Sistem za preporuku i savete pri heklanju

Autor:

Katarina Mirković SV6/2020

Motivacija:

U poslednjih nekoliko godina heklanje je ponovo postalo popularno kao hobi, ne samo među starijima, nego i mlađima. Nekome ono predstavlja čistu razonodu I opuštanje, dok se drugi odluče heklanjem da zarađuju džeparac, prodavajući svoje kreacije. Mnogi bi iz istih razloga hteli da se oprobaju o ovu veštinu, ali nemaju konkretan plan, niti ideju šta bi hteli ili mogli da naprave. Čak se može desiti i da znaju šta bi otprilike hteli, ali nisu upućeni koliko i kojeg materijala im treba, niti da li njihova želja odgovara njihovim trenutnim sposobnostima.

Pregled problema:

Tu u pomoć dolazi ova aplikacija. Namenjena je kako početnicima, tako i naprednijim heklašima, da dobijaju preporuke šta da heklaju, šta im je sve potrebno, kao i sistem uz koji mogu da prate svoj napredak.

Često početnici dobiju ideju sa interneta šta bi hteli da heklaju, ali ne znaju šta im je sve potrebno. U tom slučaju bi im sistem na osnovu željenog šablona (igračka, odeća) preporučio koja im vunica i koliko treba, te koji pribor za to ide.

Ako korisniku treba ideja šta da hekla, a ima već nekog starog materijala pri ruci, može da unese koliko kojeg materijala ima, pa će mu sistem na osnovu njegovog iskustva i preferiranog heklanja preporučiti šta bi mogli da heklaju.

Baza znanja:

Kao osnova baza znanja čuvali bi se nekoliko patterna za heklanje, koji svaki zahteva određeni nivo znanja (beginner, intermediate, advanced), određenu vrstu i količinu vunice, kao i odgovarajuću heklaricu. Heklanje samo ima više vrsta (amigurumi, micro crochet, tunisian, basic), od kojih svaki može imati podvrstu stvari koje tim tipom želimo heklati (aksesoari, odjeća, posteljina, plišane životinjice), i svakoj od kategorija pripadaju određeni patterni. Jedan pattern ne može da pripada više od jednoj kategoriji.

**Forward chaining**

Za forward-chaining bi se koristila pravila koja kombinuju sve potrebne materijale koje je korisnik naveo da ima pri ruci, i pronalazi se pattern koji odgovara tim materijalima i nivou znanja. U slučaju da se korisniku taj pattern svideo, može da ga favoritizuje, ili čak i celu vrstu patterna (igračke, majice, odeća generalno) da lajkuje. Za ova pravila se lako može iskoristiti **templates** za svaki ponuđeni pattern. Primer template:

when   
**Difficulty = lvl** and **Vunica = type** and **Kolicina = x**   
and **heklarica in (0.5, 1.0, 1.5)** and **Not done**  
then  
**pattern**

Ovaj template predstavlja prvi korak u lancu. Ako je pronađen pattern koji odgovara uslovima, poziva se sledeći korak. Drugi korak u lancu jeste pokretanje **BC** koji proverava da li korisnik pronađeni pattern ne voli da hekla, ako je bilo koji član u povratnom lancu *dislike-*ovan, onda se treba odbaciti taj pattern i ponovo pokrenuti prvi korak. Kada se pronađe prihvatljiv pattern, on se korisniku preporuči u interfejsu.

Ako korisnik prihvati preporučeni pattern, sledeći korak u sistemu će bit da se taj pattern označi kao „urađen“. U slučaju da korisnik nije prihvatio pattern, onda se on treba setovati da je *dislike*-ovan.

Korisnik odlaskom na svoj profil imaće uvid u svoje statistike heklanja. Može da vidi koliko projekata je isheklao do sada. Tu funkcionalnost ćemo obezbediti uz pomoć accumulate funkcije koja će da prebrojava koliko ima završenih patterna u sesiji.

Takođe kako bi korisnik pratio koliko je heklao u toku nedelje, čuvaće se činjenice o vremenskim intervalima heklanja u sesiji, tako što ćemo na svaki Event kada korisnik prekine heklanje, izračunati koliko je heklao od starta do pauze.

Funkcija **accumulate** će da sračuna koliko je korisnik ukupno heklao, pod uslovom da su se ti periodi heklanja desili u prethodni dan, 7 dana ili mesec dana, te će imati uvid u koje to on doba dana najviše hekla. Npr:

rule "Count hours crocheted past 7 days“  
 no-loop true  
 when  
 $c: CrochetTime($minutes: minutesCrocheted, date > currentDate.minusDays(7) )  
 $m: Number() from accumulate(  
 init(int count = 0;),  
 action(count += $minutes;),  
 reverse(count -= $minutes;),  
 result(count)  
 )  
 then  
 System.out.println($n);  
end

**Backward chaining**

Za backward-chaining bi se iskoristili prethodno sačuvani podaci o ne favoritnim patternima, tako da kada se sledeći put preporuči pattern, proverava se da li je favorit (\*\*\*Ili da radim obrnuto, ako je nesto dislikeovano, da se ono ne preporučuje?\*\*\*). U ovom slučaju bi postojalo stablo sa korenom heklanje, koje je svakako favorit, heklanje može imati više podvrsta heklanja, koje mogu da se lajkuju, te svaka podvrsta može imati nivo kompleksnosti (onaj koji smo izabrali u pocetku, on je lajkovan), takođe svaki pattern određene kompleksnosti može posebno da se lajkuje.

Dakle stablo bi bilo:  
 heklanje, liked

/ \

Igracke, liked odeca, liked

/ | \ | \ \

Zivotinje, liked Vocke, liked Igracke Kapa, Dzemper, carape,

za ljubimce, liked liked liked….. itd

liked

**CEP**

Nakon odabira patterna za heklanje, korisnik može označiti da je započeo heklanje (event). Ako nakon početka heklanja u roku od 2 sata se ne pojavi event za prekinuto heklanje, korisnik se treba obavestiti da je vreme da napravi pauzu. Ako nakon 10 minuta od eventa za upozorenje ne dodje do pojave prekida heklanja, treba ponovo da se opomene korisnik da napravi pauzu. U slučaju da korisnik i dalje ne napravi pauzu, treba da se okine Event da se automatski prekine praćenje heklanja.