# **Pronalazivač studentskog smještaja**

#### Uvod

Cilj seminarskog rada je izrada ekspertnog sustava za preporuku studentskog smještaja prvotno za studente koji tek upisuju fakultet, ali i za starije studente koji su nezadovoljni sadašnjim smještajem. Naime jedan od glavnih problema brucoša nakon odabira i upisa fakulteta je odabir odgovarajućeg smještaja u slučaju da im je mjesto studiranja udaljeno od mjesta stalnog stanovanja. Navedeni sustav za preporuku studentskog smještaja izgrađen je za grad Zagreb. U ponudi se nalaze studentski smještaj odnosno studentski domovi koji su pod nadležnosti Studentskog centra ujedno i pristupačniji cijenom, zatim učenički domovi koji primaju određeni broj studenata te privatni smještaj za one koji si to mogu priuštiti i kojima takav način stanovanja najviše odgovara. U sljedećim poglavljima dan je kratak opis korištenog sustava te njegova primjena pri rješavanju pronalazivača studentskog smještaja.

#### Ekspertni sustav s pravilima - CLIPS

Pri rješavanju navedene problematike korišten je ekspertni sustav s pravilima - CLIPS. Ekspertni sustav s pravilima omogućuje heuristički prikaz znanja odnosno skup akcija se izvodi ovisno o danoj situaciji kroz definiranje pravila i činjenica. Pravila, odnosno znanja o domeni, sastoje se od lijevog dijela (AKO) koji je uzrok odnosno pretpostavka i desnog dijela (ONDA) koji čini određeni zaključak. Činjenice koje se dobivaju od korisnika čine tzv. "sliku svijeta" te je potrebno postići njihovo slaganje s AKO stranama pravila što dovodi do izvođenja određenog pravila (ONDA dio pravila).

#### Primjena ekspertnog sustava s pravilima

Budući da se koristi navedeni sustav potrebno je najprije definirati pravila. Pravila se definiraju na osnovi cijene, kapaciteta unutar smještajne jedinice, načinu korištenja zajedničkih prostorija i sl. Primjer pravila definiranog za smještaj Studentski dom Stjepan Radić (kategorija 1-1) iz baze dan je kodom Table.

|  |
| --- |
| (defrule stjepanradick11 ""  (vrsta ?vrst)  (or (test (= 100 ?vrst)) (test (= 10000 ?vrst)) )  (cijena ?cije)  (or (test (>= ?cije 1000)) (test (= 10000 ?cije)) )  (studinval ne)  (velicina ?vel)  (or (test (<= ?vel 300)) (test (= 10000 ?vel)) )  (kupaonica ?kup)  (or (test (<= ?kup 200)) (test (= 10000 ?kup)) )  (kuhinja ?kuh)  (or (test (<= ?kuh 200)) (test (= 10000 ?kuh)) )  (namjestenost ?namj)  (or (test (<= ?namj 300)) (test (= 10000 ?namj)) )  (menza ?menz)  (or (test (<= ?menz 200)) (test (= 10000 ?menz)) )  (praonica ?prao)  (or (test (<= ?prao 200)) (test (= 10000 ?prao)) )  (trgovina ?trg)  (or (test (= ?trg 100 )) (test ( = 10000 ?trg)))  (sport ?spor)  (or (test (<= ?spor 200)) (test ( = 10000 ?spor)))  (stanari ?stan)  (or (test (= ?stan 100)) (test ( = 10000 ?stan)))  =>  (printout t crlf)  (printout t "Smjestaj: Studentski dom Stjepan Radic Kategorija 1 Vrsta 1" crlf)  (assert (smjestaj stjepan-radic-k1-1))) |

Kod 1: Pravilo SDSRK11

Agent potražuje informacije od korisnika kroz 12 pitanja o osnovnim preferencijama tipa stanovanja korisnika. Pitanja su osnovnog tipa od maksimalne cijene koju je korisnik mjesečno spreman izdvojiti za troškove stanovanja, zatim o željenom broju stanara, sanitetskim čvorovima i sl. Kod Table prikazuje pravilo za dobivanje informacije o željenom broju stanara unutar smještajnog objekta s 4 ponuđena odgovora. Gotova sva pravila za potraživanje informacija od korisnika su konstruirana na način da sadrže više ponuđenih odgovora.

|  |
| --- |
| (defrule getBrojStanara ""  (declare (salience 5))  =>  (printout t crlf "Broj stanara:" crlf)  (printout t " a) 1" crlf)  (printout t " b) 2" crlf)  (printout t " c) >=3" crlf)  (printout t " d) nije vazno" crlf)  (bind ?response (ask-question "Odabir (a/b/c/d):" a b c d))  (switch ?response  (case a then (bind ?response 100))  (case b then (bind ?response 200))  (case c then (bind ?response 300))  (case d then (bind ?response 10000))  (default none))  (assert (stanari ?response))) |

Kod 2: Broj stanara

Kada korisnik odgovori na 12 postavljenih pitanja stvorena je tzv. "slika svijeta" odnosno moguće je, korištenjem prethodno definiranih pravila, promatrati njihovo poklapanje s dobivenim činjenicama. Npr. korisnik je odabrao da želi veličinu sobe "velika", a veličina sobe je opisana kako je prikazano izrazom Text.

|  | (1) |
| --- | --- |

Definirana su pravila koja, između ostaloga, opisuju i sve moguće smještaje po veličini sobe, npr. smještaj Studentski dom Stjepan Radić (kategorija 3-1) pruža uslugu sobe veličine – "mala" te slijedi da je ova opcija smještaja eliminirana jer se ne poklapaju činjenica i pravilo, Kod Table.

|  |
| --- |
| (defrule stjepanradick31 ""  (...)  (velicina ?vel)  (or (test (= ?vel 100)) (test (= 10000 ?vel)) )  (...)  =>  (printout t crlf)  (printout t "Smjestaj: Studentski dom Stjepan Radic Kategorija 3 Vrsta 1" crlf)  (assert (smjestaj stjepan-radic-k3-1))) |

Kod 3: Kod1

Nadalje promotri li se npr. smještaj Studentski dom Stjepan Radić (kategorija 1-1) vidljivo je da takav smještaj pruža sobu veličine – "velika" i prema činjenici dobivenoj od korisnika ovaj smještaj po ovom dijelu pravila prolazi dalje. Agent također dopušta da se korisniku koji je odabrao npr. veličinu sobe – "srednja" omogući da se takva činjenica poklapa i s onim smještajima koji pružaju veličinu sobe – "velika" ili "ogromna". Naime iako je korisnik odabrao željenu veličinu sobe, agent dopušta da, ako postoji smještaj koji pruža uslugu još veće sobe, koja istovremeno odgovara i ostalim preferencijama korisnika (cijena, blizina menze, blizina trgovine itd.) ponudi i to smještajno rješenje. Za dani smještaj (koji pruža uslugu veličine sobe "velika") definira se pravilo koje omogućuje da se to pravilo poklopi i s činjenicom koja za odabir veličine sobe ima "mala", "srednja" ili "nije važno", kod Table.

|  |
| --- |
| (defrule stjepanradick11 ""  (...)  (velicina ?vel)  (or (test (<= ?vel 300)) (test (= 10000 ?vel)) )  (...)  =>  (printout t crlf)  (printout t "Smjestaj: Studentski dom Stjepan Radic Kategorija 1 Vrsta 1" crlf)  (assert (smjestaj stjepan-radic-k1-1))) |

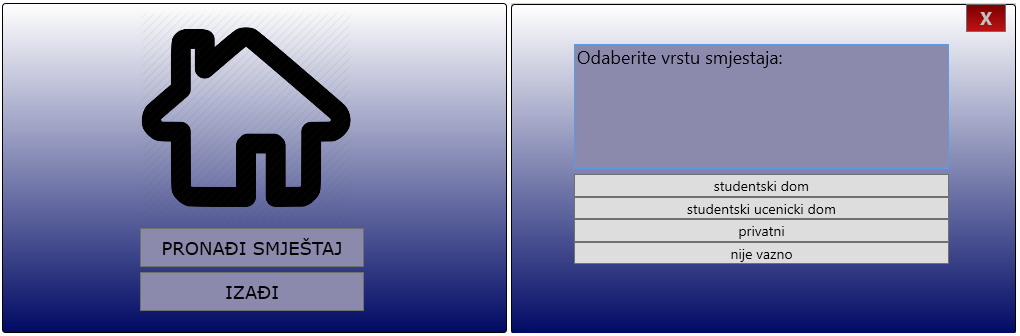
Kod 4: Kod2

Nadalje promotri li se dio pravila koji opisuje vrstu željenog smještaja: studentski, učeničko/studentski ili privatni smještaj vidljivo je da činjenica npr. odabran je studentski smještaj automatski eliminira mogućnost prikaza ostala dva smještaja kao rješenja.

U bazi ponuđenih smještaja se trenutačno nalazi ukupno 20 različitih kategorija u studentskim domovima, 3 učeničko/studentska doma te 7 privatnih smještaja.

#### Implementacija grafičkog korisničkog sučelja

Sučelje je napravljeno u Microsoftovoj tehnologiji Windows Presentation Fundation (WPF) i programskom jeziku C\#. Da bi se sustav CLIPS mogao integrirati u ovu tehnologiju, bilo je potrebno koristiti programsku knjižicu (eng. library) CLIPS.NET. Ova knjižica omogućava učitavanje .clp datoteke te interakciju sa skupom dobivenih pravila i činjenica.



Slika 1: Izgled grafičkog korisničkog sučelja

##### Stanje sučelja

Važan dio .clp datoteke koja se koristi uz pomoć CLIPS.NET je definiranje stanja u kojem se korisničko sučelje nalazi. To se postiže pomoću predložaka (eng. templates) koji su usporedivi sa strukturama u programskom jeziku C. Ovdje korišteni predložak naziva se UI-State koji u sebi, između ostalih, ima slijedeće članove:

* *valid-answers* (Mogući odabiri za trenutno stanje.)
* *display* (Poruka za prikaz u trenutnom stanju, najčešće pitanje.)
* *coded-answers* (Kodirani mogući odgovori (ove vrijednosti se postavljaju kao činjenice).)

Tu su još i drugi članovi koji služe primarno za ispravan rad sa grafičkim korisničkim sustavom, kao naprimjer *id* vrijednost.

|  |
| --- |
| (deftemplate UI-state  (slot id (default-dynamic (gensym\*)))  (slot display)  (slot relation-asserted (default none))  (slot response (default none))  (multislot valid-answers)  (multislot coded-answers)  (slot state (default middle))) |

Kod 5: Stanje sučelja

##### Tipovi pravila

Pravila se dijele na tri skupine. Prva skupina je skupina pravila za početak izvođenja. Ova skupina u sebi ima samo jedno pravilo i ono se uvijek pokreće prvo te postavlja činjenicu koja omogućava daljnje izvođenje pravila. Druga skupina je skupina pravila upita. Ovdje se zapravo nalazi svako pitanje koje je postavljeno korisniku prilikom korištenja sustava, te se ovdje grade činjenice koje su potom korištene u zadnjoj skupini pravila. Ta, treća skupina, je namijenjena donošenju odluka. Ovo je ujedno i jedina grupa pravila gdje se nakon korisnikove interakcije pokreće više pravila. Tako dobivamo mogućnost više preporuka smještaja za pojedini unos zahtjeva.