SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4350

Razvoj podatkovnog sloja i aplikacijske logike za potrebe sustava elektroničkog učenja

Alen Murtić

Zagreb, lipanj 2016.

Umjesto ove stranice umetnite izvornik Vašeg rada.
Da bi ste uklonili ovu stranicu obrišite naredbu `\izvornik`.

SADRŽAJ

1. Uvod **1**

2. Sustavi za elektroničko učenje **2**

2.1. Inteligentni sustavi za elektroničko učenje (ITS) 3

2.1.1. Povijest ITS-a i zaključci iz nje 3

2.1.2. Komponente ITS-a 6

2.1.3. Karakteristike dobrih sustava 8

2.1.4. Algotimi sustava za inteligentno učenje 11

2.1.5. Poznati sustavi 13

2.1.6. Nedostaci ITS-a 18

3. Slojevi aplikacija **20**

4. Opis implementiranog rješenja **22**

4.1. Korištene tehnologije 22

4.2. ER dijagram 22

4.2.1. Slika ER dijagrama i entiteta u bazi 23

4.3. Mogućnosti sustava 25

4.3.1. Reprezentacija znanja 25

4.3.2. Parametrizirani zadaci 27

4.3.3. Mogućnosti administratora 28

4.3.4. Mogućnosti korisnika 29

4.4. Logika posluživanja pitanja 29

5. Zaključak **30**

1. Uvod

Znanje i prenošenje znanja najznačajniji je faktor sve bržeg razvoja ljudske civilizacije, iako je često nailazilo na prepreke u širenju. Prije početka 20. stoljeća ponajveći je problem bila dostupnost znanja, a ta je prepreka postajala sve manja izumima kao što su tiskarski stroj, sveučilište te obveznim školstvom i masovnom proizvodnjom knjiga. Dostupnog znanja danas je minoran ili čak nepostojeći problem, no to je stvorilo novi izazov - odabir pravog načina učenja. Moderno elektroničko doba samo ga je još pojačalo jer je ogromna količina e-knjiga od korisnika udaljena na samo nekoliko klikova.

Moderni svijet donosi golemu količinu dostupnih informacija što je dovelo do toga da je ljudska pažnja sve neodređenija, a vrijeme fokusa sve kraće. Moderne generacije nemaju imperativ učenja kao njihovi preci, uglavnom zato što nikada nisu iskusile težinu ne-modernog života. Stoga se može dogoditi da čovjek željan znanja jednostavno odustane zbog ogromne količine mogućnosti, a posebno zbog činjenice se samo u ponekim školskim sustavima uči kako učiti.

Shvativši da je znanje i prenošenje znanja izuzetno bitno te da je moderni svijet stvorio nove izazove učenju, postavlja se pitanje kako poboljšati njegovu kvalitetu? Odgovor je relativno jednostavan: pretvoriti učenje u nešto zanimljivo i jednostavno za korištenje, ali ipak izazovno i korisno - specijalizirane aplikacije, tj. sustave za elektroničko učenje.

Ovaj završni rad opisat će što su sustavi za elektroničko učenje, što znači da je takav sustav inteligentan, neke od algoritama inteligencije, kako trebaju izgledati podatkovni i aplikacijski slojevi takvog sustava te objasniti što su oni uopće. U konačnici, temeljem principa navedenih u teoretskom dijelu, opisat će podatkovni i aplikacijski sloj sustava koji sam implementirao.

2. Sustavi za elektroničko učenje

Ideja sustava za elektroničko učenje je ujediniti postupak prenošenja informacija, upijanja novih činjenica i provjere znanja, baš kao na akademskim institucijama. No, prednosti sustava za elektroničko učenje nad tradicionalnim akademskim institucijama su veća prostorna dostupnost (praktički jedini uvjet sudjelovanja je internetska povezanost) te brže ažuriranje gradiva novim informacijama u odnosu na spori obrazovni sustav.

Sustavi za elektroničko učenje mogu biti dio nekog formalnog obrazovnog sustava ili samostalni. Moodle je primjer sustava za elektroničko učenje čiji je cilj obavljati samo dio obrazovanja, uglavnom provjere znanja i prenošenja informacija (materijala za učenje). Od korisnika se očekuje da sam proučava te materijale ili pohađa predavanja predmeta unesenog na sustav Moodle. Samostalni sustavi za elektroničko učenje trebali bi imati dostupne sve elemente učenja jer samo tako mogu osigurati kvalitetu. Eventualno može nedostajati formalna provjera znanja ako je sustav dizajniran tako da je postupak upijanja novih činjenica rigorozan u korištenju da ga korisnik ni u kojem slučaju ne može preskočiti ili ignorirati.

Neki od popularnih i besplatnih sustava za elektroničko učenje su Codeacademy, Doulingo i Coursera. Coursera je zanimljiv koncept učenja koji na internet stranicu prenosi predmete stvarnih sveučilišta. Ona je za kranjeg korisnika samostalna, no stvaranje njezinog sadržaja nije. Duolingo Najpoznatiji akamdemski sustav za učenje je Moodle, a na FER-u se također koriste Ahyco i Ferko.

2.1. Inteligentni sustavi za elektroničko učenje (ITS)

Inteligentni sustavi za elektroničko učenje posebna su vrsta sustava za elektroničko učenje koja korisniku pruža reakcije ili upute prilagođene isključivo njemu. Cilj svakog ITS je smanjiti ovisnost korisnika sustava o ljudskim učiteljima. Za razliku od ljudi, elektronički sustavi se ne umaraju, ne stare, gotovo uvijek imaju vremena za učenika. Naravno, inteligentni sustavi za elektroničko učenje također imaju svoje limite: kvalitetni su koliko je kvalitetan dizajner sustava, ne mogu pratiti fizičke reakcije korisnika i iskustveno zaključivati iz njih te, ipak, ne mogu biti potpuno individualni jer bi u tom slučaju količina praćenih podataka morala biti enormna.

2.1.1. Povijest ITS-a i zaključci iz nje

Mehanički strojevi za učenje

Ideja inteligentnih sustava za učenje nije nastala tek u 21. stoljeću, dapače, već u 17. st. dvojica od najvećih europskih matematičara, Blaise Pascal i Gottfried Wilhelm Leibniz razmatrali su koncepte zaključivanja uz pomoć strojeva. Pravi zamah elektroničko učenje dobiva u 20.

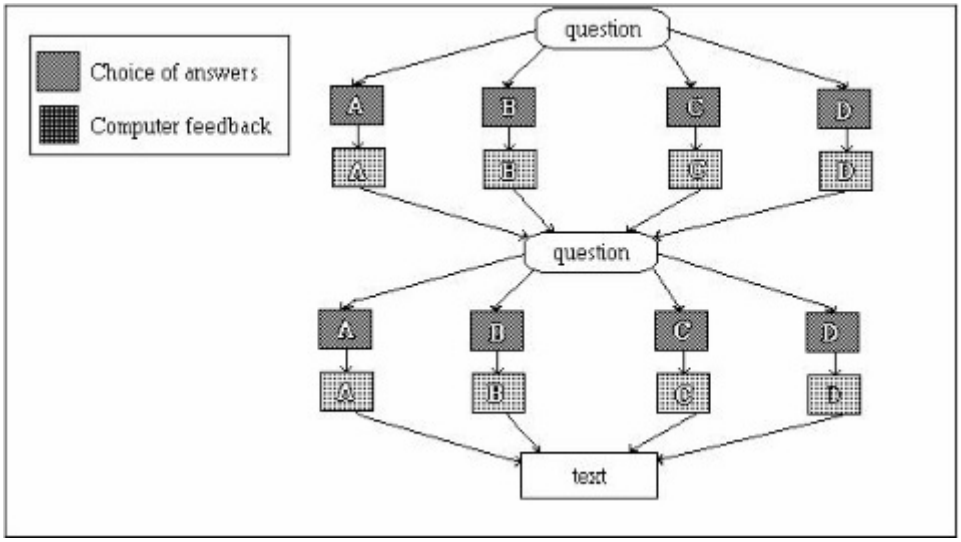
Prvi značajan korak prema učenju uz pomoć strojeva je napravio Sidney Leavitt Presley, profesor na Sveučilištu Ohio, koji je izumio stroj za samoocjenjivanje. Korisnik je odgovarao na pitanja s višestrukim izborom, stroj bi spremao odgovore te na kraju seta pitanja dao povratnu poruku o točnosti odgovora. Taj pristup koristi se i danas u većini akademskih sustava za ocjenivanje, sva tri, Moodle, Ferko i Ahyco ga koriste.

Slika 2.1: Stroj za ocjenjivanje S. L. Presleyja

Iako su mehanički strojevi pružili kvalitetan početak ideje, jasno je da zbog neefleksibilnosti i skupoće izrade nikada nisu mogli prijeći fazu demonstracije. No, računalne aplikacije je lako umnožavati i mijenjati. Zato su one pogodne za razvoj sustava za učenje, ovog puta elektroničkih.

Početak elektroničkih sustava za učenje

U ranim godinama razvoja računala i računarstva napravljeni su mnogi koncepti bliskih tematika, npr. Turingov test, ideja umjetne inteligencije i strojnog učenja, no sustavi za učenje kao koncept kakav danas postoji je relativno zanemaren. Jedini važan napredak su definiranja gramatika u računarstvu, kao jednog od budućih načina parsiranja korisnikovih odgovora. No, dolaskom računala u širu upotrebu, u američkim školama se stvaraju CAI (*Computer-Assisted instruction*) projekti s ciljem učenja programiranja. CAI sustav je takav da se korisniku prezentira neki materijal s uputom korištenja, nakon čega on na *input-output* principu komunicira s računalom. Najbolji primjer nečega sličnog CAI sustavu u današnje vrijeme je Codeacademy. U početku se CAI koristio u učionicama, a ne individualno, ali na sličnom principu korištenja.



Slika 2.2: Shema rada pomoću CAI sustava

Velik korak za stvaranje ITS-a napravio je Jaime Carbonell koji je 1970. postavio tezu da "sustavi za učenje ne moraju biti samo alat, nego i učitelj". U 1970.-ima je najpopularnije novo područje bila umjetna inteligencija temeljena na znanju. Najpoznatiji primjer takvog sustava je *Dendral*, sustav zaključivanja o kemijskim strukturama korištenim u organskoj kemiji. Izgradnja složenijih sustava je bila lakša zbog poboljšanja brzine računala što je fokus rada konačno maknulo s tehnologije na tehniku.

Početkom 1980. umjetna inteligencija se pomiče k neuronskim mrežama i kognitivnoj psihologiji što otvara prostor razvoju sustava za elektroničko učenje zbog sličnih područja interesa. To dovodi do stvaranja koncepta ICAI-ja (*Intelligent Computer Assisted Instruction*), vrste CAI sustava koji bi radio na principu prilagođenog posluživanja instrukcija. ICAI je jako blizu modernom konceptu ITS-a, jedina razlika između njih je što ICAI ne pokušava modelirati znanje korisnika.

ITS u doba interneta i analize podataka

Najvažniji korak za veću primjenu elektroničkog učenja je početak masovnog korištenja interneta. Internet je konačno omogućio neovisnost fizičkih pozicija korisnika sustava i kvalitete znanja koju mu sustav može pružiti. Slabost prijašnjih rješenja je bila u neadekvatnoj aktualizaciji sustava zbog činjenice da su se morali distribuirati na prijenosnim medijima. Nije slučajno da su svi sustavi koje sam na početku nabrojao internetske stranice.

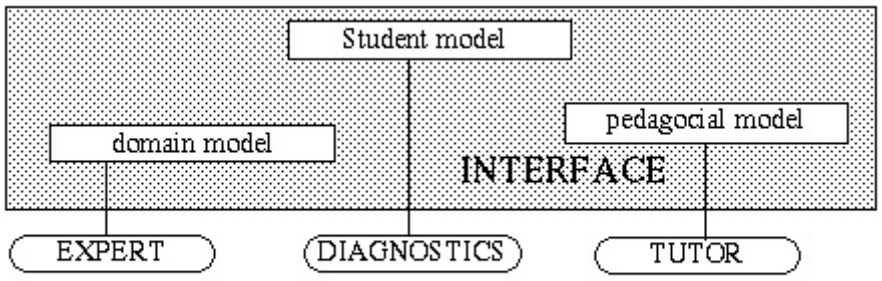
TODO nekakva slika internet connecting world

Analiza podataka, *data mining*, *big data* i *data science* je najveći korak u razvoju inteligencije ITS-a. Iako su principi analize podataka bili matematički postavljeni i prije 1990.-ih, rastom brzine računala, popularizacijom SQL-a i internet tražilica, analiza podataka doživjela je pravi *boom* zadnjih 20-tak godina. Razlog tolike važnosti analize podataka u inteligenciji ITS-a je što kompleksnost procesa učenja (čime se bavi kognitivna psihologija), "udaljnost" sustava i potreba za prilagođenim (personaliziranim) sadržajem korisniku. Ti faktori se jednostavno ne mogu opisati malim brojem varijabli. Svaki učitelj kroz godine upoznaje nove učenike te nadopunjuje svoje znanje i tehniku, baš kao što bi ITS bio spreman za nadogradnju temeljem iskustva korištenja.

Za razliku od UI čije rezultate velike tvrtke sigurno mogu naplatiti, razvoj sustava za inteligentno učenje je donedavno bio guran od strane akademske zajednice. Nove velike tehnološke tvrtke imaju potrebu educirati velik broj zaposlenika pa sustavi za učenje u današnje vrijeme konačno dobivaju zamah za napredak kao koncept, ali i u realizaciji.

2.1.2. Komponente ITS-a

Većina modernih ITS-ova dijeli se na 4 komponente rada: domenski i korisnički model, model učenja te korisničko sučelje. Tri prvonavedene komponente se nazivaju modeli zato što su pojednostavljena reprezentacija stvarnih pojava, koncepata i sl. One nemaju nužno veze s modelom kao aplikacijskim slojem, iako se uglavnom oslanjaju na njega.



Slika 2.3: Komponente ITS-a

Domenski model

Domenski model je dio sustava čiji je smisao organizirati znanje u jedinice spremne za pohranu u bazu podataka. Svaki sustav osmišljava vlastitu podjelu znanja s obzirom na različitu širinu podjele i broj razina. U pravilu je neizbježna podjela na predmete u složenim sustavima i koncepte unutar predmeta, a na autorima i administatorima sustava je definirati više ili niže razine u odnosu na koncepte. Granulacija znanja je zato prilično kompliciran posao većoj količini autora, no budući da je kreiranje domenskog modela kreativan posao, u procesu zamišljanja treba sudjelovati što veći broj ljudi.

Domenski model ne uključuje samo podjelu znanja, nego i pravila te strategije koje treba naučiti. Oni se mogu prikazati eksplicitno formulama i materijalima za učenje ili implicitno redoslijedom prikaza pitanja ili slično. Implicitno zadavanje pravila i strategija tipično je za formalne školske sustave zato što postoji određen redoslijed predavanja te nastavnici očekuju da učenici/studenti poznaju gradiva prethodnih predavanja. U sustavima za elektroničko učenje takav je pristup teže izvesti te se pravila i strategije najčešće zadaju eksplicitno u obliku najniže razine podjele u bazi podataka, iako korisnik uglavnom tu eksplicitnost ne vidi.

Prilikom kreiranja domenskog modela važno je misliti na to da je takav model kvalitetna reprezentacija različitih vrsta znanja, da bude efikasan i razumljiv za kreiranje

algoritama te da može odgovoriti na potrebe korisničkog modela.

Tipični domenski model je kvalitetno organizirana baza podataka te pravilno uneseni podaci. Za sve iznad toga se brinu ostale komponente, počevši sa korisničkim modelom.

Korisnički model

Učenički, studentski ili jednostavno korisnički model je komponenta sustava čija se svrha većim dijelom preklapa s domenskom komponentom i modelom učenja. Razlog odvajanja korisničkog modela je bolja analiza znanja pojedinog korisnika.

Korisnički model su uglavnom algoritmi koji analiziraju točnost odgovora (te možda daju dodatne savjete korisniku) i algoritmi koji omogućuju prikaz analize odgovora korak po korak. Zbog relativno male opsežnosti, neki izvori zanemaruju korisnički model cijepajući njegove dijelove u domenski i model učenja.

Model učenja

Nakon postavljanja domenskog i korisničkog modela, ostatak aplikacijske logike sadržan je u modelu učenja. Upravo je model učenja najbitniji dio inteligentnog sustava za elektroničko učenje jer on zamjenjuje ljudskog učitelja. Ovoj komponenti pripadaju algoritmi koji na temelju rezultata prethodno navedenih evaluiraju znanje korisnika, algoritmi korisnikove navigacije sustavom te algoritmi posluživanja pitanja.

Svrha modela učenja je praćenje korisnikova napretka u pojedinim konceptima/predmetima. Iako je na prvi pogled to relativno jednostavan problem, on jest takav za ljudskog učitelja, ali za računalo nije. Takvi problemi umjetne inteligencije (UI) nazivaju se UI-potpuni. Nakon što algoritmi procijene znanje korisnika, ono se treba prikazati nekim vizualnim načinom da korisnik bolje razumije što ne zna i zašto mu sustav postavlja neka pitanja.

Model učenja je srž ITS-a te glavni odraz kvalitete pojedinog sustava. Čest način implementacije modela učenja je logičkim produkcijama koje dovode do zaključka je li korisnik nešto naučio ili ne. Prednost takvog pristupa je formalna evaluacija korisnika i sustava samog, no nedostatak mu je nejasnost širem spektru administratora. Alternativni način je sustav bodovanja pojedinih pitanja, skupova pitanja i odnosa između koncepata koje korisnik treba savladati.

Korisničko sučelje

Korisničko sučelja je standardan element gotovo svake računalne aplikacije. Ono nužno ne mora biti grafičko, iako je takvo sigurno zanimljivije kranjem korisniku. Učenik treba vidjeti svoj trenutni napredak prikazan na jasan način, razumjeti na koji način sustav komunicira s njim i on sa sustavom te područje koje pitanja ispituju.

2.1.3. Karakteristike dobrih sustava

Nakon definiranja komponenata sustava za inteligentno učenje, potrebno je razmotriti koje karakteristike imaju kvalitetni sustavi za inteligentno učenje. Većina su obvezni dijelovi ITS-a te ih je baš zato važno navesti.

Razrađena podjela znanja

Iako se ovo čini kao očekivan uvjet iz opisa domenskog modela, važno ga je ponoviti i u ovom dijelu. Ispravna podjela znanja omogućuje lakše i kvalitetnije projektiranje algoritama procjene korisnika. Postoje dvije razine razrade podjele znanja: projektiranje sustava - odabir mogućih granula, povezanosti između njih i načina evaluacije znanja te projektiranje pitanja - punjenje sustava pitanjima i formulama za evaluaciju u skladu s dogovorenom podjelom.

Granulacija podjele znanja je pitanje bez jednoznačnog odgovora, a s puno rizika. Ukoliko sustav ima premalo razina podjele, postoji rizik od nedovoljno dobrog praćenja korisnikova znanja, a sustav s previše razina podjele zahtijeva ogroman broj različitih pitanja da svaka od razina ima reprezentativni uzorak ocjene korisnika.

Elegantno rješenje problema je određivanje srednje velikog broja granulacija (jasno, što je srednji broj ovisi o opsegu domene za koju se ITS projektira), a onda između pojedinih granula uspostaviti različiti broj odnosa. Na taj način sam i ja riješio problem u svojoj implementaciji.

Materijali za učenje

Svaki sustav čiji je cilj biti samostalan trebao bi sadržavati materijale za učenje poput e-knjiga ili uputa koju bi literaturu korisnik morao proučiti prije odgovaranja. Samo takav pristup sa sigurnošću korisniku omogućuje točnost znanja ili, ako se kasnije utvrdi pogreška u materijalima, referencu na pogrešku. Ukoliko sustav korisniku nudi samo provjeru znanja s objašnjenjima za pojedino pitanje, postoji velik rizik da korisnik neće dovoljno dobro razumjeti povezanost činjenica koje sustav ispituje.

Parametrizirana pitanja

Zamislite slučaj u kojem ITS služi za formalnu provjeru znanja na fakultetu čiji studenti imaju organizirane forume na kojima dijele zadatke. Nerealno je očekivati da može postojati više od 20 pitanja neke domene, a postoji preko 500 studenata u generaciji. Kako osigurati adekvatnost provjere? Odgovor je jednostavan: parametrizacijom pitanja.

Parametriziranje pitanja je postupak kojim se numerička (moguće i na tekstualnim, ali teže) pitanja u ITS zapisuju u obliku formula za evaluaciju. Korisnicima se prikazuju njihove vlastite vrijednosti pojedinih parametara, a točnost odgovora evaluira se formulom. Takav pristup ne samo da omogućuje jedinstvene vrijednosti različitim korisnicima, nego i različite vrijednosti istim korisnicima. Ukoliko neki korisnik želi ili mora više puta rješavati pitanja nekog koncepta, mogu mu se servirati jedinstvene vrijednosti parametara svaki put kada otvara to pitanje. Jedan od načina ostvarivanja parametrizacije pitanja je spremanje formule kao odgovora te popisivanja vrijednosti parametara i njihovih ograničenja (npr. imaginarni dio broja mora biti != 0, apsolutna vrijednost mora biti > 0 i sl.).

Praćenje tipkanja i reakcija lica korisnika

Jedna od najvažnijih karakteristika dobrog učitelja je uočavanje učenikovih reakcija, čitanje fizičkih znakova te zaključivanje iz toga. Najveći nedostatak sustava za učenje, kao i svake druge vrste udaljenog učenja, je fizička udaljenost učenika i učitelja. Stoga, elektronički sustavi za učenje moraju naći drukčije načine praćenja reakcija korisnika.

Dva najpoznatija načina su praćenje tipkanja i izraza lica. Rastom količine prostora na tvrdim diskovima i brzine računala, stvorila se mogućnost praćenja naizgled trivijalnih detalja kao što je obrazac tipkanja pojedine osobe. Neki izvori kažu da svaka osoba ima jedinstven obrazac tipkanja, baš kao otisak prsta. Zato bi bilo važno pratiti u kojim je situacijama korisnik ubrzan, o kojim pitanjima dulje razmišlja te je li mijenjao neki odgovor.

Praćenje izraza lica također je postalo moguće rastom brzine računala, njihove procesorske i grafičke snage te razvojem računalnog vida kao područja računarstva. Ideja takvog pristupa je aproksimirati ljudskog učitelja, no to je vrlo složen i težak posao. Nedostatak takvog pristupa je što svaki korisnik mora imati neki oblik web kamere.

Oba dijela su dio najnaprednijih sustava s velikim budžetima za izradu zato što ih je složeno implementirati.

Prilagođene reakcije

Neizostavna komponenta sustava za inteligentno učenje su reakcije (engl. feedback) temeljene na znanju korisnika. Jasno je da računalne aplikacije ipak ne mogu biti jedinstvene kao ljudski učitelji. Svejedno, reakcije sustava moraju biti kvalitetne i prilagođene pojedinom korisniku da bi ITS imao smisla. Postižu se analizom odgovora korisnika i njegova načina odgovaranja (trajanjem računanja, brojem izmjena odgovora prije potvrde i sl.) te dobrim projektiranjem sustava (postojanjem reakcije za svaku kombinaciju razine znanja).

Određivanje kada korisniku treba dati uputu (engl. hint) ako je zapeo na nekom zadatku je pitanje bez jednoznačnog odgovora, uostalom, kao i ljudskim učiteljima. S jedne strane učitelj mora znati detektirati kada je njegov učenik iscrpio sve mogućnosti, a s druge je jasno da otvoreno pitanje u sustavu za elektroničko učenje može značiti to da je korisnik jednostavno otišao od svojeg računala. Dva rješenja koja se nameću su pamćenje mogućih pogrešnih kombinacija parametara i broja pokušaja te dopuštanje korisniku da zatraži uputu ako sam shvaća da nema ideju. Nedostatak prvog rješenja je moguća prevelika nepotrebna složenost, a drugog to što umorni korisnici ponekad odustaju prerano.

Alternativni način rješavanja reakcija je jednostavno pustiti korisnika da odgovori krivo i dati mu uputu nakon što završi provjeru.

Sigurnost

Sigurnost je jako važan element svakog sustava, a to se ističe u dva aspekta: zaštita osobnih podataka i sprječavanje lažnog predstavljanja. Iako osobni podaci u ITS-u gotovo sigurno ne bi trebali sadržavati osjetljive informacije, važno ih je zaštiti nekim sustavom lozinki.

Slika 2.4: Ilustracija oznake sigurne stranice

Sprječavanje lažnog predstavljanja je puno zanimljiviji problem. Ukoliko sustav ima ugrađeno praćenje tipkanja korisnika ili izraza lica, prilično je očito da bi takav pristup minimizirao rizik lažnog predstavljanja. No, što ako nema? Alternativna opcija je formalno rješavanje ispita ITS-a uz prisustvo administratora što vodi do gotovo

potpunog eliminiranja samostalnog učenja. Tako da niti to nije najbolje rješenje. Koje onda jest? Zapravo ga i nema, sustav nažalost mora vjerovati korisniku ako ne želi implementirati skupe tehnologije praćenja tipkanja ili računalnog vida.

Kvalitetni algoritmi

Kvalitetni algoritmi su nešto što se gotovo podrazumijeva ukoliko želimo kvalitetan sustav. Sljedeća lekcija govori više o algoritmima sustava za elektroničko učenje, no bilo ih je važno navesti i u ovoj.

Dizajn sučelja

Posljednja, ali ne i najmanje važna karakteristika ITS-a je dizajn grafičkog korisničkog sučelja. To nije važno samo za ITS, nego i za svaku drugu računalnu aplikaciju jer loše sučelje odbija korisnike više nego bilo što drugo. Prilično je teško definirati što je točno kvalitetno grafičko korisničko sučelje, no ono bi sigurno moralo imati sljedeće karakteristike: intuitivnost rasporeda kontrola, lakoća upravljanja (brzina, dosljednost, dozvole promjena kontrola), ugodan izgled (boje, fontovi, izbjegavanje neugodnih efekata promjena boja") te prikladan prikaz informacija.

Korisničko sučelje ITS-a također bi se trebao držati nekih principa jedinstvenih za to područje. Tri najvažnija su: mogućnost traženja upute (također su opcije gumb za odustajanje ili ispis odgovora), konzistentnost znakovlja (npr. cos ili csin za kosinus, točka ili zarez kao decimalna oznaka) te prikaz napretka u pojedinoj granulaciji znanja.

2.1.4. Algotimi sustava za inteligentno učenje

Najprije smo saznali koji su dijelovi inteligentnog sustava za elektroničko učenje te koje principe moramo slijediti prilikom implementacije jednog takvog sustava, no sljedeće je pitanje što sve treba implementirati da bi sustav bio inteligentan. Najbolji odgovor je odrediti koje sve algoritme ITS treba imati, tj. koje bi poslove njegovi najvažniji algoritmi trebali odrađivati.

Odabir pitanja

Vjerojatno najznačajniji algoritam sustava za inteligentno učenje je algoritam odabira pitanja. On može biti oblikovan na različite načine, no dva su najčešća: logičkim produkcijama koje nadopunjuju produkcije procjena korisnikova znanja ili samostalan programski odabir. Prednost logičkih produkcija je konzistentnost tehnologija koje

sustav koristi, no značajni nedostaci su im ovisnost o apsolutnoj točnosti produkcija evaluacije znanja te potreba da autor pitanja sastavlja algoritam. Zato se često koristi druga metoda.

Programski odabir pitanja definiran je na razini sustava, a radi na principu procjene korisnikova znanja pojedine granulacije i odnosa između njih te odabira podskupa pitanja iz skupa onih koja je moguće postaviti

Pitanja se odabiru tako da se uračuna korisnikovo znanje, zahtijevnost i složenost pojedinog pitanja, što bi korisnik naučio ili dokazao točnim odgovorom na to pitanje te ukupna složenost provjere koja sadrži ta pitanja.

Generiranje parametara

Nakon što je sustav odabrao pitanja koja će postaviti, mora odrediti konkretne parametre svakog od njih. Algoritam generiranja parametara ih može generirati iz neke mape mogućih slučajeva ili nasumičnom generacijom uz određene uvjete. Uvjeti generiranja parametara važan su element tog postupka kako se ne bi dogodilo da se nekom korisniku postavi pitanje u čijem je rješenju npr. greška dijeljenja s 0. Ukoliko se unutar sustava dobro definiraju ograničenja, generiranje parametara postaje posao poziva funkcija slučajnih brojeva dok se ne zadovolje ograničenja.

Evaluacija odgovora

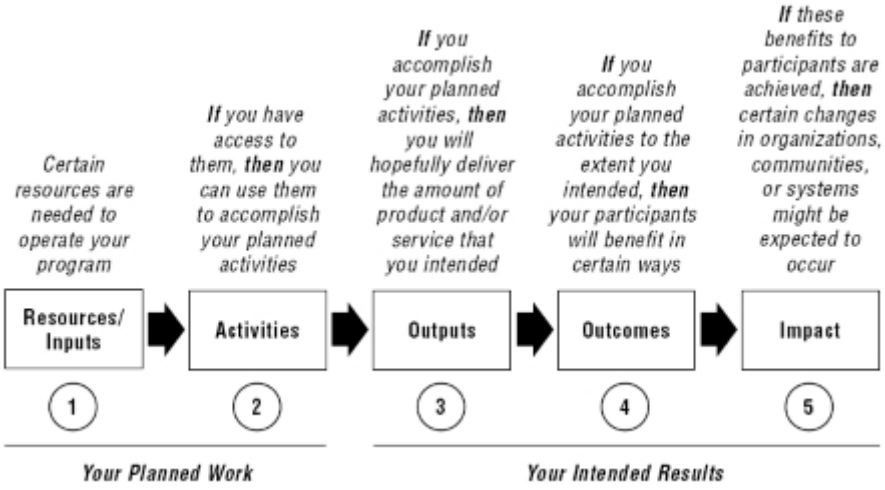
U dobro projektiranom sustavu evaluacija odgovora svodi se na procjenu točnosti formule zapisane u bazi podataka za određene parametre. U formuli se parametri promijene u stvarne brojeve te ona prolazi provjeru jednakosti +/- n% s korisnikovim rješenjem.

Procjena znanja

Algoritmi procjene znanja mogu se temeljiti na različitim pogledima na korisnikov odgovor. Neki od najzanimljivijih faktora je li ili nije korisnik nešto naučio su točnost pitanja koja pripadaju nekoj granulaciji, odnosi između znanja, složenost zadatka u odnosu na ostale zadatke koje je korisnik riješio, koliko je vremena prošlo od zadnjeg ispitivanja pojedine granulacije, težina tog pitanja drugim korisnicima koji su došli do njega i koliko je dugo korisnik rješavao to pitanje.

U praksi je algoritam procjene znanja najkreativniji dio sustava za inteligentno učenje jer su mu jedina ograničenja da se mora izvesti u nekom razumnom vremenu i da, naravno, sustav pamti tražene vrijednosti.

Algoritam procjene znanja također može biti zapisan logički ili programski, u njegovom slučaju je logički češća pojava. Takav algoritam na temelju zapisanih produkcija donosi odluke koliko neke granulacije je korisnik naučio. Za svaki njezin dio se izbacuje vrijednost 1 (naučeno) ili 0 (nenaučeno). Programska alternativa može raditi isti taj posao drukčijom tehnologijom, oslanjati se na praćenje korisnika za *data mining* ili donositi odluke na nekom trećem principu. Prednost programskog nad logičkim izračunavanjem je što programski u pravilu može elegantije doći do neke ocjene i što ne mora ovisiti o konkretnim granulacijama.



Slika 2.5: Općenita shema logičke evaluacije

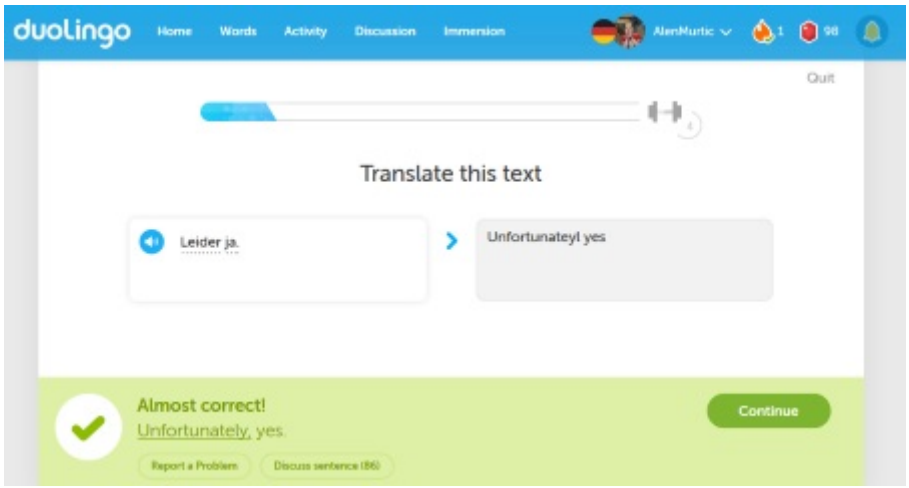
2.1.5. Poznati sustavi

Znamo što sve treba napraviti prilikom dizajniranja sustava za inteligentno učenje, no nismo pogledali što su pametni ljudi prije nas radili u tom pogledu. A radili su puno, stoga bi bilo šteta propustiti izvući znanje i iskustva njihovih sustava u cilju dizajniranja jednog kvalitetnog ITS-a. Čak i neinteligentni sustavi mogu ponuditi korisne zaključke, možda čak i jasnije nego inteligentni. Zašto? Zato što su konceptualno relativno jednostavni te kod njih lakše možemo uočiti neke probleme.

Bitno je naglasiti da nisu svi sustavi u ovom dijelu samo pitalice, već on sadrži neke slobodnije poglede na učenje jer različiti ljudi uče na različite načine. Učiti se može vizualnom tehnikom, slušanjem, igrom, ponavljanjem / čitanjem naglas, rješavanjem zadataka, gradnjom rješenja itd. Zato bi bilo kontraproduktivno da svaki sustav za učenje ima oblik pitalice.

Neinteligentni sustavi

Duolingo je sustav za učenje jezika u čijoj izradi mogu sudjelovati i korisnici. Princip rada Duolinga je da korisnik odabere svoj materinji (ili dobro poznati jezik) te dobije ponudu jezika za koje postoji "tečaj" uz pomoć njegova materinjeg jezika. Pitanja mogu biti dopunjavanje, prevođenje ili pitanje s odabirom od ponuđenih odgovora. Sustav je neinteligentan osim u dva pogleda: znanje nekog područja zastarjeva ukoliko korisnik nije dugo odgovarao pitanja iz tog kategorije te sustav odabire područje s najmanjom vrijednosti znanja ukoliko korisnik odabrere opciju "pojačavanja znanja" (engl. strengthen skills).



Slika 2.6: Inteligentni aspekt sustava Duolingo: prepoznavanje "tipfelera"

Coursera je internetska stranica koja sadržava kolegije najpoznatijih svjetskih sveučilišta s ciljem smanjivanja ovisnosti fizičke pozicije ljudi i njihovih mogućnosti učenja. Sadrži besplatne i plaćene tečajeve čije materijale korisnik može (i mora) proučavati, odgovarati na pitanja te na kraju dobije potvrdu svojeg znanja. Jasno je da položiti neki tečaj na Courseri nije jednakovrijedno kao završiti kolegij na fakultetu, no ukoliko korisnik ima stvarnu volju za učenjem, količina stečenog znanja može biti jednaka. Iako su materijali velikog broja fakulteta (pogotovo tehničkih) javno dostupni, Coursera eliminira potrebu za iscrpnom pretragom materijala i stavlja fokus na učenje.

Urban Jungle je simulacija vožnje koju su izradili Autoklub i Udruga darovitih informatičara Rijeka da bi mladima olakšali polaganje vozačkog ispita. Korisnik može voziti virtualni automobil kao i u sličnim računalnim igrama, no Urban Jungle ima edukativnu svrhu jer za uspješno "igranje" korisnik mora paziti na stvarna prometna pravila. Odličan je primjer da računalni sustav namijenjen za učenje ne mora nužno

biti samo niz pitanja i odgovora.

Slika 2.7: Slika ekrana prilikom vožnje u Urban Jungle edukativnoj igri

Moodle je vrlo popularan akademski sustav za učenje koji omogućuje neke koncepte ITS-a (parametrizirana pitanja, interaktivno sučelje), no nije ITS jer ne sadrži inteligentno odlučivanje. Sustav Moodle gotovo uvijek funkcionira kao podrška formalnom sustavu učenja zato što omogućuje planere, kalendare, obavijesti korisnicima i sl. Moodle može biti osnova za izgradnju drugog sustava za učenje, kao što je slučaj s CARNet-ovim sustavom Merlin. Moodle je aplikacija otvorenog koda te ju je zato vrlo lako moguće prilagođavati. Sustav Ferko razvijen na Fakultetu elektrotehnike i računarstva ima sličnu domenu primjene kao Moodle.

Slika 2.8: Logo sustava Moodle

Inteligentni sustavi

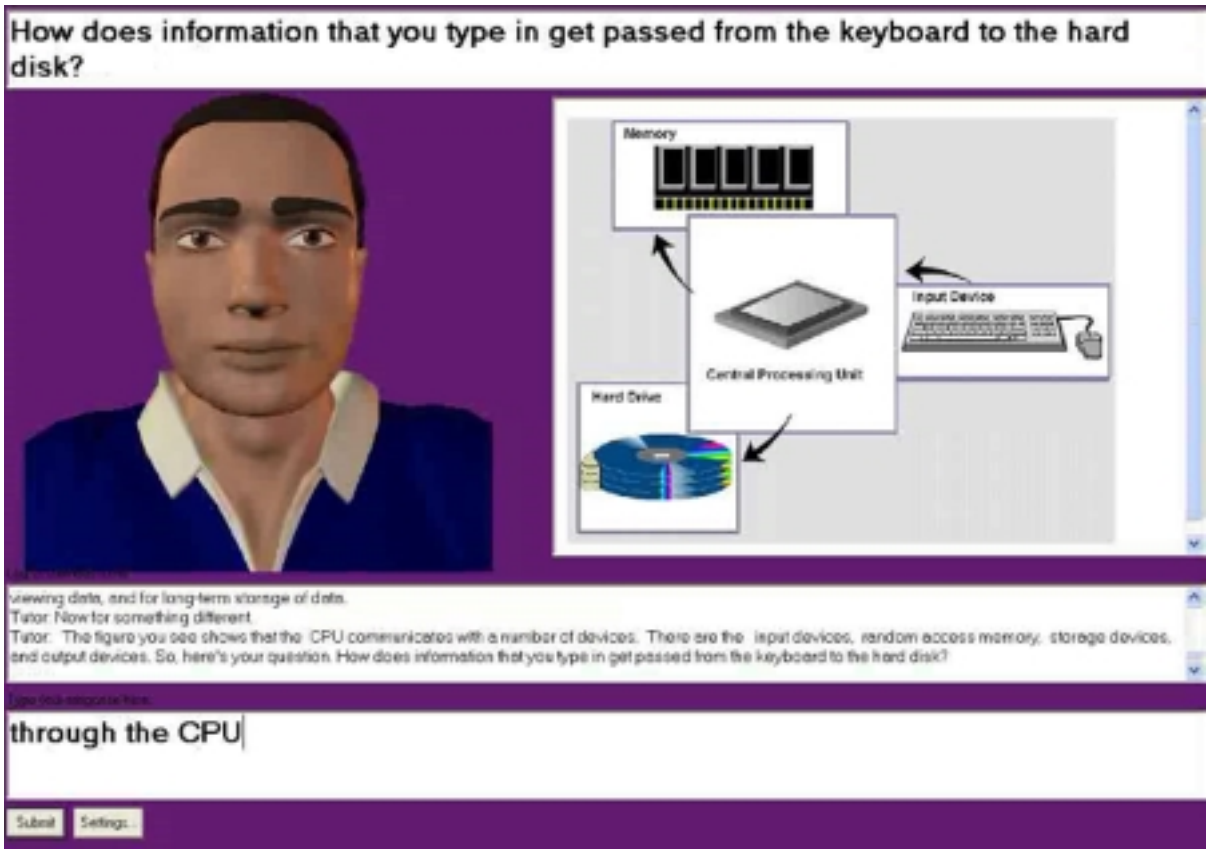
Postoje mnogi sustavi za inteligentno učenje koji rade na principu pitalica te sam zato odabrao najzanimljivije primjere ITS-a koji možda i ne slijede dosad navedeni

Slika 2.9: Logo sustava Merlin temeljenog na Moodleu

obrazac, no svejedno pripadaju skupini sustava za inteligentno učenje.

Why2-Atlas je ITS koji uči korisnika koncepte kvalitativne fizike tako da traži od njega da napiše nekoliko rečenica od određenom pojmu te abduktivnom sintaktičkom analizom pokušava dokučiti znanje korisnika. Takav proces može se odvijati u nekoliko iteracija dok sustav ne odredi znanje korisnika te ga uputi na točno rješenje ukoliko je u krivu.

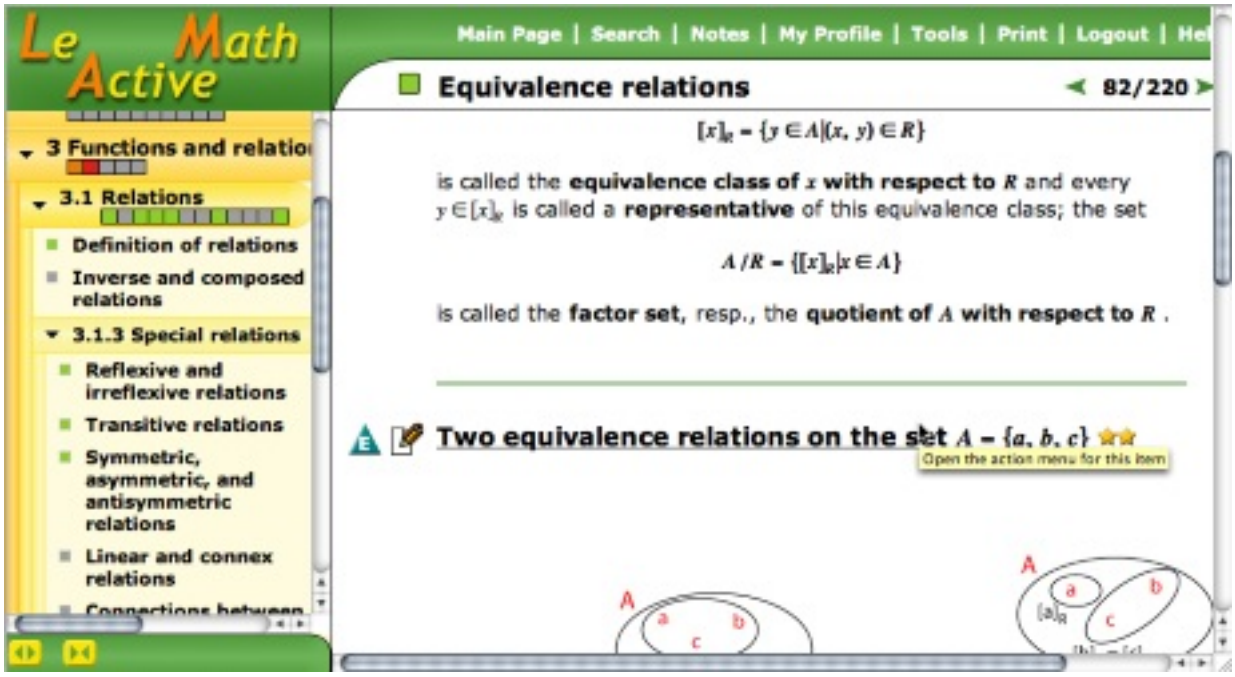
AutoTutor je sustav za inteligentno učenje razvijen na Arizona State sveučilištu koji razgovara s korisnikom prirodnim jezikom. Princip rada AutoTutora je parsiranje govora na sličan način na koji se prevode programski jezici.



Slika 2.10: Sučelje razgovora sustava AutoTutor

ActiveMath je ITS vrlo sličan prototipskom pogledu na sustave za inteligentno učenje. To je sustav za podučavanje dodatno matematike u školama ili učenje na daljinu.

ActiveMath je internetska stranica na kojoj pored sustava za učenje postoje i materijali koji pokrivaju gradivo. Kvaliteta ActiveMath ITS-a je u tome što ga autori prilagođavaju s obzirom na empirijske rezultate kako su korisnici učili.



Slika 2.11: Slika ekrana ActiveMath sustava

Igre s prilagodljivom umjetnom inteligencijom (adaptive AI) nisu namijenjene za učenje, no svejedno imaju mnogo dodirnih točaka sa sustavima za inteligentno učenje. Ideja takvih igara je uočiti najčešće igračeve obrasce ponašanja te nove protivnike stvoriti takvima da igrač mora odabrati drukčiju strategiju. Mnogo računalnih igara tvrde da imaju prilagodljivu UI, no zasad taj koncept nije stvarno baš dobro implementiran. Nogometne simulacije i strateške igre u stvarnom vremenu (engl. RTS) najviše reklamiraju takvu UI, no iskustva korisnika kažu da je UI u njima lošija nego prije desetak godina.

2.1.6. Nedostaci ITS-a

Podučavanje i inteligencija

Metode uspješnog podučavanja su očito najveći izazov svakog elektroničkog sustava za učenje. Iako neki izvori navode da kod problema koji se mogu rastaviti na sitne korake elektronički sustavi i ljudski učitelji imaju sličan rezultat usvojenosti gradiva kod učenika, podučavanje nije samo prenošenje znanja. Ono uključuje i metode usvajanja gradiva, procjene korisnikova znanja te oblikovanja odgovarajućih procesa učenja.

Ponajveći problem metoda podučavanja i inteligencije ITS-a je što (zasad) nemaju pitanja "zašto?", "kako?" na korisnikov odgovor te kao takvi ne mogu dovoljno dobro procijeniti je li korisnik samo naučio "kuharicu", tj. formule za rješavanje zadataka. Jasno da sustav može pitati ta pitanja, no ona su svejedno organizirana po nekom predviđenom obrascu. Možda je nekada uputno pitati korisnika neka pitanja koja navodno zna kako bi bili sigurni da nešto stvarno razumije. Tu do izražaja dolazi intuicija učitelja koju sustavi (zasad) nemaju. Intuiciju donosi umjetna inteligencija temeljena na potpornom učenju, a ona je prilično različita od produkcijske umjetne inteligencije koju ITS koristi za definiranje je li korisnik nešto naučio.

Drugi problem podučavanja je koliko korisniku sustav treba pomagati u rješavanju zadataka. Teza takve kritike je da korisnik jednostavno ne nauči dovoljno ukoliko mu sustav ponudi rješenje, da je takvo učenje previše površno. Osobno se slažem s tim pogledom i mislim da bi korisniku bilo korisnije prikazati bitnije upute tek nakon završene provjere, a onda ga manje kažnjavati (u dugoročnom pogledu) za krive odgovore.

Naravno da je i problem ITS-a što autori pitanja jednostavno moraju odlično razumijevati tehničku pozadinu sustava (smisao pojedine granulacije, veze između granula) te ako ne razumiju ili sustav nije dovoljno dobro projektiran, u sustavu bi moglo doći do problema u definiranju korisnikova znanja ili otvaranju pitanja. Rizik površnog učenja je u takvim situacijama značajno veći.

Cijena izrade

Nadam se da je čitatelju jasno iz prethodnog dijela završnog rada da je izrada kvalitetnog ITS-a vrlo složen posao. Čak ne niti toliko težak, koliko opsežan. Problem izrade ITS-a je nužna bliska suradnja između računalnih stručnjaka i učitelja domene primjene. Velik broj ljudi koji moraju raditi istovremeno znači da je potreban velik broj

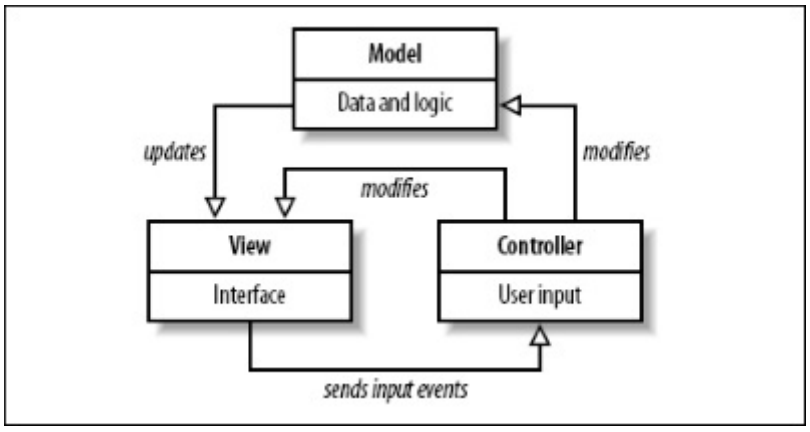
radnih sati, što znači da ih treba platiti. Ukoliko vlasnik sustava ne očekuje da će mu ITS značajno povećati broj korisnika, logično je postaviti pitanje opravdanosti investicije. Pogotovo imajući u vidu cijenu održavanja sustava i rizik pogreške u sustavu. Zato je dosad ITS-ove uglavnom razvijala akademska zajednica.

Slika 2.12: Ilustracija timskog rada potrebnog za dobar ITS

Drugi problem kod cijene izrade je kompleksnost sustava. Sustav za učenje jednostavno mora biti efikasan i brz, a sustav koji prati sve faktore to jednostavno ne može biti. Inženjeri koji ga projektiraju moraju napraviti *trade-off* između brzine i kvalitete sustava.

3. Slojevi aplikacija

Sve kvalitetno oblikovane moderne aplikacije moraju sadržavati nekoliko slojeva rada. Slojevita arhitektura omogućuje sigurno zadržavanje dva najvažnija logička načela dobrog oblikovanja: nadogradnju bez promjene i načelo jedinstvene ovisnosti. Takav pristup također intervencije u kod aplikacije čini lakšim. Arhitektura koju koristim je MVC (Model - View - Controller). Prevedeno na hrvatski, model je podatkovni sloj, controller je aplikacijska logika, a view pogled. Najveća prednost ovakvog pristupa je jednostavna i intuitivna višeplatformska modularnost - podatkovni sloj je zajednički za sve platforme, aplikacijsku logiku može koristiti više različitih platformi (npr. kod pisan u Javi za Android i web aplikacije ili C# za Windows 10 i web), a svaka platforma određuje svoje korisničko sučelje, tj. pogled.



Slika 3.1: Komunikacija u MVC modelu

Podatkovni sloj je sloj pohrane, dohvata i ažuriranja podataka, gotovo uvijek izveden u obliku baze podataka te ponekad pripadnih tehnologija kao npr. Entity Framework koji mapira objekte i bazu podataka. Podaci spremljeni u podatkovnom sloju moraju biti u *display-neutral* obliku, što znači da bez daljnje obrade nisu previše pogodni za sami prikaz. Cilj takvog pristupa spremanja podataka je imati bazu u prve tri normalne forme te posljedično očuvati performanse baze na visokoj razini čak i uz veću količinu podataka spremljenih u nju. Podatkovni sloj implementiran u ovom

završnom radu je SQL baza podataka te Entity Framework.

Aplikacijska logika je sloj upravljanja korisničkim zahtjevima, a radi kao poveznica između prikaza i podatkovnog sloja. Aplikacijska logika prima naredbe koje je korisnik zadao na pogledu, zatim provodi neke operacije nad modelom, uzima podatke iz modela te ih u *display-neutral* obliku šalje pogledu. Ona je uglavnom implementirana kao nekoliko slojeva različitih poslova, npr. *utlity* sloj rada s bazom, sloj koji prima podatke od pogleda te algoritamski sloj koji provodi operacije. U ovom završnom radu aplikacijska logika će upravo raditi na taj način: statičke klase rada s bazom, klase algoritama odabira pitanja te klase koje primaju pozive operacija s korisničkog sučelja.

Pogled, tj. korisničko sučelje je način na koji se aplikacija prezentira kranjem korisniku te način na koji je on koristi. Njegov je cilj rada obraditi podatke tako da ih može lijepo i kvalitetno prikazati korisniku te slati zahtjeve korisnika aplikacijskoj logici. Budući da zadatak ovog završnog rada nije implementirati taj sloj aplikacije, korisničko sučelje bit će opisano samo na način komunikacije između sustava i korisnika.

Uspoređujući slojeve inteligentnog sustava za elektroničko učenje i MVC višeslojne aplikacije, vidimo jasnu povezanost izemđu domenskog modela i modela kao dio MVC-a, controllera te korisničkog modela i modela znanja te grafičkog sučelja i pogleda. Upravo to je smisao MVC pristupa, jednostavno je logičan za ogromnu količinu situacija. Zašto onda ovaj završni rad navodi oba? Zato što je MVC tehnološki presjek, a slojevi ITS-a logički. Njihovo poklapanje je potvrda da je aplikacija projektirana na ispravan način. Filozofskiji odgovor bi bio da je MVC pogled programskog inženjerstva na aplikaciju, a slojevi ITS-a računarske znanosti.

4. Opis implementiranog rješenja

Zadatak ovog završnog rada osim dokumenta je implementirati konkretan sustav za inteligentno učenje koji sadrži većinu opisanih elemenata ITS-a. Njega čine baza znanja predstavljenog različitim granulacijama, parametrizirana pitanja, reprezentacija usvojenosti pojedinog korisnika, aplikacijska logika posluživanja pitanja te logika slaganja ispita.

4.1. Korištene tehnologije

Budući da je zadatak ovog završnog rada sadržavao izradu baze podataka i aplikacijske logike za sustav inteligentnog učenja, u izradi programskog rješenja potrebno je bilo koristiti nekoliko različitih tehnologija. One su:

Deklarativni programski jezik **SQL** i programsko okruženje **Microsoft SQL Management Studio** za izradu i kontrolu relacijske baze podataka te popunjavanje iste konkretnim podacima.

Programski jezik **C#** i razvojno okruženje **Microsoft Visual Studio** za izradu aplikacijske logike.

Skup tehnologija **Entity Framework** za dvosmjerno preslikavanje entiteta iz baze podataka i objekata iz aplikacije.

Biblioteka **mXparser** za izračunavanje parametriziranih matematičkih izraza.

TODO: **dijagrami**

4.2. ER dijagram

Dijagram entiteta i relacija temeljni je opis svake relacijske baze podataka. Njegova važnost nije samo bolje razumijevanje značenja baze i njezinih entiteta, nego i činjenica da dobro organizirani ER dijagram za jednostavnije baze gotovo uvijek znači da su one u trećoj normalnoj formi.

4.2.1. Slika ER dijagrama i entiteta u bazi

slika ER dijagrama

slika entiteta s atributima

4.3. Mogućnosti sustava

Mogućnosti sustava podijeljene su na one administratora, korisnika koji uči i zajedničke mogućnosti. Za obje vrste korisnika važne su granulacije znanja i način parametrizacije zadataka, za administratora je važno kako će dodavati pitanja u sustav, generirati ispite korisnicima te pregledavati njihove rezultate, a za korisnika koji uči registracija na sustav, način inteligentnog učenja te pregled vlastitih rezultata.

4.3.1. Reprezentacija znanja

Drugo poglavlje ovog završnog rada između ostaloga iznosi tezu o važnosti kvalitetne granulacije znanja u sustavu za inteligentno učenje. Stoga je njoj bio posvećen velik dio planiranja implementacije sustava.

Granulacije znanja

Znanje se u ovom ITS-u dijeli na 3 kategorije: predmet, koncept i granula znanja.

Predmet je najveća kategorija te reprezentira sličnu količinu znanja kao jedan fakultetski kolegij, iako to nije nužno ukoliko se sustav ne koristi u svrhu formalne provjere. Entitet predmet je logičan početak zbog činjenice da sustav može biti korišten u svrhu različitih domena primjena s računskom osnovom te zbog toga da je jednostavno prirodan kao početna točka razmišljanja. Svaki predmet može imati svoje administratore koji mogu pregledavati rezultate, generirati testove i sl.

Entitet predmet sadrži pripadne koncepte, a oni mogu pripadati samo jednom predmetu (1..N). Koncept je veća cjelina unutar predmeta koja se u dobro strukturiranim fakultetskim kolegijima može definirati kao jedan tjedan predavanja. Jedan predmet bi trebao sadržavati najmanje tri, a najviše 10-tak koncepata. Ne postoji konkretna preporučena brojka zato što svaki od predmeta može biti drukčije logički strukturiran.

Svaki od koncepata sadrži granule znanja, najmanje granulacije ovog sustava. Granule znanja namijenjene su prikazivati neke specifične postupke, manje dijelove većih pojmova ili bilo koje druge nedjeljive cjeline čije se značenje pojavljuje u pojedinim zadacima. Također su vezane 1..N vezom, eksplicitno s konceptom, a time implicitno i s predmetom.

Primjer koji neće biti odrađen u ovom završnom radu, a mogao bi pomoći razumijevanju predviđene strukture podjele:

- Predmet: Umjetna inteligencija
  - Koncept: Pretraživanje prostora stanja
    - Granula znanja: Pretraživanje u dubinu
    - Granula znanja: Pretraživanje u širinu
    - Granula znanja: A-zvjezdica pretraživanje
  - Koncept: Igranje igara
    - Granula znanja: Minimax algoritam
    - Granula znanja: Minimax s podrezivanjem
- Predmet: Osnove elektrotehnike
  - Koncept: Magnetizam
    - Granula znanja: Coulonova jednadžba
    - Granula znanja: Naboji u prostoru
    - Granula znanja: Dielektrici
  - Koncept: Istosmjerna struja
    - Granula znanja: Strujni i naponski izvori
    - Granula znanja: Kirchhoffovi zakoni
    - Granula znanja: Načelo superpozicije

Odnosi između granulacija

Prilično je očito da samo tri granulacije znanja nisu dovoljne za opis svih mogućih situacija u znanju. Ukoliko dva različita predmeta imaju povezane koncepte (npr. Fouriereova transformacija u MAT3 i SIS koji su dio 2. godine FER2 programa) želimo omogućiti da administratori predmeta mogu zabilježiti prethodno korisnikovo znanje. Ili unutar jednog predmeta, ponekad nema smisla korisnika dvaput ispitivati analogne stvari (npr. osnove Theveninovog teorema za izmjeničnu i istosmjernu struju). Ili jednostavno želimo korisniku onemogućiti da otvara neke nove koncepte prije dokazivanja prihvatljive razine znanja u prethodnima (npr. korisnika koji ne poznaje kombinatoriku nema smisla ispitivati pitanja iz diskretne vjerojatnosti). Stoga ovaj sustav donosi dodatnu mogućnost modeliranja znanja: odnose između granulacija.

Odnosi mogu biti definirani samo između dvije granulacije jendake vrste i to samo za koncepte i granule znanja. Definicija na razini predmeta nije potrebna zato što je predmet "sačinjen" od koncepata, a nema ograničenja u pogledu odnosa koncepata dvaju različitih predmeta. U sustavu su definirani sljedeći odnosi:

- **Preduvjet** Ovaj odnos definira nemogućnost otvaranja sljedeće granulacije prije postizanja prihvatljive razine znanja u baznoj. Privatljiva razina znanja ekvivalent je ocjene dovoljan na pojedinoj granulaciji.
- **Podskup (nadskup)** Odnos između dvije granulacije u kojom je jedna u potpunosti sačinjena od druge. Podrazumijeva se da je niža granulacija pravi podskup jer semantički nema smisla imati dvije identične granulacije, iako to sustav nigdje eksplicitno ne provjerava.
- **Korištenje** Ukoliko je znanje definirano nekom granulacijom uvijek nužno koristiti u drugoj, nema potrebe za stalnim eksplicitnim povezivanjem zadatka i te granulacije nego administrator može zadati odnos korištenja.
- **Analogno** U situacijama u kojima postoji jasna povezanost između dvaju pojmova, takva da znanjem jednog postoji razumijevanje drugog, može se definirati ovaj odnos. U takvom slučaju korisniku se dodaju male količine znanja analognog pojma te postoji mogućnost za manjim brojem elementarnih pitanja u početku ispitivanja novog pojma.
- **Paralelno** Postoji mogućnost da dvije granulacije opisuju istu situaciju na različite načine (npr. pristupi umjetnoj inteligenciji - logički, kognitivni, itd.) te zapravo korisnik uči više različitih pogleda. Za logiku sustava i model učenja paralelnost granulacija znanja nema nikakvog utjecaja te je ona tu samo da korisnik bolje razumije što i kako uči.

Niti jedan od odnosa ne isključuje korištenje bilo koje drugog, no preporuka autoru pitanja je da ne pretjeruje s kombiniranjem više od dva-tri odnosa po jedinki granulacije da sustav ne bi zapetljao tako da eliminira inteligentno posluživanje pitanja.

4.3.2. Parametrizirani zadaci

Jedan od zadataka ovog završnog rada je parametrizacija pitanja koja se poslužuju korisniku. Ona je izvedena na razinama baze podataka i aplikacijske logike. Izrazi su zapisani u bazi podataka, a konkretne parametre generira aplikacijska logika.

Zapis u bazi podataka i generiranje

Svako pitanje u bazi predstavljeno je entitetom zadatak koji sadrži tekstualno pitanje, matematički izraz koji treba izračunati, eventualno sliku te parametre. Parametri trebaju biti zapisani u standardnom tekstualnom obliku: "parametar:uvjet parametar:uvjet", itd. Član parametar predstavlja samo ime tog parametra, a uvjet činjenicu

da mora biti veći ili manji od nekog broja. Standardan zapis uvjeta je: >broj1&<broj2. Ukoliko parametar nema uvjeta, zapisuje se "none". Vrijednosti parametara koje se zadaju pojedinom korisniku generira sustav i one mogu biti isključivo decimalne.

Parametre uz poštovanje uvjeta generira aplikacijska logika. Parametri su decimalni brojevi predodređenog raspona između -1000 i 1000 uz mogućnost manjeg skupa temeljem uvjeta.

Izračun rezultata

Nakon što je korisniku predstavljena vrijednost parametara i on odgovori na pitanje, na temelju tih vrijednosti sustav izračuna predviđeni odgovor te s tolerancijom od 5 posto uspoređuje njegov i predviđeni odgovor. Izračun se izvodi uz pomoć mXparser biblioteke.

4.3.3. Mogućnosti administratora

U sustavu postoji veza između predmeta i administratora koja znači da samo na određenom predmetu sljedeće operacije mogu raditi samo oni administratori koji imaju ovlasti. To je logično kod svakog većeg sustava, no ipak je važno i eksplicitno napomenuti.

Dodavanje pitanja

TODO

Registracija korisnika na predmet

Prilikom registracije korisnika u sustav, on nije povezan s predmetima, već ga administrator mora dodati na predmet kojim upravlja. On ne može izbrisati korisnika sa predmeta, tj. učiniti kao da nikada nije postojao, već mu samo dati negativnu zaključnu ocjenu.

Generiranje ispita

Administrator na svojem predmetu ima mogućnost generiranja ispita za zadane korisnike. Ispitu se zadaje broj bodova, broj zadataka i faktor složenosti između 1 i 5. Faktor složenosti određuje prosječnu složenost zadataka na ispitu. Moguće je zadati i maksimalnu/minimalnu složenost pojedinog zadatka na ispitu s ciljem Ukoliko sustav nema dovoljno pitanja s takvim parametrima o tome obavještava administratora

te čeka ili dozvolu generiranja ispita manjeg broja zadataka ili unos novih parametara za generiranje. Administrator nema mogućnost određivanja bodovanja pojedinog zadatka zbog inteligencije sustava koja sama određuje najbolje bodovanje. Na to administrator može utjecati samo prilikom zapisivanja pitanja u bazu biranjem njegove složenosti. Ispit mora biti generiran za određene korisnike sudionike na pojedinom predmetu. Ukoliko sustav shvati da korisnicima ne bi smjela biti ponuđena pitanja s parametrima koje je odredio administrator (tj. neki od korisnika nisu zadovoljili preduvjete), sustav nudi mogućnost poništenja generacije ispita ili nastavak. Pretpostavka je da je administrator možda pogriješio ili da korisnik nije bio dovoljno odgovoran u rješavanju zadataka.

Pregled rezultata

Administrator može vidjeti rezultate korisnika na svome predmetu, a oni uključuju: pregled korisnikova znanja na pojedinoj provjeri, korisnikovo znanje na pojedinom konceptu tog predmeta te preporučenu konačnu ocjenu za navedeni predmet. Ukoliko administrator odluči potvrditi tu ocjenu, korisniku se zaključava navedeni predmet i u sustav zapisuje konačna ocjena.

4.3.4. Mogućnosti korisnika

Sljedeće mogućnosti korisnici mogu odgovarati samo na pitanja iz predmeta kojima pripadaju, baš kao i administratori. Za registraciju na pojedini predmet korisnici koji uče moraju čekati odobrenje administratora.

Registracija

posoljeni

Inteligentno učenje

Pregled rezultata

4.4. Logika posluživanja pitanja

logika

5. Zaključak

Zaključak.

Razvoj podatkovnog sloja i aplikacijske logike za potrebe sustava elektroničkog učenja

Sažetak

Ovaj završni rad opisuje modernizaciju procesa učenja sustavom za inteligentno učenje. Rad počinje uvodom u edukacijske procese, zatim prikazuje povijest ideje automatizacije učenja te suvremeni model sustava za inteligentno učenje (ITS). ITS je opisan svojim komponentama, očekivanim karakteristikama i algoritmima uz kritički pogled na nedostatke ITS-a. Slijedi opis slojeva aplikacije koje je potrebno implementirati te opis implementiranog rješenja. On sadrži popis korištenih tehnologija, ER dijagram baze te mogućnosti koje sustav nudi korisniku koji uči te administratoru. Opis završava logikom posluživanja pitanja. Posljednje poglavlje rada je zaključak o prethodno iznesenim idejama.

Ključne riječi: Učenje, inteligentno učenje, informacijski sustav, ITS, znanje, prikaz znanja, parametrizacija, posluživanje pitanja, podatkovni sloj, aplikacijski sloj

Application logic and data layer development for an e-learning system

Abstract

This paper describes modernization of the learning processes using intelligent tutoring system. It begins with introduction to educational processes, it then describes the history of automazied learning and contemporary model of an intelligent tutoring system (ITS). The paper then describes layers of the application to be implemented and the implemented solution. This description contains a list of used technologies, ER diagram of the database and all options the system offers to both learning user and an administrator. It is followed by the description of the logic of serving questions. The last chapter is a conclusion about aformentioned ideas.

Keywords: Learning, intelligent learning, information system, ITS, knowledge, knowledge representation, parametrization, question serving, data layer, application logic