

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.

Zagreb, 11. ožujka 2016.

## **ZAVRŠNI ZADATAK br. 4350**

Pristupnik: **Alen Murtić (0036478740)**  
Studij: Računarstvo  
Modul: Programsko inženjerstvo i informacijski sustavi

Zadatak: **Razvoj podatkovnog sloja i aplikacijske logike za potrebe sustava elektroničkog učenja**

### Opis zadatka:

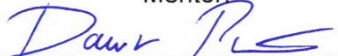
Edukacijski procesi proživljavaju mnoge promjene kao rezultat pojave modernih tehnologija. Sustavi inteligentnog mentoriranja imaju mogućnost procjenjivanja trenutnog stanja usvojenosti koncepata učenja te optimiziraju proces učenja kroz pružanje personaliziranih materijala i zadataka koji korisniku omogućuju učinkovito usvajanje koncepata vezanih uz domenu učenja.

Vaš zadatak je razviti podatkovni sloj i aplikacijsku logiku informacijskog sustava koji pruža usluge inteligentnog mentoriranja. Sustav mora pružiti podlogu za procjenu i reprezentaciju stanja usvojenosti koncepata te personalizirano posluživanje zadataka. Sustav treba sadržavati bazu parametriziranih zadataka povezanih sa konceptima učenja i logiku inteligentnog posluživanja zadataka te slaganja ispita tj. kolekcije zadataka koja odgovara zadanim uvjetima.

Zadatak uručen pristupniku: 18. ožujka 2016.

Rok za predaju rada: 17. lipnja 2016.

Mentor:



Doc. dr. sc. Damir Pintar

Djelovodja:



Doc. dr. sc. Ivica Botički

Predsjednik odbora za  
završni rad modula:



Prof. dr. sc. Krešimir Fertalj

*Zahvaljujem svojem mentoru, doc. dr. sc. Damiru Pintaru, na suradnji i uputama za izradu završnog rada, djelatnicima Fakulteta na nesebičnom prenošenju znanja iz svojih kolegija, obitelji koja me podržava u studiranju, a posebno mami na lektoriranju teksta rada.*

# SADRŽAJ

<b>1. Uvod</b>	<b>1</b>
<b>2. Sustavi za elektroničko učenje</b>	<b>3</b>
2.1. Inteligentni sustavi za elektroničko učenje (ITS) . . . . .	4
2.1.1. Povijest ITS-a . . . . .	4
2.1.2. Komponente ITS-a . . . . .	7
2.1.3. Karakteristike dobrih sustava . . . . .	9
2.1.4. Algoritmi sustava za inteligentno učenje . . . . .	13
2.1.5. Poznati sustavi . . . . .	15
2.1.6. Nedostaci ITS-a . . . . .	19
<b>3. Slojevi aplikacija</b>	<b>22</b>
<b>4. Opis implementiranog rješenja</b>	<b>24</b>
4.1. Korištene tehnologije . . . . .	24
4.2. ER dijagram . . . . .	24
4.3. Mogućnosti sustava . . . . .	25
4.3.1. Reprezentacija znanja . . . . .	26
4.3.2. Parametrizirani zadaci . . . . .	28
4.3.3. Mogućnosti administratora . . . . .	29
4.3.4. Mogućnosti korisnika . . . . .	30
4.4. Procjena znanja korisnika . . . . .	32
4.4.1. Procjena znanja za granule znanja . . . . .	32
4.4.2. Procjena znanja za koncepte . . . . .	33
4.4.3. Ocjena za predmete . . . . .	34
4.5. Logika posluživanja pitanja . . . . .	34
4.5.1. Algoritam posluživanja pitanja . . . . .	34
4.6. Konkretna pitanja u sustavu . . . . .	35

4.6.1. Test točnosti parametrizacije . . . . .	35
4.6.2. Znanje u sustavu . . . . .	36
<b>5. Zaključak</b>	<b>38</b>
<b>Literatura</b>	<b>40</b>

# 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 izumom tiskarskog stroja, osnivanjem sveučilišta, obveznim školstvom i masovnom proizvodnjom knjiga. Dostupnost 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 dovodi do toga da ljudska pažnja postaje 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 da 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.

Ovim završnim radom opisat ću 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 ću podatkovni i aplikacijski sloj sustava koji sam implementirao.

Implementirani sustav izgrađen je za korištenje u edukacijske svrhe predmeta s matematičkom pozadinom (kao npr. Osnove elektrotehnike, Fizika, Matematika). Trenutno je ispunjen pitanjima iz područja diskretne matematike (skupovi, kombinatorika, osnove diskretne vjerojatnosti), no implementacija je skalabilna te ne

postoji prepreka dodavanju novog znanja. Svrha implementacije je ispunjavanje zadatka zadanog na predmetu Završni rad s ciljem edukacije i demonstracija primjene teoretski opisanih elemenata u konkretnom sustavu za inteligentno učenje.

## 2. Sustavi za elektroničko učenje

Elektroničko učenje (e-učenje, *e-learning*) je prijenos znanja ili obrazovnog programa putem elektroničkih uređaja poput računala ili mobitela. E-učenje može prenositi informacije i materijale, provjeravati znanje te biti korišteno za postupno upijanje novih činjenica, baš kao na akademskim institucijama.[41] 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. [25]

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.[34] Samostalni sustavi za elektroničko učenje trebali bi imati dostupne sve elemente učenja jer samo tako mogu osigurati dovoljnu kvalitetu. Eventualno može nedostajati formalna provjera znanja ako je sustav dizajniran tako da je postupak upijanja novih činjenica rigorozan da ga korisnik ni u kojem slučaju ne može preskočiti ili ignorirati.

Neki od popularnih i besplatnih sustava za elektroničko učenje su Codecademy, Duolingo 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.[21] Duolingo je besplatan sustav za učenje stranih jezika u čijem održavanju, uz zaposlenike, sudjeluje i zajednica korisnika.[24] Na FER-u se uz Moodle također koriste AHyCo i Ferko.[27]



## **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-a je smanjiti ovisnost korisnika sustava o ljudskim učiteljima. Za razliku od ljudi, elektronički sustavi se ne umaraju, ne stare i gotovo uvijek imaju vremena za učenika. Inteligentni sustavi za elektroničko učenje sadrže 3 glavne komponente: domenski model, korisnički model i model učenja te korisničko sučelje.[28][43] 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.[31]

### **2.1.1. Povijest ITS-a**

#### **Mehanički strojevi za učenje**

Ideja inteligentnih sustava za učenje nije nastala tek u 21. stoljeću, dapače, već su u 17. st. dvojica važnih europskih matematičara, Blaise Pascal i Gottfried Wilhelm Leibniz razmatrali koncepte zaključivanja uz pomoć strojeva.[38][18] Pravi zamah elektroničko učenje dobiva u 20. stoljeću.

Prvi značajan korak prema učenju uz pomoć strojeva napravio je Sidney Leavitt Presley, profesor na Sveučilištu Ohio, koji je izumio stroj za samoocjenjivanje.[33] 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 ocjenjivanje. Slika 2.1 prikazuje stroj za ocjenjivanje na kojem se posebno ističu gumbi za odabir točnog odgovora između ponuđenih.

Iako su mehanički strojevi pružili kvalitetan početak ideje, jasno je da zbog nefleksibilnosti 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.

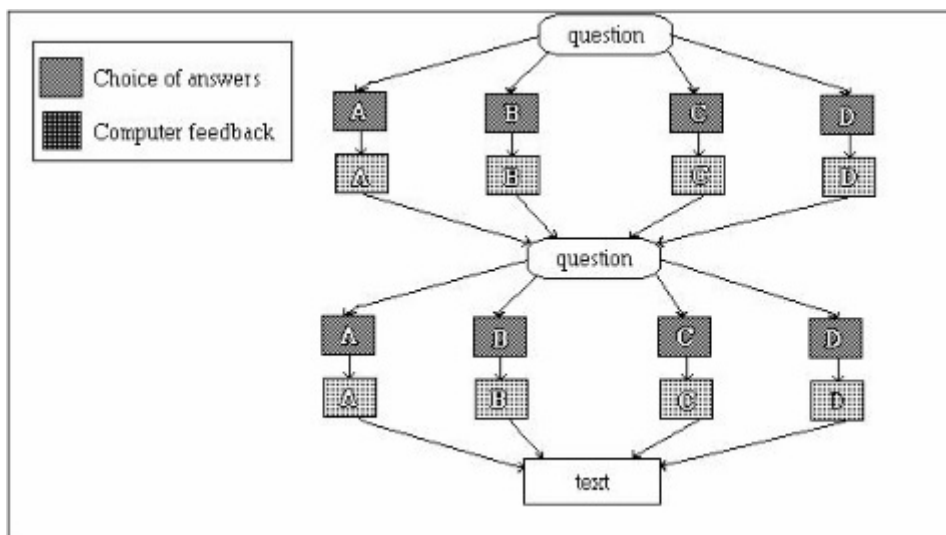
#### **Početak elektroničkih sustava za učenje**

U ranim godinama razvoja računala i računarstva izmišljeni su mnogi koncepti bliskih tematika, npr. Turingov test, ideja umjetne inteligencije i strojnog učenja, no koncept sustava za učenje kakav danas postoji je relativno zanemaren. Jedini važan



**Slika 2.1:** Stroj za ocenjivanje S. L. Presleyja[11]

napredak je definiranje gramatika u računarstvu, kao jednog od budućih načina parsiranja korisnikovih odgovora. No, dolaskom računala u širu upotrebu, u američkim školama se stvaraju CAI projekti (*Computer-Assisted instruction*) 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 Codecademy. U početku se CAI koristio u učionicama, a ne individualno, ali na sličnom principu korištenja.[43] Na slici 2.2 nalazi se predviđena shema rada prilikom korištenja CAI sustava.



**Slika 2.2:** Shema rada pomoću CAI sustava[3]

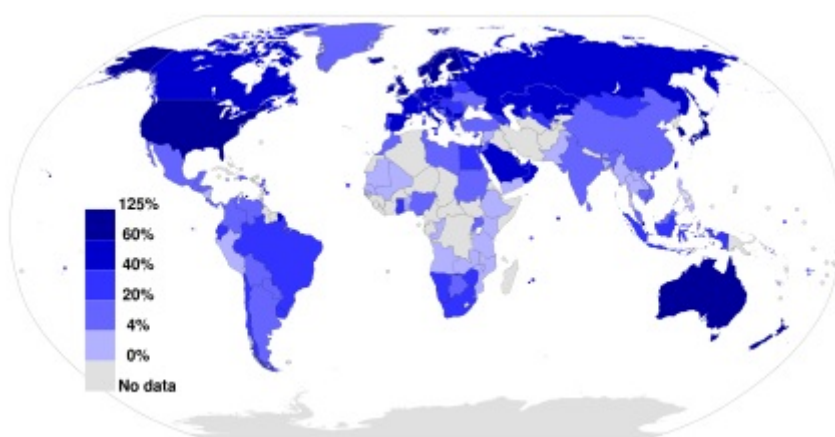
Velik korak za stvaranje ITS-a napravio je Jaime Carbonell koji je 1970. postavio

tezu da "sustavi za učenje ne moraju biti samo alat, nego i učitelj".[36] 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.[20] Izgradnja složenijih sustava je bila lakša zbog poboljšanja brzine računala, što je fokus rada konačno maknulo s tehnologije na tehniku.

Početkom 1980.-ih umjetna inteligencija pomiče se 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 radi na principu prilagođenog posluživanja instrukcija. ICAI je jako blizu modernom konceptu ITS-a, a jedina razlika između njih je što ICAI ne pokušava modelirati znanje korisnika.[43]

### 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. Slika 2.3 prikazuje broj korisnika mobilnog interneta s ciljem podržavanja dojma sveprisutnosti istog (osim, nažalost, u Africi). Slabost prijašnjih rješenja je bila u neadekvatnoj aktualizaciji sustava zbog činjenice da su se morali distribuirati na prijenosnim medijima. Nije slučajno da svi sustavi koje sam na početku nabrojao postoje kao internetske stranice.



**Slika 2.3:** Broj korisnika mobilnog interneta prema državama u svijetu[8]

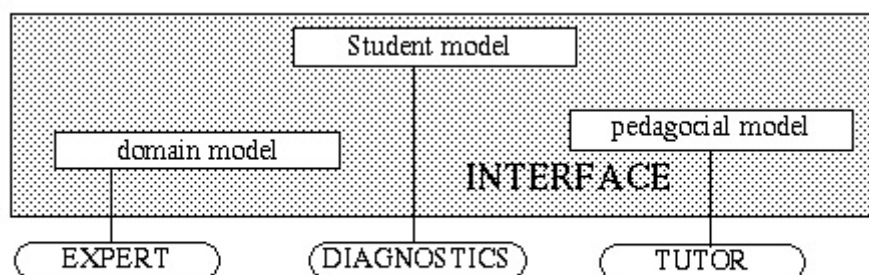
Najveći korak u razvoju inteligencije ITS-a je razvoj analize podataka. Iako su principi analize podataka bili matematički postavljeni i prije 1990.-ih, rastom brzine

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 kompleksnost procesa učenja (čime se bavi kognitivna psihologija), udaljenost 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 trebao biti spreman za nadogradnju temeljem iskustva korištenja.

Za razliku od umjetne inteligencije (UI), čije rezultate velike tvrtke sigurno mogu naplatiti, razvoj sustava za inteligentno učenje je donedavno bio guran gotovo isključivo od strane akademske zajednice. No, nove velike tehnološke tvrtke imaju potrebu educirati velik broj zaposlenika pa sustavi za učenje u današnje vrijeme konačno dobivaju zamah za napredak koncepta, ali i realizacije.[35]

### 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 nazivaju se modeli, zato što su pojednostavljena reprezentacija stvarnih pojava, koncepata i sl.[28] One nemaju nužno veze s modelom kao aplikacijskim slojem, iako se mogu oslanjati na njega. Tri najvažnije komponente ITS-a i njihovo značenje mogu se vidjeti na slici 2.4.



Slika 2.4: Komponente ITS-a[5]

#### Domenski model

Domenski model je dio sustava čiji je smisao organizirati znanje u jedinice spremne za pohranu u bazu podataka.[43] Svaki sustav ima osmišljenu 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 autorima i administratorima sustava preostaje definirati više ili niže razine u odnosu na koncepte. Budući da je izrada domenskog modela kreativan zadatak, u tom procesu treba sudjelovati što veći broj ljudi.

Domenski model ne uključuje samo podjelu znanja, nego i pravila te strategije koje korisnik treba naučiti. Oni se mogu prikazati eksplicitno formulama i materijalima za učenje ili implicitno, npr. redoslijedom prikaza pitanja. Implicitno zadavanje pravila i strategija tipično je za formalne školske sustave jer postoji određen redoslijed predavanja zbog čega nastavnici očekuju da učenici/studenti poznaju gradivo prethodnih satova 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 tu eksplicitnost često ne vidi.

Prilikom kreiranja domenskog modela važno ga je izraditi takvim da bude kvalitetna reprezentacija različitih vrsta znanja, da bude efikasan i razumljiv za osmišljavanje algoritama. Domenski model također mora moći odgovoriti na potrebe korisničkog modela.

Tipični domenski model je kvalitetno organizirana baza podataka s pravilno unesenim podacima. Za sve iznad toga brinu se ostale komponente, počevši s 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 omogućavaju prikaz analize odgovora korak po korak.[43] Zbog relativno male opsežnosti, neki izvori zanemaruju korisnički model, cijepajući njegove dijelove u domenski i model učenja. Algoritme korisničkog modela opisat ću u zasebnom poglavlju.

### **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. Model učenja

mora prikupljati podatke o korisniku, znati procijeniti znanje korisnika te isto prikazivati nekom opisnom ili grafičkom metodom.[43]

Ovoj komponenti pripadaju algoritmi koji na temelju korisnikova odgovora evaluiraju njegovo znanje, kao i algoritmi 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 nazivaju se UI-potpuni. Nakon što algoritmi procijene znanje korisnika, ono se treba prikazati nekim vizualnim načinom kako bi korisnik bolje razumio što ne zna i zašto mu sustav postavlja neka pitanja.

Model učenja srž je ITS-a te glavni odraz kvalitete pojedinog sustava. Logičke produkcije koje dovode do zaključka je li korisnik nešto naučio ili ne čest su način implementacije modela učenja.[45] Formalna evaluacija korisnika i samog sustava prednost je takvog pristupa, no njegov nedostatak je nejasnost širem spektru administratora. Alternativni način implementacije modela učenja je sustav bodovanja pojedinih pitanja, skupova pitanja i odnosa između koncepata koje korisnik treba savladati.

### **Korisničko sučelje**

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

### **2.1.3. Karakteristike dobrih sustava**

Nakon definiranja komponenata ITS-a, potrebno je razmotriti karakteristike kvalitetnih sustava za inteligentno učenje. Većina su obvezni dijelovi ITS-a, zbog čega ih je važno navesti.

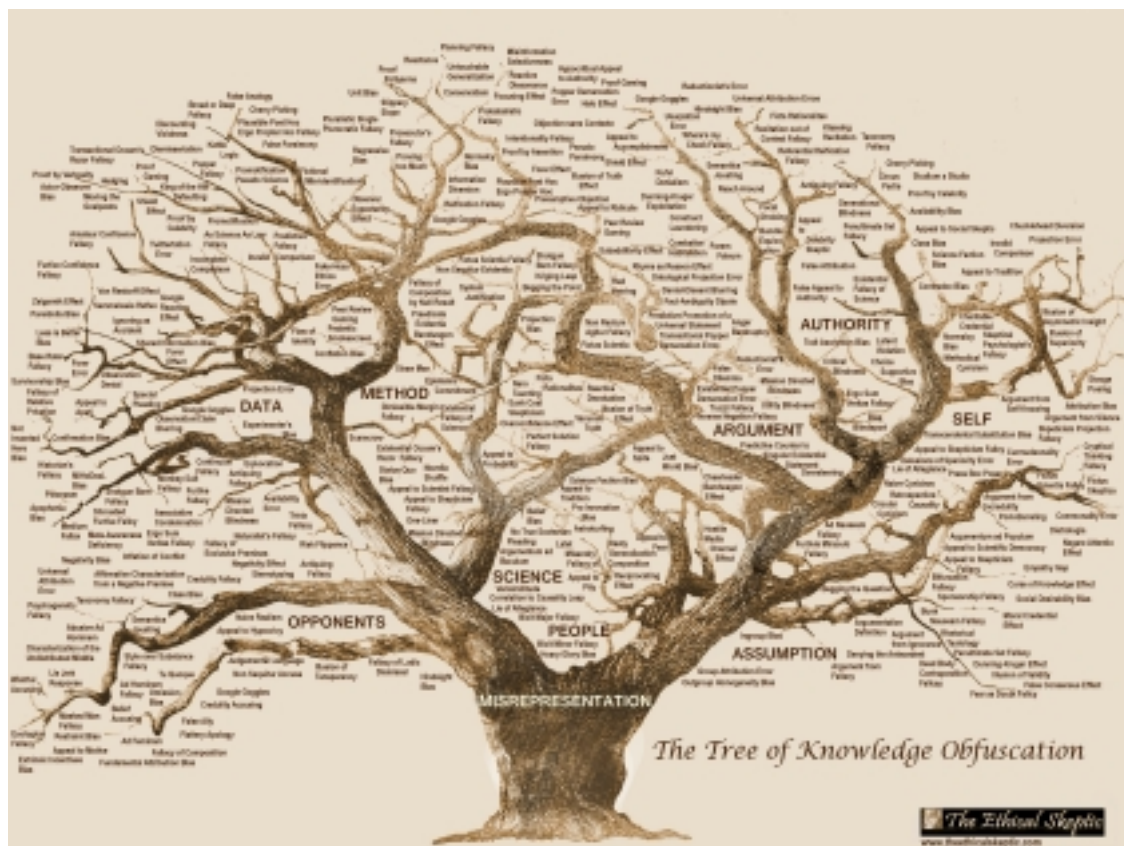
#### **Razrađena podjela znanja**

Iako se ovo čini kao očekivan uvjet iz opisa domenskog modela, treba ga ponoviti i u ovom dijelu. Ispravna podjela znanja omogućava lakše i kvalitetnije projektiranje

algoritama procjene korisnika.[22] 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 tako da svaka od razina ima reprezentativni uzorak ocjene korisnika.

Elegantno rješenje problema određivanje je srednje velikog broja granulacija (jasno, pojam srednjeg broja ovisi o opsegu domene za koju se ITS projektira), a onda između pojedinih granula uspostaviti različiti broj odnosa.[19] Na taj način i ja ću riješiti problem u svojoj implementaciji ITS-a. Slika 2.5 ilustrira podjelu ljudskog znanja.



Slika 2.5: Ilustracija: drvo znanja[4]

## **Materijali za učenje**

Svaki sustav, čiji je cilj samostalnost, trebao bi sadržavati materijale za učenje poput e-knjiga ili uputa koju bi literaturu korisnik morao proučiti prije odgovaranja. Samo takav pristup sa sigurnošću omogućava korisniku 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.[39]

Parametriziranje pitanja je postupak kojim se numerička pitanja (moguće i na tekstualnim, ali teže) 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 omogućava jedinstvene vrijednosti različitim korisnicima, nego i različite vrijednosti istim korisnicima. Ukoliko neki korisnik želi ili mora više puta rješavati neko pitanje, mogu mu se servirati jedinstvene vrijednosti parametara svaki put kada ga otvori. Jedan od načina ostvarivanja parametrizacije pitanja je spremanje formule kao odgovora te popisivanja vrijednosti parametara i njihovih ograničenja (npr. imaginarni dio broja mora biti različit od 0, apsolutna vrijednost mora biti  $> 0$  i sl.).

## **Praćenje tipkanja i reakcija lica korisnika**

Važna karakteristika dobrog učitelja je uočavanje učenikovih reakcija, čitanje fizičkih znakova te zaključivanje iz njih. 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 najrazvijenija načina su praćenje tipkanja i praćenje 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



je važno pratiti u kojim je situacijama korisnik ubrzan, o kojim pitanjima dulje razmišlja te je li mijenjao neki odgovor.[16]

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 praćenja izraza lica je u tome što svaki korisnik mora imati neki oblik web kamere koja će ga snimati.

Oba načina praćenja reakcija dio su 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.[30] 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).[28]

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 svoje 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 dopuštenje korisniku da zatraži uputu ako sam shvaća da nema ideju odgovaranja. Nedostatak prvog rješenja je moguća prevelika nepotrebna složenost, a drugog to što umorni korisnici ponekad od odgovaranja odustaju prerano.

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

### **Sigurnost**

Sigurnost je jako važan element svakog sustava, a ističe se 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 zaštititi nekim sustavom lozinki.[37]

Sprječavanje lažnog predstavljanja puno je 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 može dovesti do gotovo potpunog eliminiranja samostalnog učenja. Tako da ni 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. U sljedećem poglavlju više ću govoriti o algoritmima sustava za elektroničko učenje, no bilo ih je važno navesti i u ovom.

### **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 točno definirati što je kvalitetno grafičko korisničko sučelje, no ono bi sigurno moralo imati sljedeće karakteristike: intuitivan raspored kontrola, lakoća upravljanja (brzina, dosljednost, dozvole promjena kontrola), ugodan izgled (boje, fontovi, izbjegavanje neugodnih efekata i promjena boja) te prikladan prikaz informacija.

Korisničko sučelje ITS-a također bi se trebalo 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  $\text{csin}$  za kosinus, točka ili zarez kao decimalna oznaka) te prikaz napretka u pojedinoj granulaciji znanja.[29]

#### **2.1.4. Algoritmi sustava za inteligentno učenje**

Najprije sam opisao dijelove inteligentnog sustava za elektroničko učenje te koji principi moraju biti poštovani prilikom implementacije jednog takvog sustava, no sljedeće je pitanje što sve treba implementirati da bi sustav bio inteligentan? Najbolji je odgovor objasniti 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. Upravo je on najvažniji element umjetne inteligencije ITS-a.[28] Može biti oblikovan na različite načine, no dva su najčešća: logičkim produkcijama koje nadopunjuju produkcije procjena korisnikova znanja ili samostalnim programskim odabirom. 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, zahtjevnost 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 svodi se na poziv 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  $\pm n\%$  s korisnikovim rješenjem.

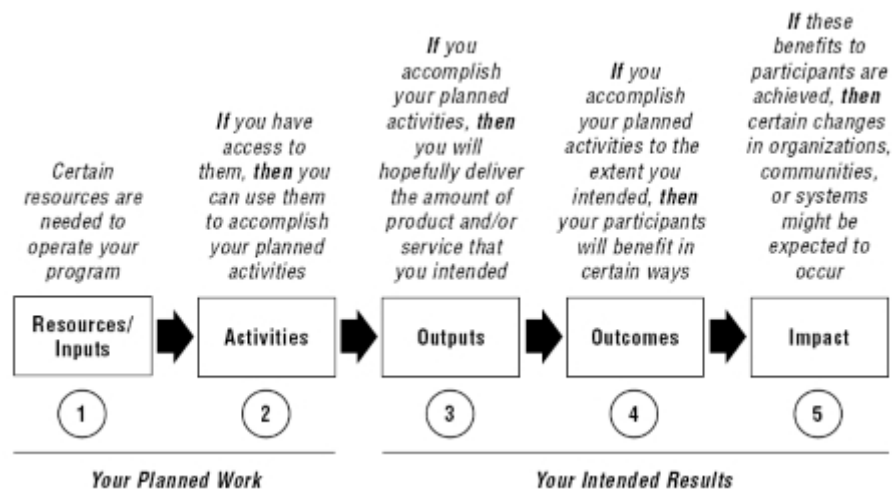
## **Procjena znanja**

Algoritmi procjene znanja mogu se temeljiti na različitim pogledima na korisnikov odgovor. Neki od najzanimljivijih faktora je li korisnik nešto naučio ili nije 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 te 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 je neke granulacije korisnik naučio. Za svaki njezin dio izbacuje se vrijednost 1 (naučeno) ili 0 (nenaučeno). Programska alternativa može raditi isti taj posao drukčijom tehnologijom, oslanjati se na praćenje korisnika ili donositi odluke na nekom trećem principu. Prednost programskog nad logičkim izračunavanjem je što programski u pravilu može elegantnije doći do neke ocjene i što ne mora ovisiti o konkretnim granulacijama. Slika 2.6 prikazuje općenitu shemu redoslijeda u logičkoj evaluaciji.



**Slika 2.6:** Općenita shema logičke evaluacije[6]

## 2.1.5. Poznati sustavi

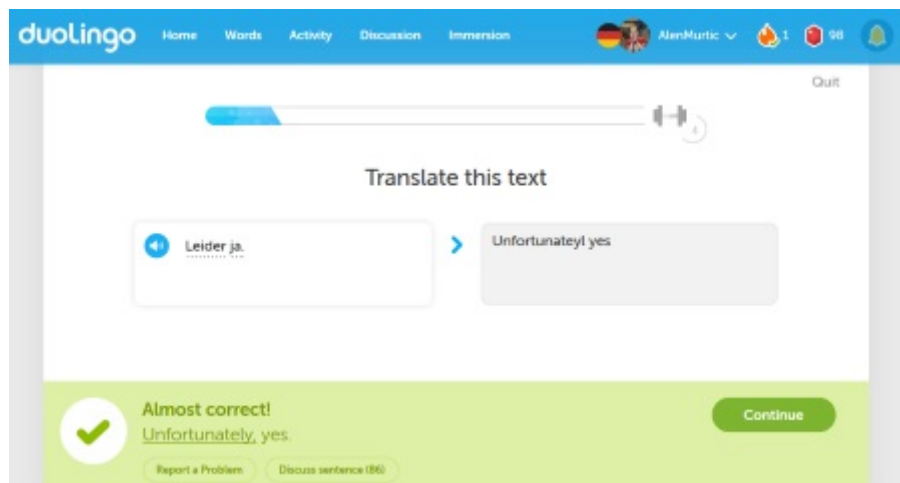
Opisao sam što sve treba napraviti prilikom dizajniranja sustava za inteligentno učenje, no još nisam iznio što su pametni ljudi prije mene radili u tom pogledu. A radili su puno, stoga bi bilo šteta propustiti izvući znanje i iskustva njihovih sustava u cilju dizajniranja kvalitetnog ITS-a. Čak i neinteligentni sustavi mogu ponuditi korisne

zaključke, možda čak i jasnije nego inteligentni. Zašto? Zato što su konceptualno relativno jednostavni te se kod njih lakše mogu uočiti neki problemi.

Bitno je naglasiti da nisu svi sustavi u ovom dijelu samo pitalice, već sadrže neke slobodnije poglede na učenje zato što 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 da dobije ponudu jezika za koje postoji tečaj uz pomoć njegova materinjeg jezika.[24] Tipovi pitanja mogu biti nadopunjavanje, prevođenje ili pitanje s odabirom od ponuđenih odgovora. Sustav je neinteligentan osim u dva pogleda: znanje nekog područja zastarjeva ako korisnik dugo nije odgovarao na pitanja tog područja te sustav odabire područje s najmanjom vrijednosti znanja ukoliko korisnik odabere opciju "pojačavanja znanja" (engl. strengthen skills). Na slici 2.7 može se vidjeti prepoznavanje pogreške tipkanja u sustavu Duolingo.



**Slika 2.7:** 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 dobiti potvrdu svog znanja.[21] Jasno je da položiti neki tečaj na Courseri nije jednakovrijedno kao završiti kolegij na

fakultetu, no ako korisnik ima stvarnu volju za učenjem, količina stečenog znanja može biti jednaka. Iako su materijali velikog broja fakulteta (pogotovo tehničkih) javno dostupni na njihovim web-stranicama, 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.[44] 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. Urban Jungle odličan je primjer da računalni sustav namijenjen za učenje ne mora nužno biti samo niz pitanja i odgovora. Slika 2.8 prikazuje ekran tijekom simulacije vožnje u aplikaciji Urban Jungle.



**Slika 2.8:** Slika ekrana prilikom vožnje u Urban Jungle edukativnoj igri[13]

Moodle je vrlo popularan akademski sustav za učenje koji nudi neke koncepte ITS-a (parametrizirana pitanja, interaktivno sučelje), no nije ITS jer ne sadrži inteligentno odlučivanje.[34] Sustav Moodle gotovo uvijek funkcionira kao podrška formalnom sustavu učenja zato što nudi 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 moguće lako prilagođavati. Sustav Ferko razvijen na Fakultetu elektrotehnike i računarstva ima sličnu domenu primjene kao Moodle. Slika 2.9 prikazuje logo sustava Moodle, a slika 2.10 sustava Merlin.



**Slika 2.9:** Logo sustava Moodle[9]



**Slika 2.10:** Logo sustava Merlin temeljenog na Moodleu[7]

### **Inteligentni sustavi**

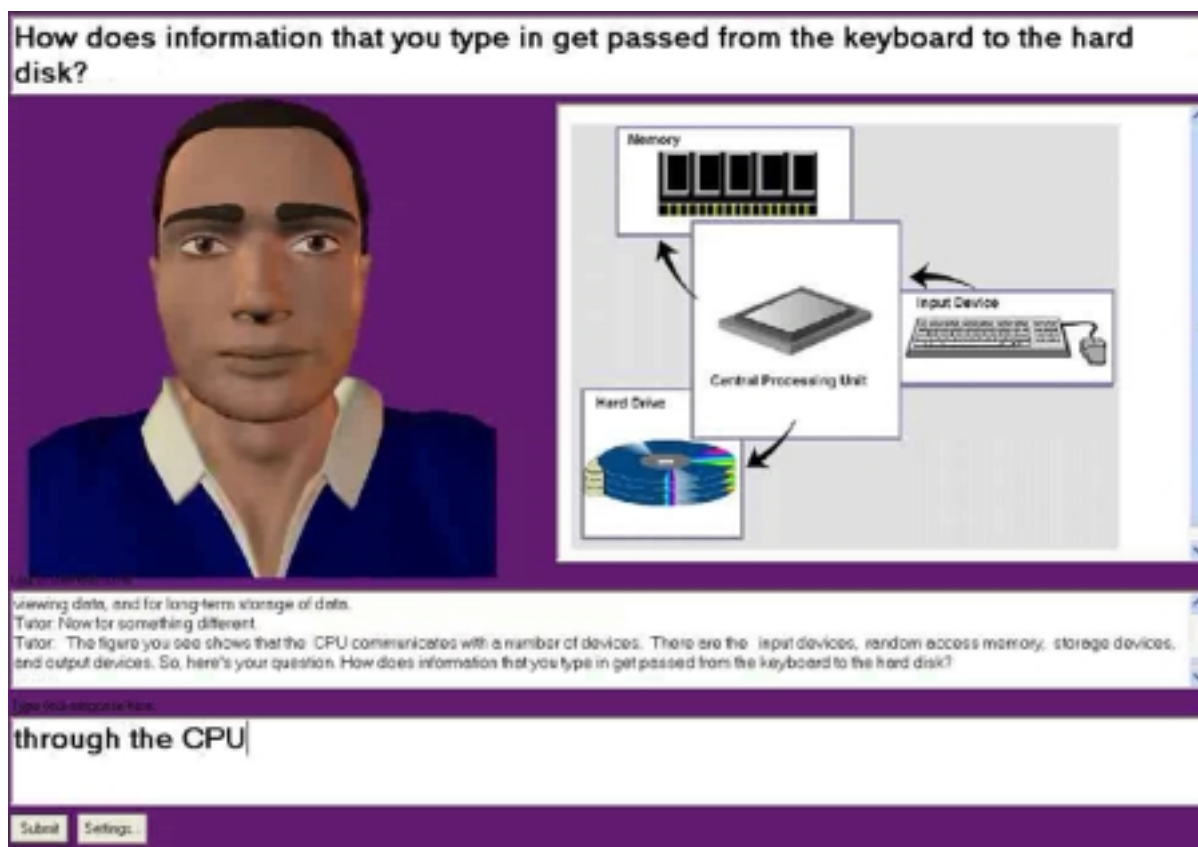
Postoje mnogi sustavi za inteligentno učenje koji rade na principu pitalica te ću zato opisati najzanimljivije primjere ITS-a koji možda i ne slijede dosad navedeni obrazac, no svejedno pripadaju skupini sustava za inteligentno učenje.

Why2-Atlas je ITS koji uči korisnika koncepte kvalitativne fizike tako da od njega traži da napiše nekoliko rečenica o 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 njegovo pogrešno.[15]

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.[32] Prikaz sučelja sustava AutoTutor nalazi se na slici 2.11.

ActiveMath je ITS vrlo sličan prototipskom pogledu na sustave za inteligentno učenje. To je sustav za dodatno podučavanje 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.[14] Slika 2.12 prikazuje korisničko sučelje sustava ActiveMath.

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 uočiti je najčešće obrasce ponašanja igrača te nove protivnike stvoriti takvima da igrač mora odabrati drukčiju strategiju. Mnogo



Slika 2.11: Sučelje razgovora sustava AutoTutor[2]

računalnih igara tvrde da imaju prilagodljivu UI, no zasad taj koncept nije posebno 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.[40]

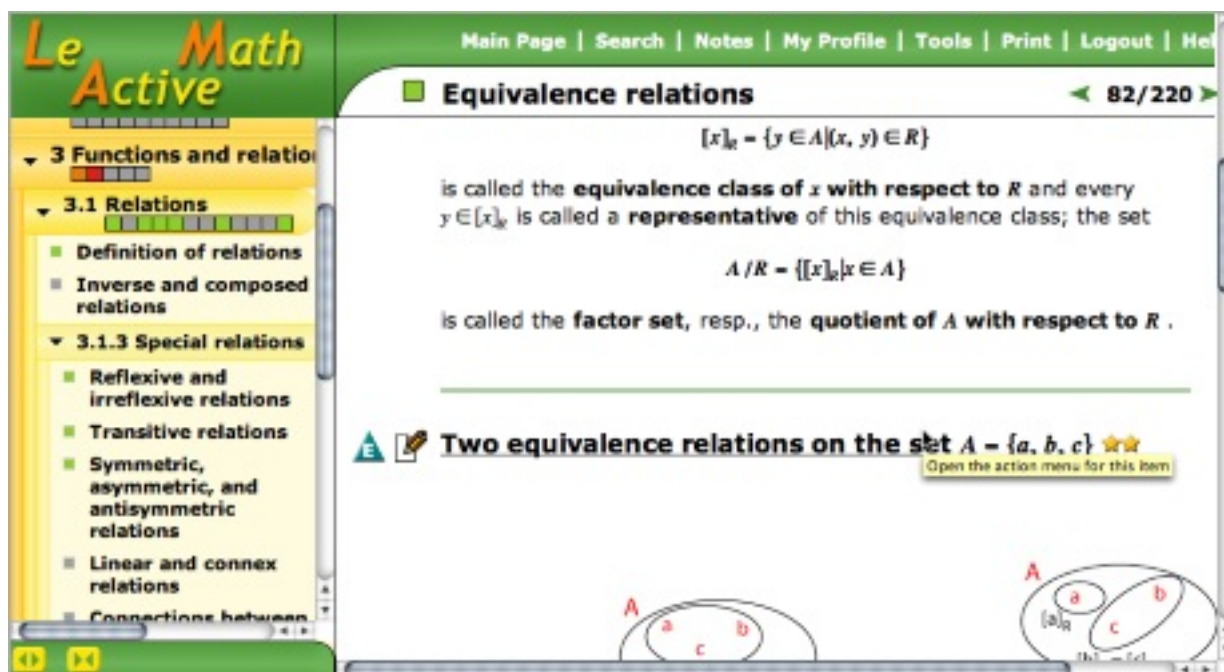
## 2.1.6. Nedostaci ITS-a

### Podučavanje i inteligencija

Metode uspješnog podučavanja očito su 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





Slika 2.12: Slika ekrana ActiveMath sustava[1]

procijeniti je li korisnik samo naučio "kuharicu" ili je stekao veću razinu znanja. 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 te 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.[17]

Naravno da je i problem ITS-a što autori pitanja jednostavno moraju odlično razumijevati tehničku pozadinu sustava (smisao pojedine granulacije, veze između granula). 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 u takvim situacijama značajno je veći.

## Cijena izrade

Nadam se da je čitatelju ovog završnog rada jasno da je izrada kvalitetnog ITS-a vrlo složen posao. Čak ne niti toliko težak, koliko opsežan. Kod izrade ITS-a nužna je bliska suradnja između računalnih stručnjaka i učitelja domene primjene. Velik broj ljudi znači velik broj radnih sati, što znači da ih treba više platiti. Ukoliko vlasnik sustava ne očekuje da će mu ITS značajno povećati broj i znanje korisnika, logično je postaviti pitanje opravdanosti investicije, pogotovo imajući u vidu cijenu održavanja sustava i rizik pogreške u sustavu. Dosad je ITS-ove uglavnom razvijala akademska zajednica. Na slici 2.13 može se vidjeti ilustracija timskog rada potrebnog za projektiranje kvalitetnog sustava za učenje.



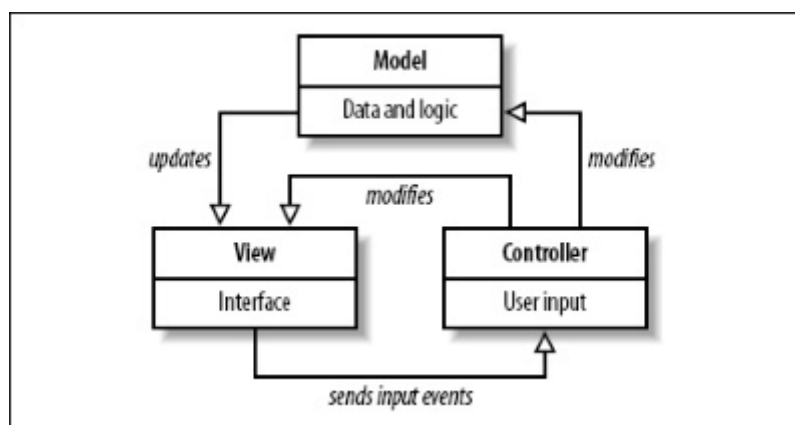
**Slika 2.13:** Ilustracija timskog rada potrebnog za dobar ITS[12]

Cijena izrade zapravo i nije nedostatak, njezin problem je što se mora platiti prije nego što sustav počne s radom. Dugoročno je ITS jeftiniji nego financiranje prostora, profesora, opreme učenika i sličnih elemenata tradicionalnog edukacijskog procesa.[17]

Još jedan izazov izrade je kompleksnost sustava. Sustav za učenje jednostavno mora biti efikasan i brz, a sustav koji prati sve faktore to 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. Slojeva arhitektura omogućava sigurno zadržavanje dva najvažnija logička načela dobrog oblikovanja: nadogradnju bez promjene i načelo jedinstvene ovisnosti. Takav pristup intervencije u kod aplikacije čini lakšim. Arhitektura koju ću koristiti u implementaciji sustava 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 višepatformska 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.[23] Slika 3.1 prikazuje dijagram komunikacija u MVC modelu.



**Slika 3.1:** Komunikacija u MVC modelu[10]

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 sloj je 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. *utility* sloj rada s bazom, sloj koji prima podatke od pogleda te algoritamski sloj koji provodi operacije. U ovom završnom radu aplikacijska logika će 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 implementacija tog sloja aplikacije, korisničko sučelje opisat ću samo na način komunikacije između sustava i korisnika.

Uspoređujući slojeve inteligentnog sustava za elektroničko učenje i MVC višeslojne aplikacije, vidimo jasnu povezanost domenskog modela i MVC modela, korisničkog modela, modela znanja i controllera te grafičkog sučelja i pogleda. Upravo to je smisao MVC pristupa. Zašto onda navodim obje arhitekture komponenata jednog ITS-a? Zato što je MVC tehnološki presjek, a slojevi ITS-a logički. Njihovo poklapanje je potvrda da je aplikacija projektirana na ispravan način.

## 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 znanja pojedinog korisnika, aplikacijska logika posluživanja pitanja te logika slaganja ispita. U ovom poglavlju opisat ću sve navedene elemente.

### 4.1. Korištene tehnologije

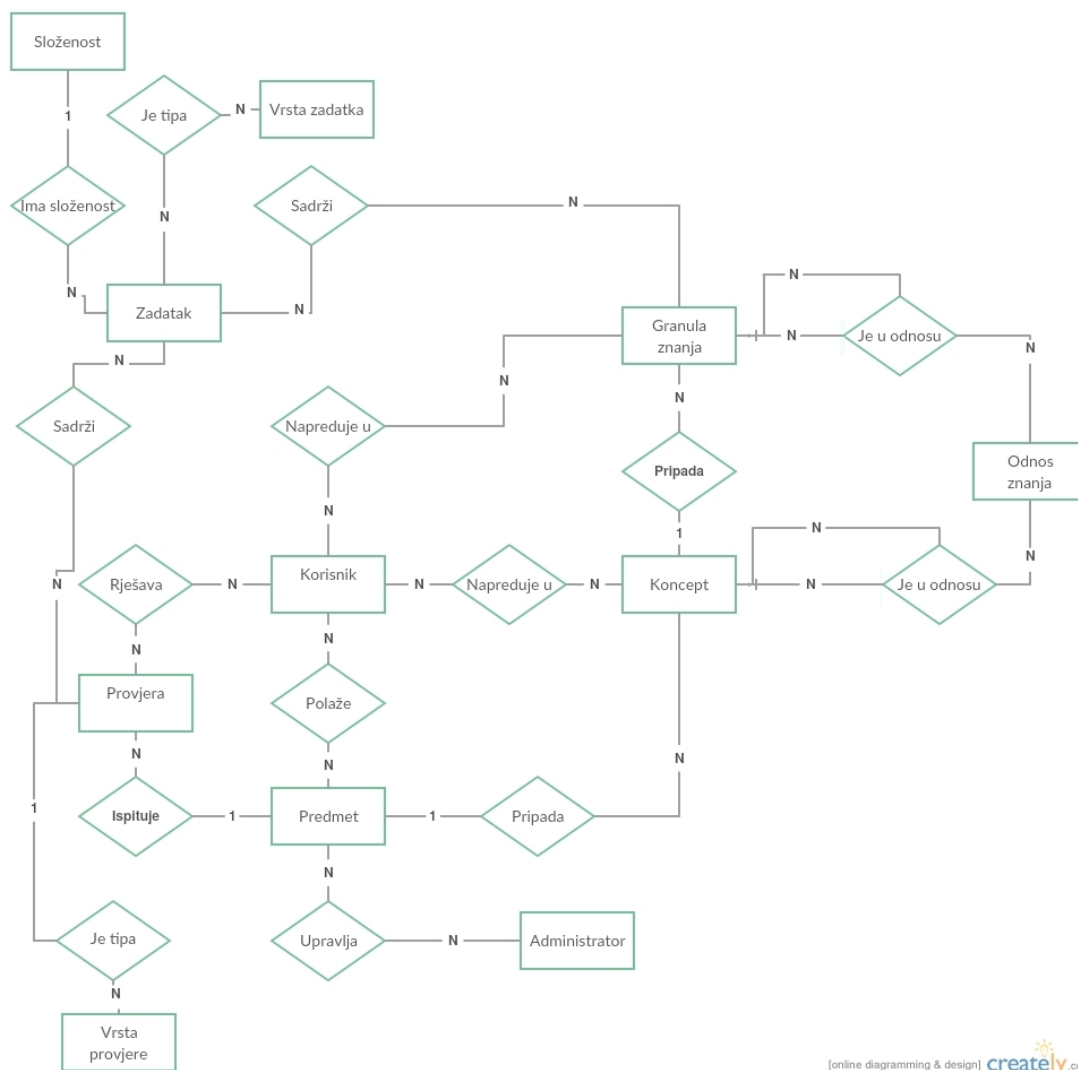
Budući da je zadatak sadržavao izradu baze podataka i aplikacijske logike za sustav inteligentnog učenja, u izradi programskog rješenja koristio sam nekoliko različitih tehnologija i to:

- 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 te generiranje dijagrama entiteta.
- Biblioteku **mXparser** za izračunavanje parametriziranih matematičkih izraza.
- Program **Creately** za izradu ER dijagrama.

### 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. ER dijagram prikazan je na slici 4.1, a dijagram entiteta na slici 4.2. na kraju ovog poglavlja.



**Slika 4.1:** ER dijagram baze

### 4.3. Mogućnosti sustava

Mogućnosti sustava podijeljene su na mogućnosti administratora, korisnika koji uči i zajedničke mogućnosti. I za administratora i za 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 važna je registracija na sustav, način inteligentnog učenja te pregled vlastitih

rezultata.

### 4.3.1. Reprezentacija znanja

U drugom poglavlju ovog završnog rada između ostaloga iznio sam tezu o važnosti kvalitetne granulacije znanja u sustavu za inteligentno učenje. Stoga sam njoj posvetio velik dio planiranja implementacije sustava.

#### Granulacije znanja

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

Predmet najveća je granulacija te reprezentira sličnu količinu znanja kao jedan fakultetski kolegij, iako to nije nužno ukoliko se sustav ne koristi u svrhu formalne provjere. Entitet predmet logičan je 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. 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 prikazivanju nekih specifičnih postupaka, manjih dijelove većih pojmova ili bilo kojih drugih nedjeljivih cjelina čije se značenje pojavljuje u pojedinim zadacima. Također su vezane, eksplicitno s konceptom, a time implicitno i s predmetom.

Primjer koji nisam obradio u ovom završnom radu, a čitatelju bi mogao pomoći u razumijevanju predviđene strukture podjele je sljedeći:

- Predmet: Umjetna inteligencija
  - Koncept: Pretraživanje prostora stanja
    - \* Granula znanja: Pretraživanje u dubinu
    - \* Granula znanja: Pretraživanje u širinu
    - \* Granula znanja: A-zvezdica pretraživanje
  - Koncept: Igranje igara
    - \* Granula znanja: Minimax algoritam

- \* Granula znanja: Minimax s podrezivanjem

- Predmet: Osnove elektrotehnike

- Koncept: Magnetizam

- \* Granula znanja: Coulombova 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 njemu. Ukoliko dva različita predmeta imaju povezane koncepte (npr. Fourierova transformacija u MAT3 i SIS koji su dio 2. godine FER2 programa), želio sam 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 sam jednostavno želio 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 jednake 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 slijedeće granulacije prije postizanja prihvatljive razine znanja u baznoj. Prihvatljiva razina znanja ekvivalent je ocjene dovoljan na pojedinoj granulaciji.
- **Podskup (nadskup)** Odnos između dvije granulacije u kojem 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** Administrator može definirati odnos korištenja ako se znanje jedne granulacije mora nužno koristiti u drugoj. Takav odnos znači da bit nekog znanja nije u povezanom znanju, no to povezano znanje je neizostavno koristiti da bi se došlo do rješenja.
- **Analogno** Ovaj odnos može se definirati ako postoji povezanost između dvaju pojmova takva da znanjem jednog postoji osnovno razumijevanje drugog. U tom 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. Nju sam izveo 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 zadatka 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 decimalne.

Parametre uz poštovanje uvjeta generira aplikacijska logika. Parametri su decimalni brojevi predodređenog raspona između atributa minParametar i maxParametar koji su trenutno postavljeni na 0 i 120. Svaki parametar može odrediti svoju domenu uvjetima, a niti jedan od parametara trenutno sadržanih zadataka ne nalazi se izvan tog skupa. Jedina svrha tih dvaju atributa je optimizacija jer nema smisla generirati brojeve koji nikada neće zadovoljavati uvjete parametara.

### **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% 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 mišljenja sam da je važno i eksplicitno napomenuti.

#### **Registracija korisnika na predmet**

Prilikom registracije u sustav korisnik nije povezan s predmetima, već ga administrator mora dodati na predmet kojim upravlja. Administrator ne može izbrisati korisnika s 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 zadataka ako administrator želi imati veću kontrolu nad generiranjem ispita. 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. Ako 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 ili nastavka generiranja ispita. 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**

Korisnici mogu odgovarati samo na pitanja iz predmeta kojima pripadaju, baš kao što administratori mogu nadzirati samo predmete na kojima rade. Za registraciju na pojedini predmet korisnici moraju čekati odobrenje administratora.

### **Registracija**

Novi korisnici koji žele učiti mogu se registrirati na sustav te od administratora predmeta zatražiti pristupanje nekom predmetu. Nakon uspješne registracije i dodavanja predmeta, korisnik odmah može započeti proces učenja. Lozinku korisnika sustav sprema na siguran način posoljenim hash SHA256 algoritmom, kao što se očekuje od svakog ozbiljnog ITS-a.

### **Inteligentno učenje**

Svi korisnici učenje počinju od samog početka što znači da ne postoji mogućnost prenošenja rezultata iz nekog drugog ITS-a. Korisniku se nudi inteligentno učenje na pojedinom predmetu, tj. mogu se generirati samo ona pitanja čije su granule znanja povezane s konceptima točno određenog predmeta. No, odgovaranje na pitanja jednog

predmeta može dovesti i do promjene stanja na drugom predmetu u slučaju da je baza znanja koncipirana tako da su koncepti ili granule znanja jednog predmeta u odnosu s onima drugog.

Bit inteligentnog učenja je u praćenju korisnikova znanja i odabiru pitanja što se ponajprije očituje u tome da korisniku sustav ne smije postavljati pitanja na koja još ni u kojem slučaju ne može znati odgovor. Zbog toga su vrlo važni složenost pitanja i odnos preduvjeta. No, o konkretnom načinu posluživanja pitanja više ću reći u sljedećem poglavlju.

Korisnici mogu pisati dvije vrste provjere: vježbe i ispite. Vježbe su provjere kojima se ostvaruje napredak u sustavu na kojima korisnik može više puta griješiti bez dugoročnih posljedica po konačnu ocjenu iz predmeta. Od korisnika se očekuje da stalno otvara vježbe, uspješno napreduje te kad se osjeća sigurnim, zatraži generiranje ispita za određeni predmet. Prvi ispit ne znači završavanje ili padanje predmeta, već znači samo to da se njegovi bodovi pamte. Sam sustav određuje koja su pitanja najprikladnija za nekog korisnika na pojedinom predmetu. Zadaci u sustavu mogu se dijeliti na ispitne i zadatke za vježbu, s obzirom na to u kojim tipovima provjera se mogu pojaviti. Svi zadaci inicijalno zapisani pripadaju objema grupama.

Ispiti su formalne provjere kojima se polažu predmeti. Generiranje ispita zadaje administrator ili traži korisnik. Korisnik sve teoretski može polagati putem ispita, no to nema previše smisla zato što u reprezentaciji znanja nema veće koristi u rješavanju ispita, oni su tu samo za formalno određivanje ocjene. Ispiti koje traži korisnik također se generiraju putem algoritma za inteligentno učenje. Za reprezentaciju korisnikova znanja nema razlike u odgovaranju na isto pitanje u vježbi i ispitu.

## **Pregled rezultata**

U svakom trenutku izvan odgovaranja na provjeru, korisnik može pogledati svoje rezultate na nekom predmetu. Mogu mu se ispisati detalji provjera kojima je pristupio (ne i konkretna pitanja koja je na njima vidio), trenutna predviđena ocjena na tom predmetu i prikaz znanja po konceptima tog predmeta.

Osim svojih rezultata korisnik može vidjeti i sve granule znanja pojedinog koncepta na predmetu te veze koje taj koncept ima.

## 4.4. Procjena znanja korisnika

Algoritam procjene znanja radi u dva koraka, najprije procjenjuje znanje pojedinih granula, a zatim na temelju te procjene procjenjuje znanje koncepata. Algoritam se aktivira nakon što korisnik završi rješavanje provjere. Ako se tijekom vježbe ili ispita dogodi neočekivani prekid rada aplikacije, korisnikovo znanje granula i koncepata ostaje isto kao prije generiranja, no na ispitima to znači da je korisnik osvojio 0 bodova.

### 4.4.1. Procjena znanja za granule znanja

Procjena usvojenosti znanja granula temelji se na vezi zadatka i granule, složenosti zadatka i odnosima između granula. Točno odgovoreni zadatak korisnikovom znanju pojedine granule dodaje broj složenosti tog zadatka. Npr. prvim odgovaranjem na osnovno pitanje iz nekog pojma koje ima broj složenosti 1, korisnikovo znanje granule se povećava s 0 na 1. Ako je odgovor netočan, korisnikovo znanje smanjuje se za isti broj, tj. u ovom primjeru bilo bi -1.

Nakon dodavanja znanja osnovnoj granuli zadataka algoritam iterativno prolazi po svim odnosima te granule na sljedeći način:

- Ako je granula podskup neke druge granule, granuli koja je sadrži, na jednak se način dodaje točnost/netočnost odgovora.
- Ako granula koristi neku drugu granulu prilikom rješavanja zadatka (npr. diskretnu vjerojatnost je nemoguće izračunati bez poznavanja osnova kombinatorike), korištenoj granuli dodaje se ili oduzima  $\text{faktorKorištenja} \times \text{složenost}$ . Trenutni faktor korištenja postavljen je na 0.4, no postoji mogućnost promjene temeljem iskustva (promjena nije automatizirana).
- Ako postoji granula čije je znanje analogno znanju trenutne granule, korisnikovo znanje analogne granule mijenjamo za  $\text{faktorAnalognosti} \times \text{složenost}$ . Trenutni faktor analognosti je 0.2. Valja naglasiti da analognost nije dvosmjerna, tj. ako administrator želi dvosmjernu analognost, u bazi mora postaviti da je X analogno s Y i Y analogno s X.

Granule u bazi sadrže atribut ukupna složenost koji se računa kao suma složenosti zadataka te granule + ukupna složenost granula koje su njezini podskupovi.

## Prednosti ovakvog pristupa

Razlozi za ovakav pristup procjeni znanja granula su: trajnost znanja, inkrementalni pristup i brzina izračuna.

Trajnost znanja očituje se u tome da jedna pogreška u sustavu znači manje nego višestruki broj točnih odgovora, no istovremeno znači da se korisnika koji ne zna odgovoriti na teža pitanja, vraća na lakša. Znanje korisnika mapira se od minimalnog do maksimalnog *double* broja.

Inkrementalni pristup očituje se u algoritmu odabira pitanja koji procjenjuje složenost trenutnog znanja korisnika te na temelju nje poslužuje pitanja najbliže složenosti. Nakon što korisnik odgovori na ta pitanja, točni odgovori povećavaju njegovo znanje te se nude pitanja više složenosti, a netočni odgovori ga vraćaju na nižu razinu (uz uvjet da je odgovorio na dovoljan broj pitanja jedne granule).

Brzina izračuna nije bila presudni faktor projektiranja baš ovakvog algoritma, no ipak je izuzetno važno da sustavu ne treba previše vremena za aktualizaciju pa je ona ugodna nuspojava ovog pristupa.

### 4.4.2. Procjena znanja za koncepte

Procjena znanja za koncepte slijedi tek nakon procjene znanja granula. Ona se računa na sljedeći način:

1. Za sve granule koje pripadaju konceptu računa se postotak usvojenosti.
2. Za granule koncepata koji su u odnosu s trenutnim također se računa postotak usvojenosti.
3. Izračunava se težinski prosjek postotaka usvojenosti sljedećom formulom:

$$U = \sum_{i=1}^n T_i * G_i / \sum_{i=1}^n T_i$$

U - postotak usvojenosti, n = ukupni broj granula iz 1. i 2. korak,  $T_i$  = težina,  $G_i$  - postotak usvojenosti i-te granule. Težine pojedinih granula su:

- $T$  = faktorVažnosti ako granula pripada trenutnom konceptu ili konceptu koji je podskup trenutnog.
- $T$  = faktorKorištenja\*faktorVažnosti ako koncept koristi neki drugi.
- $T$  = faktorAnalognosti\*faktorVažnosti ako je koncept analogan nekom drugom konceptu

Trenutni faktor važnosti postavljen je na 5.

## **Prednosti ovakvog pristupa**

Dvije najbitnije prednosti ovakvog pristupa su razumljivost prikaza i sigurnost točnog izračuna. Razumljivost prikaza napretka korisnika samom korisniku vrlo je značajna za točnu predstavku o vlastitome znanju. Postotni prikaz razumiju gotovo svi te je relevantniji od možda nepreglednih dijagrama. Težinskim prosjekom korisnik je lišen razmišljanja o implementacijskim detaljima odnosa koncepata u prikazu njegova znanja, no on te odnose može vidjeti ako želi. Sigurnost točnog izračuna postiže se izračunom znanja iz granula, a ne prijašnjih rezultata koncepata. Kada bi se znanje računalo i na razini granula znanja i na razini koncepata, postoji mogućnost previše međuovisnosti i pogrešnog izračuna.

### **4.4.3. Ocjena za predmete**

Ocjene za predmete određuju se zbrajanjem ostvarenih bodova korisnika na svim ispitima te dijeljenjem tog broja s ukupnim mogućim brojem bodova. Ocjene su definirane sljedećim pragovima:

<50%: 1, 50% - 60%: 2, 60% - 70%: 3, 70% - 85%: 4, >85%: 5

## **4.5. Logika posluživanja pitanja**

Logika posluživanja pitanja vrlo je važan dio ovog sustava, a slijedi dva principa:

1. Korisniku treba poslužiti pitanja koja će mu predstavljati rješivi izazov, tj. ne treba mu nuditi pitanja koja je već dokazano točno riješio niti pitanja koja ne može odgovoriti temeljem trenutnog znanja

2. Svaka provjera treba sadržavati što veći broj različitih pitanja (osim ako drukčije ne postavi administrator) da ne bi naučio "kuharicu" i umjetno preskočio neke dijelove ili ako korisnik nešto ne zna, nema smisla ponuditi mu provjeru ispunjenu pitanjima te vrste.

### **4.5.1. Algoritam posluživanja pitanja**

Logika posluživanja pitanja unutar ovog sustava zapravo je vrlo jednostavna i prikazana je sljedećim algoritmom:

1. Dohvati sve koncepte unutar predmeta koje korisnik može odgovarati (zadovoljio je preduvjete ili preduvjete ne postoje).
2. Za svaki od koncepata uzmi dostupne granule, permutiraj listu svih granula.

3. Za svaku od granula uzmi po jedno pitanje prema 6. koraku.
4. Ponavljaj 3. korak dok ne ispuniš broj pitanja. Ako nema dovoljno pitanja, uzimaj po dva pitanja za svaku od granula.
5. Ponavljaj 4. korak dok se ne ispuni uvjet prekida povećavajući broj pitanja koji se uzima za jedan. Ako su sve kombinacije pitanja iskorištene, prekini traženje i ponudi manji broj pitanja unutar provjere.
6. Pitanja uzimaj po principu:
  - a) aproksimiraj prosječnu složenost znanja korisnika za neku granulu.
  - b) nađi pitanje s najbližom složenošću koje nije već uzeto za ovu provjeru.
7. Nakon završetka odabira pitanja, generiraj konkretnu provjeru s konkretnim pitanjima.

## **4.6. Konkretna pitanja u sustavu**

Pitanja unutar sustava pitanja su jednog zamišljenog predmeta - diskretne matematike, a njegovo je znanje dio dvaju predmeta na Fakultetu elektrotehnike i računarstva - Matematike 3R i Vjerojatnosti i statistike. Niti jedan od tih stvarnih predmeta nije u cijelosti sadržan unutar ovog sustava, on sadrži samo dijelove vezane uz diskretnu matematiku. Inspiracija za tekstualni opis preuzeo sam iz udžbenika [42] i [26], no tamošnja pitanja sadržavaju konkretne brojkke, a ovaj sustav sadrži formulama generalizirane verzije.

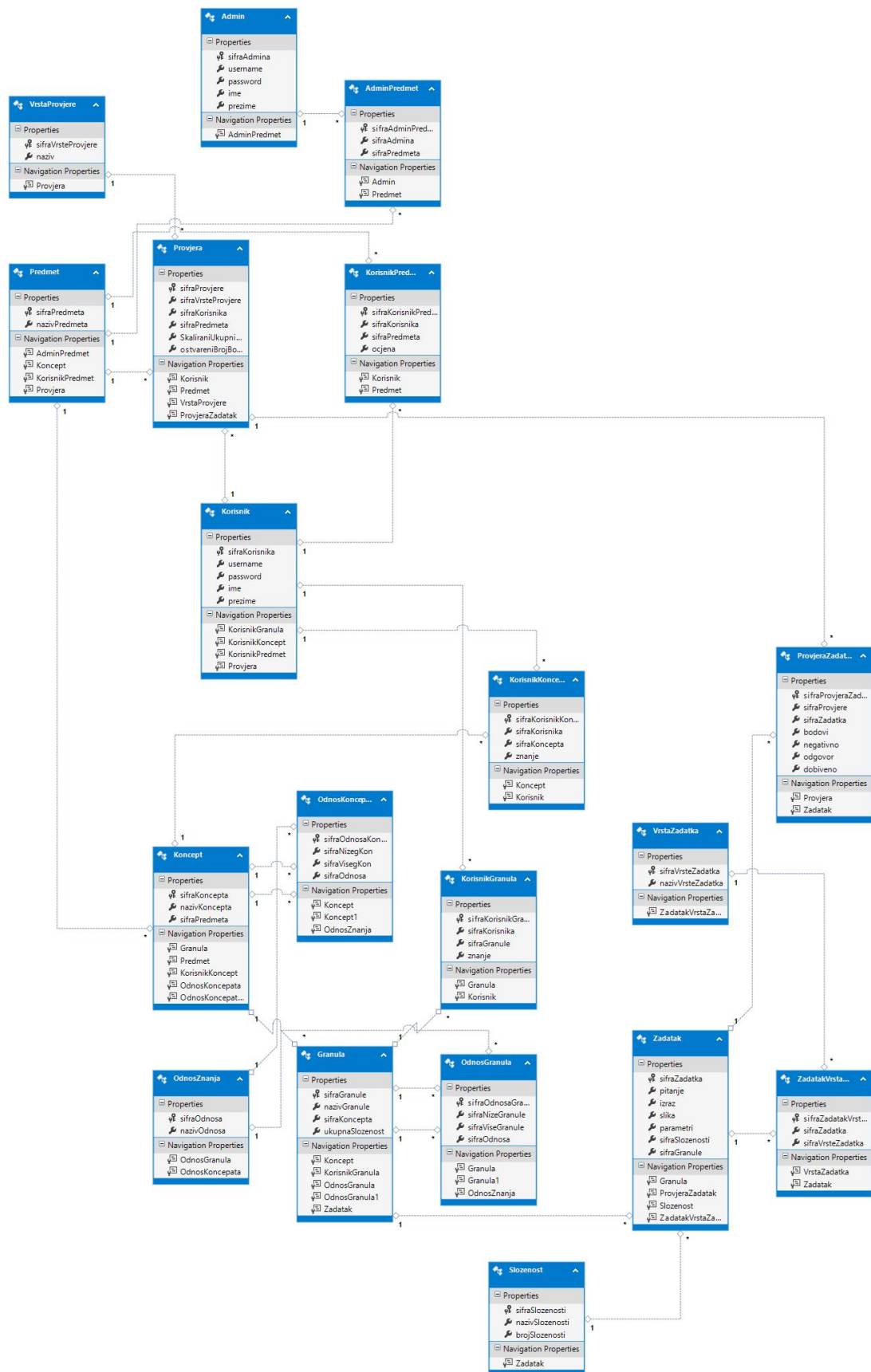
### **4.6.1. Test točnosti parametrizacije**

Generiranje parametara i računanje rezultata testirao sam na 10.000 različitih kombinacija parametara za pojedino pitanje te sam tu mogućnost i nadalje ostavio u sustavu. Tu mi se izuzetno korisnim pokazalo svojstvo mXparsera da svaku pogrešku u svom radu vraća kao NaN (nepostojeći broj) umjesto da baca grešku. Sve uočene pogreške riješene su te sustav generiranja parametara radi u potpunosti.



#### 4.6.2. Znanje u sustavu

- Predmet: Diskretna matematika
  - Koncept: Osnove kombinatorike
    - \* Granula znanja: Skupovi
    - \* Granula znanja: Produktno pravilo
      - Korištenje: Skupovi
    - \* Granula znanja: Permutacije
  - Koncept: Varijacije i kombinacije
    - Korištenje: Osnove kombinatorike
      - \* Granula znanja: Varijacije
        - Paralelno: Kombinacije
      - \* Granula znanja: Kombinacije
        - Paralelno: Varijacije
- Koncept: Napredna kombinatorika
  - Nadskup: Varijacije i kombinacije, Osnove kombinatorike
    - Granula znanja: Mješovita kombinatorika
      - Korištenje: Varijacije
      - Korištenje: Kombinacije
    - Granula znanja: Dirichletovo načelo
      - Analogno: Skupovi
- Koncept: Osnove vjerojatnosti
  - Preduvjet: Osnove kombinatorike
    - Granula znanja: Osnove vjerojatnosti



Slika 4.2: Dijagram entiteta u bazi

## 5. Zaključak

Edukacijski procesi značajan su i vrlo složen element ljudske civilizacije, a razvojem tehnologije i otklonom od manualnih djelatnosti, njihova važnost još više raste. Stoga je neizbježna potreba za unaprjeđenjem, olakšanjem i poboljšanjem procesa učenja, pogotovo u modernom svijetu u kojem je vrijeme pažnje čovjeka sve kraće.

Sustavi za inteligentno učenje (ITS) nameću se kao logično moguće rješenje novih izazova, a također istovremeno smanjuju povezanost fizičke pozicije ljudi i njihovog znanja. Internet je danas dostupan gotovo svuda i svima te su sustavi za inteligentno učenje konačno dobili platformu širenja.

No, da bi sustav za učenje bio kvalitetan, mora imati razrađenu podjelu znanja u manje granulacije i praćenje korisnikova znanja, pitanja u njemu moraju biti parametrizirana, mora biti (relativno) siguran za korištenje te izrađen kao skup tipičnih komponenti. Te komponente su domenski model, korisnički model, model učenja i korisničko sučelje. Jedini od modela koji može nedostajati je korisnički model, ako je njegova funkcionalnost rastavljena u domenski i model učenja. Kako bi taj sustav još bio i inteligentan, nužno je da ima iznimno dobro razrađeno praćenje korisnikova znanja te algoritme posluživanja pitanja. Pravi komercijalni inteligentni sustavi još mogu i pratiti tipkanje i facijalne reakcije korisnika te imati prilagođene upute za korisnika koji ih zatraži.

Iako sustavi za učenje na prvi pogled djeluju kao isključivo pitalice s odgovorima, to u praksi nije baš tako, dapače, svaki sustav koji ima edukacijsku svrhu, a nema oblik pitalice je korak unaprijed za proces učenja. Neki od takvih sustava su: edukativna igra - Urban Jungle, sustav koji vokalno razgovara s korisnikom - AutoTutor i Why2-Atlas koji komunikaciju izvodi tekstualno.

ITS je sigurno kvalitetan i zanimljiv koncept, no i on sadrži neke nedostatke, ponajprije inteligenciju ograničenu znanjem autora, nužnost kvalitetne interdisciplinarnе suradnje te potencijalno visoku cijenu izrade.

U posljednjem dijelu ovog završnog rada prikazao je dizajn jednog vlastitog

sustava za inteligentno učenje s parametriziranim pitanjima, reprezentacijom znanja različitim granulacijama, inteligentnim posluživanjem pitanja, ali i administratorskim nadzorom. Takva implementacija relativno je jednostavna u odnosu na komercijalne sustave, no ona je ITS.

Nakon poduzetog vremena zapostavljanja sustava za (inteligentno) učenje u odnosu na ostale primjene umjetne inteligencije, sa sigurnošću mogu tvrditi da oni sada konačno imaju veliku mogućnost razvoja te mnoge stvarne primjene.

# LITERATURA

- [1] <http://www.activemath.org/Software/Screenshots>. Preuzeto: 13. 6. 2016.
- [2] <https://upload.wikimedia.org/wikipedia/commons/c/cd/AutoTutor.png>. Preuzeto: 13. 6. 2016.
- [3] <http://userpages.umbc.edu/~rada/cv/pubs/hypertextbook/img/fig89.jpg>. Preuzeto: 13. 6. 2016.
- [4] <https://theethicalskeptic.files.wordpress.com/2014/11/tree-of-knowledge-obfuscation2.png>. Preuzeto: 13. 6. 2016.
- [5] <https://cdn3.vox-cdn.com/assets/4463827/2000px-MobileBroadbandInternetPenetrationWorldMap.svg.png>. Preuzeto: 13. 6. 2016.
- [6] <http://toolkit.pellinstitute.org/wp-content/uploads/2010/06/Reading-a-Logic-Model.png>. Preuzeto: 13. 6. 2016.
- [7] [http://moodle.srce.hr/2014-2015/theme/image.php/srce\\_access/theme/1465693202/logo](http://moodle.srce.hr/2014-2015/theme/image.php/srce_access/theme/1465693202/logo). Preuzeto: 13. 6. 2016.
- [8] <https://upload.wikimedia.org/wikipedia/commons/thumb/b/bb/MobileBroadbandInternetPenetrationWorldMap.svg/2000px-MobileBroadbandInternetPenetrationWorldMap.svg.png>. Preuzeto: 13. 6. 2016.
- [9] <https://moodle.org/logo/moodle-logo.png>. Preuzeto: 13. 6. 2016.
- [10] <http://www.moock.org/lectures/mvc/images/mvc-02.jpg>. Preuzeto: 13. 6. 2016.

- [11] <https://resources.instructure.com/courses/7/pages/sidney-pressey>. Preuzeto: 13. 6. 2016.
- [12] <http://www.strategies.com/blog/wp-content/uploads/2014/07/teamwork3.jpg>. Preuzeto: 13. 6. 2016.
- [13] <http://urbanjungle.dir.hr/galerija.php>. Preuzeto: 13. 6. 2016.
- [14] [activemath.org](http://www.activemath.org/Software). Activemath web - software. <http://www.activemath.org/Software>. Preuzeto: 13. 6. 2016.
- [15] Grupa autora. The architecture of why2-atlas: A coach for qualitative physics essay writing. *Lecture Notes in Computer Science*, 2363, 2002.
- [16] [biometric solutions.com](http://www.biometric-solutions.com/solutions/index.php?story=keystroke_dynamics). Keystroke dynamics. [http://www.biometric-solutions.com/solutions/index.php?story=keystroke\\_dynamics](http://www.biometric-solutions.com/solutions/index.php?story=keystroke_dynamics). Preuzeto: 14. 6. 2016.
- [17] Best T. Budd, S. 2008. [http://etec.ctlt.ubc.ca/510wiki/Educational\\_Benefits\\_and\\_Limitations](http://etec.ctlt.ubc.ca/510wiki/Educational_Benefits_and_Limitations). Preuzeto: 13. 6. 2016.
- [18] J. Chew. Blaise pascal. <https://math.berkeley.edu/~robin/Pascal/>. Preuzeto: 13. 6. 2016.
- [19] S. Colton. Knowledge representation. <http://www.doc.ic.ac.uk/~sgc/teaching/pre2012/v231/lecture4.html>, 2005.
- [20] B.J. Copeland. Dendral expert system. <http://www.britannica.com/technology/DENDRAL>. Preuzeto: 14. 6. 2016.
- [21] [coursera.org](https://www.coursera.org/about/). About | coursera. <https://www.coursera.org/about/>. Preuzeto: 13. 6. 2016.
- [22] H. Szolovits P. Davis, R. Shrobe. What is a knowledge representation? <http://groups.csail.mit.edu/medg/ftp/psz/k-rep.html>, 1993. Preuzeto: 13. 6. 2016.
- [23] [developer.apple.com / Mac Developer Library](https://developer.apple.com/library/mac/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html). Model-view-controller. <https://developer.apple.com/library/mac/documentation/General/Conceptual/DevPedia-CocoaCore/MVC.html>. Preuzeto: 13. 6. 2016.

- [24] duolingo.com. About us - duolingo. <https://www.duolingo.com/info>. Preuzeto: 13. 6. 2016.
- [25] eLearningNC.gov. Methods of elearning | elearningnc.gov. [http://www.elearningnc.gov/about\\_elearning/methods\\_of\\_elearning/](http://www.elearningnc.gov/about_elearning/methods_of_elearning/). Preuzeto: 13. 6. 2016.
- [26] N. Elezović. *VJEROJATNOST I STATISTIKA: Diskretna vjerojatnost*. Element, Zagreb, 2014.
- [27] ferko.fer.hr. Početna stranica - ferko. <http://ferko.fer.hr/ferko/About.action>. Preuzeto: 13. 6. 2016.
- [28] The Association for Educational Communications i Technology. 19.3 intelligent tutoring systems defined. <http://www.aect.org/edtech/ed1/19/19-03.html>, 2001. Preuzeto: 13. 6. 2016.
- [29] Glavinić V. Granić, A. User interface specification issues for computerized educational systems. *Journal of Computing and Information Technology*, 3, 2002.
- [30] Mokbel B. Hammer B. Pinkwart N. Gross, S. Learning feedback in intelligent tutoring systems. [https://cses.informatik.hu-berlin.de/pubs/2015/fit/learning\\_feedback\\_in\\_intelligent\\_tutoring\\_systems.pdf](https://cses.informatik.hu-berlin.de/pubs/2015/fit/learning_feedback_in_intelligent_tutoring_systems.pdf). Preuzeto: 13. 6. 2016.
- [31] A. Grubišić. Vrednovanje učinka inteligentnih sustava e-učenja. Magistarski rad, Sveučilište u zagrebu, Fakultet elektrotehnike i računarstva, 2006.
- [32] <http://ace.autotutor.org>. Autotutor. <http://ace.autotutor.org/IISAutotutor/index.html>. Preuzeto: 13. 6. 2016.
- [33] <http://faculty.coe.uh.edu>. Sidney pressey. <http://faculty.coe.uh.edu/smcneil/cuin6373/idhistory/pressey.html>. Preuzeto: 13. 6. 2016.
- [34] moodle.org. About moodle - moodledocs. [https://docs.moodle.org/31/en/About\\_Moodle](https://docs.moodle.org/31/en/About_Moodle). Preuzeto: 13. 6. 2016.
- [35] B. Park Woolf. Intelligent tutors: Past, present and future. [http://www.adlnet.gov/wp-content/uploads/2011/08/woolf\\_intelligent\\_systems\\_part1\\_iFest2011.pdf](http://www.adlnet.gov/wp-content/uploads/2011/08/woolf_intelligent_systems_part1_iFest2011.pdf), 2011. Preuzeto: 13. 6. 2016.

- [36] D. Peters. *Interface Design for Learning: Design Strategies for Learning Experiences*. New Riders, 2013. str. 21-22.
- [37] Oza T. H. Neve K. Phegade, S. G. Secure access of intelligent tutorial system on cloud using rsa algorithm. *IBMRD's Journal of Management and Research*, 3, 2014.
- [38] philosophyfaculty.ucsd.edu. G.w. leibniz: Texts and translations. <http://philosophyfaculty.ucsd.edu/faculty/rutherford/Leibniz/>. Preuzeto: 13. 6. 2016.
- [39] Shcherbinina O. Brusilovsky P. Sosnovsky, S. Web-based parameterized questions as a tool for learning, 2003.
- [40] van den Herik J. Spronck, P. Game artificial intelligence that adapts to the human player. [http://www.ercim.eu/publication/Ercim\\_News/enw57/spronck.html](http://www.ercim.eu/publication/Ercim_News/enw57/spronck.html). Preuzeto: 13. 6. 2016.
- [41] D. Stockley. E-learning definition (elearning, online training, online learning). <http://www.derekstockley.com.au/elearning-definition.html>. Preuzeto: 13. 6. 2016.
- [42] D. Žubrinić. *Uvod u diskretnu matematiku*. Element, Zagreb, 2014.
- [43] M. Urban-Lurain. Intelligent tutoring systems: An historic review in the context of the development of artificial intelligence and educational psychology. <http://www.cse.msu.edu/rgroups/cse101/ITS/its.htm>. Preuzeto: 13. 6. 2016.
- [44] urbanjungle.dir.hr/. Urban jungle. <http://urbanjungle.dir.hr/>. Preuzeto: 13. 6. 2016.
- [45] Z. Zhai. Intelligent tutoring system overview. <https://cseweb.ucsd.edu/~zzhai/blog/intelligent-tutoring-system-overview.html>, 2015.



## **Razvoj podatkovnog sloja i aplikacijske logike za potrebe sustava elektroničkog učenja**

### **Sažetak**

Ovim završnim radom opisao sam modernizaciju procesa učenja sustavom za inteligentno učenje. Rad sam započeo uvodom u edukacijske procese, zatim sam prikazao povijest ideje automatizacije učenja te suvremeni model sustava za inteligentno učenje (ITS). Opisao sam komponente ITS-a, očekivane karakteristike i algoritme uz kritički pogled na nedostatke ITS-a. Zatim sam opisao slojeva aplikacije koje je bilo potrebno implementirati te implementirano rješenja. Navedeni opis sadrži popis korištenih tehnologija, ER dijagram baze te mogućnosti koje sustav nudi korisniku i administratoru. Opis sam završio logikom posluživanja pitanja. U posljednjem poglavlju rada iznio sam 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**

In this paper I describe modernization of the learning processes using intelligent tutoring systems. I begin with introduction to educational processes, I then describe the history of automated learning and contemporary model of an intelligent tutoring system (ITS). Then I described the 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 user and an administrator. I follow with the description of the logic of serving questions. In the last chapter I presented a conclusion about aforementioned ideas.

**Keywords:** learning; intelligent learning; information system; ITS; knowledge; knowledge representation; parameterization; question serving; data layer; application logic