korisnika tokom surfovanja Internetom, pokušava da otkrije njegove ciljeve i savetuje ga o potencijalno zanimljivim dokumentima. Detaljnije informacije se mogu naći na adresi:
<http://www-sop.inria.fr/aid/broadway/>
- **CaBaTa** – pronalaženje turističkih aranžmana iz kataloga. Ovaj sistem predstavlja virtuelnu turističku agenciju. Sistem, na osnovu korisnikovih zahteva, predlaže više najslabijih turističkih aranžmana tako da korisnik dobija najslabije ponude iz celog sveta. Takođe, ovaj sistem poseduje mogućnost prodaje avionskih karata. Detaljnije informacije se mogu naći na adresi:
http://www.informatik.hu-berlin.de/~lenz/CheckIn/CABATA/cabata_e.html
- **Entree** – alat za pronalaženje po raznim kriterijumima, odgovarajućeg restorana u Čikagu. Detaljnije informacije se mogu naći na adresi:
<http://infolab.cs.uchicago.edu/entree/>

- **RECALL** – automatsko čuvanje i pronalaženje 'naučenih lekcija' u «NASA Goddard Space Flight Center». Detaljnije informacije se mogu naći na adresi: <http://hope.gsfc.nasa.gov/RECALL/homepg/recall.htm>
- **FAQ Finder** – pronalaženje 'često postavljanih pitanja'. Detaljnije informacije se mogu naći na adresi: <http://infolab.cs.uchicago.edu/faqfinder/>

4. Zaključak

Zaključivanje na osnovu slučajeva je tehnika za rešavanje problema koja se u mnogo aspekata razlikuje od ostalih pristupa u veštačkoj inteligenciji.

Umesto da se oslanja isključivo na generalnom znanju iz domena problema, ili da pravi generalnu vezu između problema i rešenja, CBR je u stanju da koristi specifično znanje prethodno rešenih, konkretnih situacija.

Druga značajna razlika je u tome što je CBR pristup inkrementalnom, konstantnom učenju jer se novo znanje integriše u sistem svaki put kada je problem uspešno rešen. Pristup tom novointegrisanom znanju omogućeno je nesmetano pri rešavanju svakog narednog problema.

Primena CBR tehnike beleži konstantan rast u poslednjih nekoliko godina što se može videti po povećanom broju radova na značajnijim konferencijama, raspoloživim komercijalnim alatima i uspešnim aplikacijama u svakodnevnoj upotrebi.

Literatura

1. Aamodt A., Plaza E., (1994), "Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches", *AI Commutations*, pp. 39-58.
2. Iglezakis I. (2001), "The Conflict Graph for Maintaining Case-Based Reasoning Systems", *4th International Conference on Case-Based Reasoning (ICCBR 2001)*, pp. 263-276, Vancouver, Canada, July/August 2001.
3. Ivanović M., Kurbalija V., Budimac Z., Semnic M. (2002), "Role of Case-Based Reasoning in Neurology Decision Support", *Fifth Joint Conference on Knowledge-Based Software Engineering (JCKBSE 2002)*, Maribor, Slovenia, 2002, in print.
4. Lenz M., Brtsch-Sporl B., Burkhard, H., Wess, S. (1998), *Case-Based Reasoning Technology: From Foundation to Applications*, Springer, 1998.
5. Reinartz T., Iglezakis I., Roth-Bergofer T., (2000), "On Quality Measures for Case Base Maintenance", *5th European Workshop (EWCBR 2000)*, pp. 247-260, Trento, Italy, September 2000.
6. Case-Based Reasoning homepage at the University of Kaiserslautern <http://www.cbr-web.org/>