Analitika učenja u sustavima e-učenja

Bajić, Natko

Master's thesis / Diplomski rad

2017

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, University of Split, Faculty of science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:166:909825

Rights / Prava: Attribution-NonCommercial-NoDerivatives 4.0 International/Imenovanje-Nekomercijalno-Bez prerada 4.0 međunarodna

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Repository of Faculty of Science







SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

DIPLOMSKI RAD

ANALITIKA UČENJA U SUSTAVIMA E-UČENJA

Natko Bajić

Mentorica:

Doc. dr. sc. Ani Grubišić

Split, rujan 2017.

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO-MATEMATIČKI FAKULTET
Teslina 12, 21000 Split

IBAN: HR17 2330003110006883 1
SWIFT (BIC): SOGE HR22
MATIČNI BROJ: 3199622
OIB: 20858497843

Split, 10.03.2016.

Studij: FIZIKA-INFORMATIKA
Smjer: Nastavnički
Predmet: Sustavi e-učenja
Nastavnik: doc. dr. sc.Ani Grubišić

D I P L O M S K I Z A D A T A K

Kandidat: **NATKO BAJIĆ**

Zadatak: **ANALITIKA UČENJA U SUSTAVIMA E-UČENJA**
LEARNING ANALYTICS IN E-LEARNING SYSTEMS

Analitika učenja koristi podatke dobivene iz procesa učenja i poučavanja za poboljšanje istog. U sustavima e-učenja vodi se detaljna evidencija o napretku učenika, pa je logično primijeniti analitiku učenja upravo na te podatke.

Oblikovati programsko rješenje koje primjenjuje metode analitike učenja na podatke dobivene iz odabranog sustava e-učenja. Napraviti formalnu specifikaciju modela, te implementirati prototip s primjerenim testiranjem.

Zadatak uručen kandidatu: 30.11.2015.

Rok za predaju verzije rada završene 70%: 15.07.2017.

Rok za predaju diplomskog rada: 01.09.2017.

Mentor:

doc.dr.sc. Ani Grubišić

Temeljna dokumentacijska kartica

Diplomski rad

Sveučilište u Splitu
Prirodoslovno-matematički fakultet
Odjel za informatiku
Ruđera Boškovića 33, 21000 Split, Hrvatska

ANALITIKA UČENJA U SUSTAVIMA E-UČENJA

Natko Bajić

SAŽETAK

Ovaj diplomski rad započinje pregledom razvoja, ideja i nekih ostvarenih inteligentnih tutorskih sustava, nakon čega je detaljno opisana njihova danas široko prihvaćena arhitektura u vidu četiri modula. Nastavlja se pogledom na analitiku učenja iz perspektive vodećih istraživača tog područja, danas neizostavne pedagoške znanstvene discipline te se zaključuje o njenoj primjeni u inteligentnim tutorskim sustavima, ističući povezanost ova dva područja istraživanja. Predstavljene su osnovne zamisli inteligentnog tutorskog sustava AC-ware Tutor, koji posebnu pažnju pridaje modulu učenika i koristi se na splitskom PMF-u od 2012. g. Podaci o radu studenata iskorišteni su za prikaz podataka o napretku za vrijeme njihovog učenja. Za obradu se koristi programskom paketu RStudio, integriranom programskom okruženju za jezik R, najpopularnijeg alata u analitici učenja, čije su osnove također metodički objašnjene kroz konkretne primjere, a dane su i idejne smjernice za buduća istraživanja u analitici učenja.

Ključne riječi:
e-učenje, inteligentni tutorski sustav, analitika učenja, obrada podataka, prilagodljivi računalom oblikovani nastavni sadržaj, stereotip, jezik R, RStudio

Rad je pohranjen u knjižnici Prirodoslovno-matematičkog fakulteta, Sveučilišta u Splitu

Rad sadrži: 59 stranica, 4 grafička prikaza, 1 tablicu i 44 literaturna navoda. Izvornik je na hrvatskom jeziku.

Mentor:
Dr. sc. Ani Grubišić, docent Prirodoslovno-matematičkog fakulteta Sveučilišta u Splitu

Ocjenjivači:
Dr. sc. Ani Grubišić, docent Prirodoslovno-matematičkog fakulteta Sveučilišta u Splitu
Dr. sc. Branko Žitko, docent Prirodoslovno-matematičkog fakulteta Sveučilišta u Splitu
Mag. Divna Krpan, predavač Prirodoslovno-matematičkog fakulteta Sveučilišta u Splitu

Rad prihvaćen: rujan, 2017.

Basic documentation card

Thesis

University of Split
Faculty of Science
Department of computer science
Ruđera Boškovića 33, 21000 Split, Croatia

LEARNING ANALYTICS IN E-LEARNING SYSTEMS

Natko Bajić

ABSTRACT

This thesis starts with the overview of development, ideas and evaluation of intelligent tutoring systems, after which widely accepted architecture of its four modules is described in detail. Paper is continued with a synthesis of learning analytics from its prominent researcher's point of view, concluding with application in intelligent tutoring systems while highlighting synergy of these two academic disciplines. Basic ideas of intelligent tutoring system AC-ware Tutor are presented. This system is focused on student module and came into use in 2012 on Faculty of science, University of Split. Student data from this system were logged and now used for representation of their learning progress. Data was processed in RStudio, an integrated development environment for R language, the most popular tool in learning analytics. Its basics are also methodically explained through the examples. Some ideas for direction of future research on this system in learning analytics are also given.

Keywords:
e-learning, intelligent tutoring system, learning analytics, data mining, adaptive courseware, stereotype, R language, RStudio

Thesis deposited in the library of Faculty of science, University of Split.

Thesis consists of: 59 pages, 4 figures, 1 table and 44 references.
Original language: Croatian.

Supervisor: Ani Grubišić, Ph. D., docent of Faculty of Science, University of Split

Reviewers:
Ani Grubišić, Ph. D., docent of Faculty of Science, University of Split
Branko Žitko, Ph. D., docent of Faculty of Science, University of Split
Divna Krpan, M. D., lecturer of Faculty of Science, University of Split

Thesis accepted: September, 2017.

Sadržaj

Uvod 3

1. Inteligentni tutorski sustavi 4

1.1. Razvoj inteligentnih tutorskih sustava 4

1.2. Didaktički i psihologijski aspekt ITS-a 6

1.3. Građa inteligentnog tutorskog sustava 10

1.3.1. Modul stručnjaka 11

1.3.2. Modul učenika 15

1.3.3. Modul učitelja 18

1.3.4. Modul komunikacije 22

1.4. Budućnost inteligentnih tutorskih sustava 24

2. Analitika učenja 26

2.1. Razvoj i definicije 26

2.2. Procesi i metode u analitici učenja 30

2.3. Analitika učenja i inteligentni tutorski sustavi 34

3. Analitika učenja u inteligentnom tutorskom sustavu AC-ware Tutor 36

3.1. AC-ware Tutor 36

3.2. Jezik R 40

3.2.1. S i R 40

3.2.2. Istaknuti nedostaci 42

3.2.3. RStudio 42

3.3. Pisanje kôda u R-u 43

3.4. Primjena 44

Zaključak 50

Literatura 52

Sažetak 56

Summary 57

Skraćenice 58

Privitak 59

Uvod

Od pamtivijeka ljudi pridaju važnost prijenosu spoznaja među generacijama kako bi podigli razinu akumuliranog kolektivnog znanja i iskoristili ga za dobrobit svojih zajednica. Ideja škole (prema grč. *σχολή, scholē* - zabava) zbog toga se ostvarila u svim antičkim civilizacijama, otpočetka s dvostrukom zadaćom odgoja (poučavanje) i obrazovanja (podučavanje). Tijekom stoljeća velik dio ljudi nije imao privilegiju obrazovati se, a sama uspješnost prijenosa znanja izrazito je ovisila o tradiciji ustanova jer se, razumljivo, i umijeće vođenja odgojno-obrazovnog procesa uči i prenosi na buduće naraštaje iskustvom prijašnjih učitelja. Zajednice s kvalitetnim školskim sustavom su uspješne u prijenosu znanja i neminovno se ubrzano razvijaju, zbog čega je došlo do napredaka koji su, poput industrijske revolucije, imali odjeka na čitavo čovječanstvo. Globalizacijom se postiže lavinski efekt u širenju informacija, od mogućnosti podučavanja na daljinu prije stotinjak godina, preko pojave interneta, čime je gotovo čitavom čovječanstvu dostupno znanje prijašnjih generacija, do realizacije nastave bez uplitanja učitelja. S obzirom na bržu razmjenu informacija pedagogija je iznimno napredovala u dvadesetom stoljeću, čime su otvorene mogućnosti poboljšanja prijenosa znanja, zbog sistematizacije i individualiziranja nastavnog procesa. Danas je učenje i podučavanje bez korištenja računala teško zamisliti u formalnoj nastavi i van nje. Osim kao puki medij za pomoć pri učenju, najvažniji način korištenja računala u nastavi su sustavi e-učenja, bilo kao alati za podučavanje, rasterećivanje učitelja ili kao potpuna zamjena za klasičnu nastavu u sklopu inteligentnih tutorskih sustava. Računalo je danas neizostavno i kod prikupljanja, obrade i analize podataka o elementima nastave, a upravo na temelju zaključaka oblikuje se i prepravlja nastavni sadržaj.

U prvom poglavlju ovog rada je opisan razvoj inteligentnih tutorskih sustava te detaljno opisana njegova građa, dok je u drugom poglavlju dan pregled analitike učenja, danas neizostavne pedagoške znanstvene discipline, kao sinteza radova vodećih istraživača tog područja, s naglaskom na primjenu u inteligentnim tutorskim sustavima, povezujući obe cjeline i na toj razini. U trećem se poglavlju koristi teorijska podloga i zaključci prve dvije cjeline kako bi se analitika učenja primijenila na podatke sustava e-učenja AC-ware Tutor. Za obradu podataka koristi se RStudio, programski paket rastuće popularnosti u svim područjima istraživanja podataka.

1. Inteligentni tutorski sustavi

Inteligentni tutorski sustav (engl. *intelligent tutoring system*, ITS) je računalni sustav koji omogućuje neposredno učenje putem prilagođenih uputa, uz mogućnost primanja povratne informacije od učenika. Takvi sustavi se kontinuirano pokušavaju unaprijediti u skladu s najnovijim socijalno-kognitivnim teorijama učenja i drugim pedagoškim istraživanjima, a koriste se i u praksi kroz formalno obrazovanje, kao i u poslovnom okruženju. ITS su individualni jer podrazumijevaju proces učenja bez uplitanja konvencionalnog učitelja, čime se pokušava riješiti problem pretjerane ovisnosti učenika o učitelju kao istaknutog problema današnjice (Shore) i u konačnici vode k moderniziranju reformi cjelokupnog obrazovnog sustava.

1.1. Razvoj inteligentnih tutorskih sustava

Od pamtivijeka postoje ideje o inteligentnim neživim predmetima. Preko mitova do antičkih automatona, u svim kulturama svjedočimo o pokušajima oživljavanja neživog, pripisivanju ljudskih osobina i sličnog načina razmišljanja kao kod čovjeka. S druge strane, alati za podučavanje postoje jednako dugo, od špiljskih crteža kao upozorenja preko lekcija na glinenim pločicama u Mezopotamiji do antičkih knjižnica kao baza znanja. Industrijskom revolucijom javlja se znanstvena fantastika i ideje o robotici, od kojih se mnoge i tehnološki ostvaruju u novije vrijeme. Upravo koncept inteligentnih strojeva za podučavanje potječe iz tog doba. Početkom 19. stoljeća patentirani su prvi strojevi za podučavanje čitanja (1809.), pisanja (1810.), geografije (1842.) i aritmetike (1846.). (Lawson, 1973.)

Sidney Pressey na institutu savezne države Ohio je ranih dvadesetih godina 20. stoljeća razvio mehanički stroj za podučavanje bez pomoći učitelja koji je imao četiri tipke slične onima na pisaćem stroju i prozorčić s pitanjem koje ima četiri ponuđena odgovora. Kada korisnik pritisne tipku s točnim odgovorom prelazi se na sljedeće pitanje. Nadao se kako će ovaj stroj pomoći i u testiranju znanja učenika pa je 1928. patentirao nešto drukčiju izvedbu svog stroja koji dopušta samo jedan pokušaj po pitanju i na poleđini ima ugrađen brojač točnih odgovora. (Pressey, 1932.) Iako njegov stroj nije postigao priželjkivanu popularnost, inspirirao je stroj za učenje koji je B. F. Skinner patentirao 1958. g. i

prilagodio ga svom bihevioristi̇čkom pristupu učenja. S obzirom da je do 1936. g. u SAD-u patentirano već preko 600 strojeva za podučavanje, Skinner je smatrao kako se previše pažnje posvećuje tehnologiji bez prikladne pedagoške teorije koja bi je pratila (Lawson, 1973.). U stroju su se mogla pojavljivati pitanja raznih tipova, a učenik bi uvijek sam morao napisati odgovor. Kada bi povukao ručicu, pojavio bi se točan odgovor, što bi učeniku osim neposrednog pružanja povratne informacije dalo vremena da gradivo obrađuje svojim tempom na zabavan način.

Iako se nijedan od ovih strojeva danas ne može smatrati inteligentnim zbog jednostavnih mehaničkih funkcija, ipak su idejna osnova nastave pomoću računala (engl. *Computer-Assisted Instruction*, CAI). Nastavu pomoću računala može se definirati kao bilo kakvu metodu podučavanja koja uključuje bilo kakav oblik dijaloga računala i učenika. Razvojem računala ranih šezdesetih godina takvi su se koncepti sve više pojavljivali na američkim sveučilištima. Primjerice, na sveučilištu države Illinois učenici su po prvi put (sustav PLATO, 1959.) umreženi u računalnu učionicu gdje su mogli terminalom pristupati lekcijama određenog predmeta kojeg pohađaju dok su istovremeno slušali zvučni ili video zapis lekcije (Woolley, 1994.). Tih godina se na sveučilištu Stanford eksperimentiralo i s podučavanjem aritmetike i pravopisa u osnovnim školama preko teleprintera. S obzirom da računala nisu ubrzo postala dostupnija ni jeftinija, ovakvi programi su bili skupi za izradu i održavanje pa je entuzijazam oko CAI tehnologije splasnuo do početka sedamdesetih.

1.2. Didaktički i psihologijski aspekt ITS-a

Podučavanje jedan na jedan se smatra najučinkovitijim oblikom nastave. Oba sudionika, učenik i tutor-učitelj, ovim procesom stječu nova znanja i vještine. Pri tome se ističe kako u primjeni ovog modela tutor ne mora nužno biti stručan u vještinama podučavanja koliko bi trebao biti za nastavu u klasičnoj učionici, iako mora jednako dobro vladati područnim znanjem. Ovaj model podučavanja osnovna je ideja nastave pomoću računala (dalje u tekstu CAI), gdje je živi tutor zamijenjen računalom. Visoku učinkovitost modela jedan na jedan, među ostalim, ističe Bloom 1984., koji ustvrđuje izrazitu razliku između nasumično odabranih učenika koji su gradivo obrađivali po klasičnom modelu i modelu jedan na jedan (tutorska grupa). Prosječan student tutorske grupe je bolji od čak 98% studenata klasične grupe, dok je razlika prosječnih učenika po grupama dvije standardne devijacije. Iako bi se ovim modelom većini učenika mogla omogućiti vrlo visoka razina znanja, Bloom navodi kako je ono preskupo u većini društava za široku primjenu u podučavanju, što naziva "2 sigma problemom".

Usporedno s CAI šezdesetih godina, razvijaju se nove pedagoške teorije, među kojima i programirana nastava, kao prvi empirijski oblik nastave koji konceptualno približava znanost i obrazovanje. Programirana nastava svoje korijene vuče još iz antičke Grčke, kad je Sokrat svoje učenike razgovorom od činjenice preko činjenice vodio do obrazovnog cilja, kako zapisuje Platon u svom djelu "*Menon*". Također, spomenuti Pressyev stroj za učenje (verzija prije stroja za testiranje) u ideji je bio jedan od ranih oblika ovakve nastave. Ipak, pojam programirane nastave po prvi puta koristi Skinner 1954. u radu "*The Science of Learning and the Art of Teaching*" gdje ga razrađuje praktičnim uputama. Ideja programirane nastave je strukturirati gradivo u uređene cjeline manjeg opsega (skup nastavnih jedinica, u lit. i koraci, okviri itd.), koje se obrađuju jedna za drugom, bez mogućnosti preskakanja. Prethodnu cjelinu je potrebno usvojiti prije prelaska na iduću, što se provjerava testiranjem. Učenici samostalno i svojom brzinom obrađuju programirani nastavni sadržaj. Prema Stankovu (2010.), postoje dvije osnovne vrste programirane nastave.

Linearna programirana nastava (engl. *linear programmed instruction*, Skinner, 1954.) zamišljena je na način da se nakon obrađene cjeline i pozitivnog ishoda testa učeniku kao potkrjepljenje pruži na obradu iduća cjelina.

Razgranata programirana nastava (engl. *branching/intrinsic programmed instruction*, Crowder, 1959.) pak nakon testiranja usmjerava učenika na različite cjeline. U slučaju ispunjavanja očekivanja testiranja postupa se kao u linearnoj programiranoj nastavi, a s obzirom na greške odlučuje se kojim će se dopunskim cjelinama pristupiti radi otklanjanja nedostatka razumijevanja gradiva.

Pojavom mikroprocesora i osobnih računala dolazi do napretka i u kvantitativnom i kvalitativnom smislu nastave pomoću računala. Do početka osamdesetih većina škola i sveučilišnih knjižnica u SAD-u je bila opremljena arhivama lekcija na računalu, a po modelu programirane nastave javljaju se i ideje o inteligentnim CAI sustavima (engl. akronim ICAI) za koje se danas uvriježio izraz inteligentni tutorski sustavi - ITS. Dok su klasični oblici CAI sve više težili boljem povezivanju s usko specijaliziranim sadržajem koji se obrađuje, ITS su se fokusirali na generaliziranje znanja o problemu na nespecifične načine. (Larkin i Chabay, 1992.). Hartley i Sleeman u svom radu "*Towards More Intelligent Teaching Systems*" (1973.) navode kako dostignuta razina razvoja računarstva omogućuje brzo računanje i dovoljan kapacitet pohrane podataka za razvoj ovakvih sustava. Skiciraju kako bi budući ITS trebao izgledati u smislu upravljanja učenja, pohranjivanja i organiziranja učeničkih odgovora u svrhu prilagođavanja nastave individualnim potrebama. S obzirom na to definiraju pojam inteligencija podučavanja (engl. *teaching intelligence*).

No, kako definirati inteligenciju? Povijesno gledano, pridjev inteligentan se pripisivao svim sofisticiranijim uređajima i strojevima svog vremena. Još u devetnaestom stoljeću strojevi su mogli obavljati sve matematičke i osnovne logičke operacije, a spomenuti Presseyev stroj za testiranje je autor također prozivao inteligentnim. Značajan razvoj arhitekture računala za vrijeme Drugog svjetskog rata otvorio je put k interdisciplinarnim raspravama o stvaranju "umjetnog mozga". S obzirom da su se sam naziv "umjetna inteligencija" kao i ciljevi u skoroj i daljoj budućnosti raspravili 1956. g. šestotjednom konferencijom na sveučilištu Darmouth, u literaturi se taj skup učestalo uzima za osnutak ovog područja kao akademske discipline. Klasična definicija inteligencije u psihologiji je snalaženje u novonastalim situacijama. Stoga je glavni i osnovni cilj ITS-a mogućnost dinamične prilagodbe u izlaganju nastavnog sadržaja na, po mogućnosti, optimalan način. Sustav je trebao razaznati ne samo točne i netočne odgovore, već i način na koji učenik pogrešno odgovori i s obzirom na dijagnozu odrediti kako pomoći pri rješavanju individualiziranjem stila podučavanja (Shute, V. J., Psotka, J., 1994.). Također, pristupom

vlastitoj bazi znanja, ITS bi induktivnim i deduktivnim procesima morao moći predstaviti, pohraniti i dohvatiti znanje i biti u mogućnosti naučiti nove načine kako odgovoriti na pitanja učenika (Stankov, 2010.). Istraživanja na području umjetne inteligencije i kognitivne psihologije temelj su principa ITS. Psiholozi su razmatrali načine kako bi računalo moglo rješavati zadaće i djelovati na tako "inteligentan" način. Ideja dijagnostike i dopunjavanja pomoći (engl. *remediation*), začeta razgranatom programiranom nastavom se i danas koristi pri njihovom programiranju.

Istraživanja ovog područja nužno uključuju stručnjake iz barem tri znanstvene discipline - informatike, psihologije i naravno pedagogije. Nwana u svom opširnom pregledu iz 1990. g. ITS jednostavno i slikovito opisuje presjekom skupova djelatnosti ovih triju znanosti. Upravo zbog naglašene interdisciplinarnosti do danas su se razvili brojni sustavi ovog tipa, ne samo iz praktičnih razloga rasterećivanja i pomoći u nastavi, već i zbog ljudskog istraživačkog duha razumijevanja našeg načina razmišljanja na svim razinama.

Projekti na području ITS-a se razvijaju postupno iz CAI sustava. J. R. Carbonell 1970. g. razvija SCHOLAR, kao prvi (prema Rosić, Glavinić, Stankov, 2006.) ITS za podučavanje južnoameričke geografije putem semantičkih mreža, zbog čega je kasnije prozvan "ocem inteligentnih sustava za učenje i podučavanje". Burton i Brown 1975. g. razvijaju sustav SOPHIE za podučavanje u elektronici, o kojemu će još biti govora, a 1982. g. i sustav WEST - How the West Was Won kao računalnu igru koja može služiti za podučavanje aritmetičkih operacija. Iste godine Brown, Burton i Larkin razvijaju sustav Buggy koji detaljno objašnjava učeniku razloge pogrešaka u zadacima (također iz aritmetike). Jedan od najvažnijih koraka u razvoju inteligentnih tutorskih sustava napravljen je projektom LISPITS (1983.), u kojem se implementacijom načela kognitivne teorije ACT (*Assumptions and related principles for a Computer-implemented Tutor*) na praktičan način učio programski jezik LISP. Sustav je pomagao učenicima s domaćim zadaćama, pružajući konstruktivne upute kada bi otkrio pogrešku u kodu. Istraživanja učinkovitosti ovog sustava (Anderson, Boyle i Reiser, 1985.), nisu pokazala statistički značajne razlike znanja u završnom testu triju skupina ispitanika koji nisu imali iskustva s programiranjem. Prvu skupinu je podučavao učitelj, drugu LISPITS, a treća je sama učila LISP. Ipak, postojale su značajne razlike u vremenu potrebnom za savladavanje gradiva (po grupama: 12 sati, 15 sati te 28 sati za samostalnu skupinu). Drugo istraživanje (Anderson i Reiser, 1985.), na učenicima koji su već učili programiranje u Pascal-u, vršilo se na dvije skupine s jednakim zadaćama. Skupina koja je učila preko LISPITS-a je završni test riješila u

prosjeku 30% brže s 43% boljim rezultatom od skupine koja je sama učila LISP (Corbett, Anderson, 1989.).

Od sličnih ITS sustava koji su se tih godina razvijali može se spomenuti još i TUTOR, općeniti alat za podučavanje kojeg je razvila Logica 1984. g. i PARNASSUS, razvijen na sveučilištu Carnegie Mellon 1989. g. za upute u programiranju. U Hrvatskoj je značajan TEx-Sys (Slavomir Stankov) u on-site okruženju razvijan od 1992. g. (Stankov, 2010.), model za sustav u proizvoljno odabranom područnom znanju, na kojemu su se kasnije razvijali vezani sustavi prilagođeni akademskom obrazovanju.

Još i danas postoje brojni kritičari ITS-a, smatrajući kako je neočekivano usporen razvoj umjetne inteligencije pokazatelj kako ovakvi sustavi nikad neće moći zamijeniti živog učitelja (npr. Ridgeway koji jedan od svojih članaka imenuje "*Naravno da je ICAI nemoguć...još gore, sama ideja je buntovna*") (prema Nwana, 1990.). Ipak, njihova široka upotreba i značajan napredak od kraja osamdesetih godina je siguran pokazatelj da su već ne samo korisni alati nego i često nezamjenjivi u raznolikoj, konstruktivističkoj nastavi.

Kao zaključak ovog odjeljka i uvod u detaljnije razmatranje građe ITS-a potrebno je ukratko poopćiti navedene probleme s kojima su se susretali tvorci i korisnici sustava CAI, kako bi se dale smjernice za oblikovanje inteligentnih sustava. Uvažavajući pedagoške teorije Anderson i sur. (1987.) su odredili osam načela za inteligentne tutorske sustave, koje su dalje razradili Corbett i sur. (1997.) posebno ističući sveobuhvatno načelo:

Načelo 0: Inteligentni tutorski sustav treba omogućiti učeniku da uspješno razradi put k rješenju problema.

Ostalih 8 specifičnijih načela su:

1. Predstaviti učenikove sposobnosti kao razine učinkovitosti (engl. *production set*).
2. Raspraviti strukturu ciljeva u sklopu rješenja problema.
3. Pružiti upute za rješenje problema.
4. Poticati apstraktno razumijevanje znanja u ovom procesu.
5. Minimalno opteretiti "radnu memoriju" učenika.
6. Pružiti neposrednu povratnu informaciju o greškama.
7. Prilagoditi veličinu uputa učenju.
8. Pripomoći uzastopnim aproksimacijama do željenog konačnog ishoda učenja.

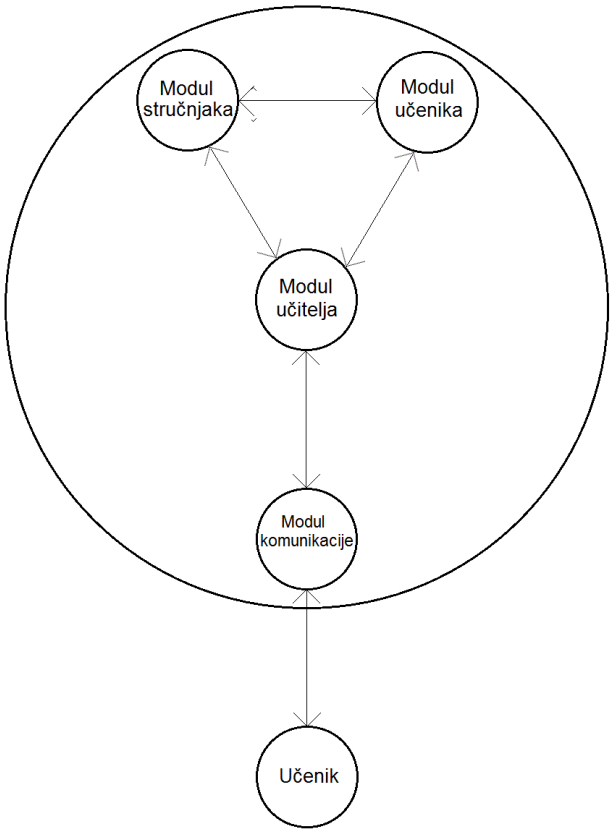
1.3. Građa inteligentnog tutorskog sustava

S obzirom na iznesena razmatranja može se zaključiti kako je sustav za inteligentno podučavanje zasnovan na algoritmu za odlučivanje s obzirom na podatke o području i ciljevima podučavanja. Oblikujući nastavu poput živog (i dobrog) učitelja, pored korištenja inteligentnog dijaloga, ITS treba prepoznati i uvažavati individualne razlike učenika za najbolji ishod učenja.

Implementacijom takvog inteligentnog odlučivanja građa ITS-a se može sintetizirati u četiri osnovne komponente. Većina stručnjaka ovog područja (primjera radi Nwana, 1990., Stankov, 2010. te Nkambou i sur., 2010.), smatra kako je ovakva arhitektura (shema na slici 1.) optimalna za većinu zadaća ovakvih sustava, a zasniva se prema idejnoj strukturi koju su predložili Hartley i Sleeman u prethodnom odjeljku spomenutom radu još 1973. godine.

1. znanje (modul) stručnjaka oblikuje područno znanje potrebno za podučavanje. (engl. *expert/domain module*)
2. znanje (modul) učenika se stječe tijekom procesa učenja i podučavanja oblikujući računalnu kopiju učenikovih sposobnosti i znanja. (engl. *the student module*)
3. znanje (modul) učitelja didaktički oblikuje nastavne sadržaje u suradnji s prva dva modula. (engl. *tutoring module*)
4. komunikacijski modul sustava se orijentira na interakciju stvarnog učenika i modula učitelja (engl. *user interface module*)

U sljedećim odjeljcima detaljno će se razmotriti svaki od ovih modula. Treba napomenuti kako postoje i u teoriji i implementaciji nešto drugačiji načini strukturiranja ITS-a, iako su načela i ideje iste. Neki se manje razlikuju, kao primjerice ranije spomenuta Andersonova ACT arhitektura korištena u LISPITS-u, koja funkcije modula učenika iz gornjeg prikaza dijeli s modulom stručnjaka i posebnom priručnom listom pogrešaka (engl. *bug catalogue*). Veće razlike očituju se primjerice u prstenastoj arhitekturi (O'Shea i sur., 1984.) koja u osnovi ima dvije vrste modula učenika (za pohranu podataka i za modeliranje učenika) i tri modula učitelja od kojih samo administrativni modul komunicira direktno s vanjskim svijetom, odnosno stvarnim učenikom. Nešto ranije ideje iste grupe istraživača (1982.) uvode smjernice za stalno samopoboljšanje ITS-a iterativnim postupkom, što je još jedna značajka ovih sustava, o čemu će također podrobnije biti govora u sljedećim odjeljcima.



Slika 1. *Građa inteligentnog tutorskog sustava, skica prema Nwana (1990.).*

1.3.1. Modul stručnjaka

Prva komponenta je u osnovi baza znanja određene problematike. Znanje mora biti predstavljeno na razini koja je prikladna za određeni zadatak podučavanja. To uključuje i problemu specifične ishode (tražene načine razmišljanja i kognitivne sposobnosti) i analizu zadatka, odnosno cjeline s kojima se upućuje na cjelokupnu strukturu i komponente nastavnog sadržaja. U idealnom slučaju to mora biti određeno s takvom preciznošću da računalni program može generirati kako zadatke tako i rješenja u okviru nastavnih sadržaja. Samo stjecanje znanja u sustavu, osim prikupljanja informacija, uključuje odlučivanje o relevantnim podacima i uspostavljanje odnosa između njih, strukturirajući ih u oblikovane cjeline. Carrico i sur. (1989.) predstavili su ovaj proces kao iteracijski u tri koraka: inicijalizacija (prikupljanje početnih podataka), legaliziranje (usporedba rezultata s očekivanima) i pročišćavanje stečenog znanja kako bi se obogatila početna baza znanja. Proces se ponavlja radi stalnog usavršavanja. Nadalje, prikaz znanja može biti moguć tek kad se znanje formalizira, tj. izgradi model područnog znanja. To je znanje namijenjeno za

podučavanje, a prema Stankovu može se predstavljati samo ako zadovoljava dva nužna uvjeta: a) zbog učenja načina rješavanja problema "moraju se izravno modelirati tehnike koje bi stručnjak primijenio u rješavanju problema s tog područja", čak i ako je zbog toga sama baza znanja nepotpunija, b) sustav pri rješavanju problema mora uzeti u obzir sve vjerojatne strategije i tehnike "čak i one koje su pogrešne tako da tutor može prepoznati što učenik pokušava učiniti."

Anderson (1988.) je razdvojio tri pristupa za strukturiranje znanja modula stručnjaka ITS. U modelu crne kutije (engl. *black box model*) mehanizam djelovanja nije pristupačan ili se ne koristi pri zadavanju uputa, ali se ulazno-izlazno ponašanje odvija valjano za određeno područje znanja. Anderson model slikovito opisuje igrom šaha s računalom koje, za razliku od ljudskog suparnika, idući potez bira između milijuna mogućih sekvenci poteza. Računalo može pokazati koji je potez najpovoljniji, ali ne zna objasniti zašto. Upravo na tom principu je na spomenutom sustavu PLATO razvijena igra WEST (Burton i Brown, 1982.) za vježbanje aritmetike. Još raniji primjer stručnjaka crne kutije je prva verzija sustava SOPHIE (također Brown i Burton, 1975.), koji je služio za podučavanje traženja grešaka u neispravnim elektroničkim sklopovima, odnosno potrebu i valjanost raznih mjerenja koja su obavljali učenici. SPICE II, tada dvije godine star elektronički simulator opće namjene na kojem je ovaj sustav zasnovan, koristi matematičke funkcije i rješava sustave jednadžbi. Stoga SOPHIE, iako nema čovjeku slično znanje o elektroničkim sklopovima pa ne može objasniti postupak ipak može procijeniti koje je mjerenje opravdano i je li valjano. Brown i Burton su ovaj sustav pokušali poboljšati da više podsjeća na tutora orijentiranog problemu (engl. *issue-based tutoring*). Ideja je prepoznati razlike u uzorcima ponašanja učenika i tutora te uskladiti instrukcije. Ideju su prvo implementirali igrom WEST, koja se prekida ako je potez nepovoljan kraćim objašnjenjem. Koncept inteligentnog tutorstva se zasniva upravo na kritičkom mišljenju i objašnjavanju, umjesto jednostavnih točno, netočno ili "uradi to i to" povratnih informacija. Ova ideja nije ograničena samo na model crne kutije, već je univerzalna u ITS. Ipak, u kasnijim verzijama sustava SOPHIE, Brown i Burton su postali svjesni mnogih problema u podučavanju koji se ne mogu riješiti ovako jednostavnim sustavima prepoznavanja greške. Anderson to naziva površinskim pristupom problemu (engl. *surface-level recognizers*) i ističe kako je tutoru potreban dublji pristup strukturi znanja za prikladna objašnjenja miskoncepcija i veće motivacije učenika.

Drugi oblik modula stručnjaka se oprečno naziva model prozirne kutije (engl. *glass box model*) jer tutor prikladno može objasniti svoj način zaključivanja. Ovaj model stvaraju, prema tradicionalnom načinu podučavanja, stručnjaci područja znanja ontološki[1] surađujući s inženjerima znanja (engl. *knowledge-engineer*), čime se teži znanju i razmišljanju sličnom ljudskom tutoru. Stručnjaci znanja prozirne kutije su zbog svoje složenosti vrlo obimni i vremenski zahtjevni jer se proces testiranja i unaprjeđivanja iterativno ponavlja. Taj proces sadrži identifikaciju problemskog područja i ciljeva, formaliziranja ključnih područnih koncepata i formuliranja sustava za implementaciju znanja. Istaknuti primjer ovakvog stručnjaka je GUIDON (Clancy, 1982.), zasnovan na modulu stručnjaka MYCIN (Shortliffe, 1976.), koji je služio za dijagnostiku bakterijskih infekcija. Sadržava 450 if-then pravila koji obrađuju djeliće informacija i logički zaključuju dijagnozu. GUIDON služi za poduku studenata medicine u tom području, a u komunikaciji sa studentima koristi dijalog podijeljene inicijative (Stankov), odnosno kako ih Anderson naziva T-pravila (engl. *T-rules*), gdje T označava i tutorstvo i grananje odluke. Ta se pravila definiraju na razlikama u ponašanju modula stručnjaka i učenika, ali i na procesu zaključivanja modula stručnjaka. Dijalog podijeljene inicijative se može odnositi i na unutarnju strukturu modula stručnjaka, kao na pravila i ciljeve podučavanja. Anderson utvrđuje kako se ovim crna kutija otvara i postaje prozirna. Ipak, bez obzira koliko je ovakva baza znanja korisna i obimna, podučavanje zahtijeva još dublju razinu strukturiranja znanja nego rješavanje problema. Clancey (1987.) zaključuje kako ljudski tutor može objasniti problem analogijama, različitim gledištima i razdvojiti razine objašnjavanja koje MYCIN ne može.

Iz zaključaka opisanih dvaju pristupa organizacije modula stručnjaka nastala je ideja o kognitivnom (spoznajnom) modelu koji upravo pokušava oponašati razinu apstrakcije kojim čovjek koristi svoje znanje. Iako ovaj model, u idealiziranoj situaciji, pruža modul stručnjaka koji najlakše i najkvalitetnije surađuje s učenikom, on mora biti daleko složeniji i razrađeniji nego spomenuta prozirna kutija. Anderson i Stankov navode ukupno tri glavna problema strukturiranja. Razvoj kognitivnog modela zahtijeva iznimno puno vremena. Na njemu mora raditi interdisciplinarni tim stručnjaka s obzirom da se bazira na kognitivnom pristupu psihologiji obrazovanja, koja je nasreću, u zadnjih tridesetak godina iznimno

[1]Ontologija u užem (informatičkom) smislu predstavlja formaliziranja imena, definicija, svojstava koncepata nekog područja i klasificiranja njihovih međusobnih odnosa. Slična definicija u npr. Gruber, T. R. "*A Translation Approach to Portable Ontology Specifications*", 1993.

napredovala, dijelom zasigurno i zbog rada na tutorskim sustavima. Nadalje, zbog složenosti, ovaj model je za pokretanje izrazito računalno skup, iako se dimenzija tog problema iz dana u dan umanjuje povećanjem računalne moći. Taj problem se dijelom preklapa i s pitanjem koliko detaljno razraditi kognitivni model kao kopiju psihologijskog. Primjerice, kratkoročno memorijsko pretraživanje koje se može simulirati nije bitno za podučavanje. Anderson smatra kako tutorski sustav ovisi o kognitivnim pretpostavkama na algoritamskoj, a ne implementacijskoj razini, odnosno orijentira se na mentalne procese, a ne problematiku strukture neurona. Analogiju u informatičkom svijetu navodi kao orijentaciju na programske jezike visoke razine, a ne arhitekturu računala. Kognitivni model mora uključivati sve oblike znanja za uspješno tutorsko podučavanje. Prema zaključcima niza istraživača ovog područja[2] razlikuju se tri tipa znanja. Deklarativno znanje je statičko i sastoji se od skupa činjenica (objekti i odnosi među njima), dok se proceduralno znanje orijentira na postupak kako izvršiti određeni zadatak i primjereno je uže specijaliziranim područjima. Iako su kasnih sedamdesetih i osamdesetih razvijani tutorski sustavi bazirani strogo na jednom od ova dva tipa znanja (deklarativni su se pokazali uspješnijim od čisto proceduralnih) primijećeni nedostaci su usmjereni k ideji da jedan oblik znanja ne isključuje drugi. Treće, kauzalno znanje je usmjereno na modeliranje kvalitativnog rasuđivanja i predviđanja ponašanja. Prema Andersonu je vrlo teško formalizirati sveobuhvatnu metodologiju ovakvih procesa, u vrijeme pisanja njegovog članka, 1988. g. vidi modele kvalitativnog rasuđivanja u povojima, ali ih smatra nezamjenjivo važnim za budućnost izgradnje modula stručnjaka. Prema Stankovu, bez obzira na njihov napredak posljednjih godina, još nisu dosegli učinkovitost čisto deklarativnih. Ciklus neumjerene pozornosti, koji će biti objašnjen na samom kraju ove cjeline, može objasniti zašto se i u vrijeme pisanja ovog rada inzistira na deklarativnom modelu.

[2]Burton, 1982.;Slemman, 1982.; Goldstain, 1982., Angelides i Tong, 1994.; Angelides i Garcia, 1993.

1.3.2. Modul učenika

Način prikaza znanja je ključan preduvjet za drugi od četiri modula ITS-a - modula učenika. Učenik predstavlja središnju komponentu ITS-a s obzirom da se podaci o njegovom postignuću u učenju neposredno mjere i koriste za oblikovanje nastavnog sadržaja. Modul učenika se odnosi upravo na te podatke. Klasični CAI sustavi, kako je već spomenuto, nisu bili uspješni u podučavanju kao što bi bio živi učitelj koji nastoji prilagoditi podučavanje učenikovim sposobnostima, predznanju, stilu učenja itd. Prilagodba učeniku je mogla biti, u najboljem slučaju, na jednoj elementarnoj razini pa su se takvi sustavi i koristili uglavnom samo za ocjenjivanje manjih upitnika. Proces učenja prema objektivistu Karlu Popperu se odvija prema teoriji triju svjetova: realnog, mentalnog i objektivnosti (sadržaj znanstvenih ideja, u našem slučaju baza znanja). Ove ideje su korisne u metodici jer se učenje može lako vizualizirati kao proces proučavanja prvog svijeta (interakcije učenika sa okolinom) uz pomoć već obređene baze znanja (trećeg svijeta), čime se postupno razvija znanje pojedinca (drugi svijet), a u idealnom slučaju učenik postaje stručnjak i doprinosi približavanju realnosti i objektivnosti. Upravo zbog toga struktura ITS-a mora sadržavati i modul učenika i modul stručnjaka kao dvije odvojene baze znanja. Modul učenika se može zamisliti kao modul stručnjaka sa zbirkom razlika. Dvije su osnovne vrste razlika. Pojmovi i dijelovi znanja koji nedostaju (engl. *missing conception*) i krivo shvaćeni pojmovi odnosno uvriježene miskoncepcije (engl. *misconceptions*), o kojima je već bilo govora. Stručnjak sadrži pojmove koje nedostaju, a učenik ih treba steći, dok se miskoncepcija ne nalazi u bazi znanja, a modul učenika je sadrži. Ove dvije vrste razlika su osnova razlikovanja trenutačnog i poželjnog znanja u modeliranju učenika. Pri razvoju prvih inteligentnih sustava, kako je također napomenuto u uvodnom razmatranju, problemima se pokušavalo pristupiti na načine neuobičajene u klasičnoj nastavi. Carbonell i Wexler su 1970. g. nezavisno gradili modele učenika za vrijeme podučavanja putem semantičkih mreža. Pristupili su problemu oprečno, dok je Wexler zasnivao modeliranje počevši od pretpostavke da učenik ne zna ništa o područnom znanju, Carbonell (model SCHOLAR) smatra kako su slučajni odgovori u tako ranoj fazi nepouzdani pa bi prikladnije bilo izgraditi model učenika na izmišljenom, idealiziranom učeniku koji je u osnovi baza znanja na potrebnoj razini pa ga tada s obzirom na odgovore mijenjati.

Stankov navodi kako je najpotpuniji prikaz modeliranja učenika predstavio VanLehn (1988.), s obzirom da su dosad izgrađeni sustavi s velikim konceptualnim razlikama. Klasifikacija modela učenika se može promatrati u tri dimenzije: ulaznim podacima o trenutačnom znanju učenika (engl. *bandwidth*), ciljanim tipovima znanja (engl. *target knowledge type*) i razlikama znanja učenika stručnjaka (engl. *differences between student and expert*). Prva dimenzija teži k zaključivanju učenikovih kognitivnih procesa tijekom podučavanja. Postavljajući dovoljno prikladnih pitanja uz detaljno izrađene mehanizme prepoznavanja (npr. analizu vremena potrebnog za odgovore) sustav može približno zaključiti o mentalnom stanju učenika. To je ujedno i prva, najviša razina ove dimenzije (engl. *approximate mental states*). Na nižim razinama postoji niz međustanja (engl. *intermediate states*) koja vode do konačnog stanja (engl. *final states*) kao najniže razine koju svaki ITS može prepoznati. Model učenika u nekim sustavima može biti izgrađen tako da je sposoban dohvatiti i međustanja, čemu se i teži u razvoju ITS-a, a time se upotpunjuje slika o načinu razmišljanja učenika. S obzirom da se od modela učenika očekuje isto ponašanje kao i od učenika, model se može koristiti i za predviđanje odgovora. Slično kao i kod oblikovanja područnog znanja, u dimenziji ciljanog znanja u modulu učenika može se razlikovati deklarativni i proceduralni pristup. Deklarativno (činjenično) ciljano znanje je najniža razina ove dimenzije (korišten npr. za prikaz znanja iz zemljopisa u spomenutom sustavu SCHOLAR), gdje je interpretacija rješavanja problema deklarativna. Kod algebre, primjera radi, potreban je proceduralni (postupkovni) pristup. Postoje dvije razine proceduralnog pristupa u ovoj dimenziji. Hijerarhijski prikaz sadrži, ekvivalentno prvoj dimenziji, međuciljeve u stanju rješavanja problema. Ravninski proceduralni (engl. *flat procedural*) prikaz sadrži samo konačno rješenje postavljenog problema. Za proces rješavanja problema u modelu učenika koriste se prevoditelji (engl. *interpreter*). Prevoditelj za proceduralno znanje donosi odluke na temelju lokalnog znanja. Stankov ga vrlo slikovito opisuje poput čovječuljka koje osvijetli dio znanja iz baze koje je potrebno za trenutačno stanje problema. U slučaju hijerarhijskog prikaza ide korak po korak i odlučuje kamo uperiti svjetiljku. Prevoditelj za deklarativno znanje mora neprestano pretraživati bazu znanja i s obzirom na to zaključiti odgovor, zbog čega je modeliranje učenika složenije. Konačno, treća dimenzija modula učenika je razlika znanja od eksperta. Spomenuti tipovi razlika u znanju također stvaraju tri razine ove dimenzije. Najjednostavniji model razlike je prekrivanje (engl. *overlaying model*), u kojem sustav ima uvid u pojmove koji učeniku nedostaju u odnosu na znanje tutora. Model učenika je pravi podskup modela tutora i učenjem se širi do potrebne razine. Naravno, znanje učenika ne

mora biti pravi podskup znanja eksperta, a način rješavanja problema početnika neće biti jednak načinu na koji bi stručnjak riješio problem. Učenik će često imati miskoncepcije, a ponekad može imati i informacije koje stručnjak ne posjeduje. Stoga je Serengul (1998.) model prekrivanja proširio na model zbrke (engl. *perturbation model*). Model se može implementirati proširenjem znanja stručnjaka, o čemu je bilo govora u odjeljku modula stručnjaka. U tom smjeru su razvijeni ITS-ovi koji model učenika oblikuju i s knjižnicom pogrešaka (engl. *bugs libraries*) koja je dodana znanju modela stručnjaka. Sustav koristi netočne odgovore učenika pretraživanjem u knjižnici kako bi ga bolje modelirao. Stankov ističe problem stvaranja ovakve knjižnice, jer ako se pojavi pogreška koju knjižnica ne sadrži model učenika uspoređuje ponašanje učenika sa skupom drugih pogrešaka što lako dovodi do pogrešne procjene miskoncepcije. Knjižnica pogrešaka se može širiti, osim primjerima iz pedagoške literature, pažljivom analizom rada ITS-a i "ručnim" ispravljanjem. Drugi način je koristiti tzv. djelomičnu knjižnicu pogrešaka gdje se uz skup pravila određuju miskoncepcije za vrijeme dijagnoze. Ovaj pristup je najnoviji i zbog najmanje istraženih svojstava i istaknutih problema oko njihovog prepoznavanja još je uvijek najmanje korišten (Stankov, 2010.).

Zaključak odjeljka o modulu učenika, koji je najveći predmet rasprava i plodno tlo za buduća istraživanja donosim iz pedagoškog aspekta. Nwana (1990.) koristi prema Self-u dvadeset različitih ciljeva i svrha (funkcija) modula učenika i poopćava ih u šest skupina: **korektivna** (za otklanjanje miskoncepcija u znanju učenika), **elaborativna** (za upotpunjavanje znanja), **strategijska** (za oblikovanje stila podučavanja u odnosu na prethodne dvije skupne), **dijagnostička** (za pronalaženje grešaka), **prediktivna** (za određivanje vjerojatnih odgovora) i **evaluacijska** (za procjenu učenika i samog sustava). Smatra kako popis nije ni potpun ni dovoljno općenit, a u modulu učenika općenito vidi dvije svrhe (funkcije) na najopćenitijoj razini: izvor informacija o učeniku i opis njegovog znanja. Korektivnom funkcijom zaključuje se o aspektima učenikovog ponašanja koji nisu uočljivi iz modela. Ti zaključci mogu pomoći pri prepravljanju znanja učenika, što je ključno za pedagošku komponentu ITS-a jer može biti korišteno za svaki od šest navedenih skupina funkcija. Opis znanja učenika koji stvori sam sustav prema poželjnoj razini znanja izravno doprinosi procjeni savladanog gradiva (svrha evaluacije), prema čemu se podučavanje usmjerava na dijelove znanja koji su slabije savladani (svrha elaboracije). To je osnova principa preklapanja po kojem se traže razlike u znanju stručnjaka i učenika. Nwana ističe problem korekcije, s obzirom da razlika može slijediti i iz miskoncepcija.

Zbog toga model učenika mora sadržavati i eksplicitni prikaz pogrešnog shvaćanja ciljanog znanja (svrha dijagnostike i korekcije). Konačno, Nwana navodi kako bi model učenika trebao biti izvršan (engl. *"executable or runnable"*) kako bi modul učitelja, koji će se biti objašnjen u sljedećem odjeljku, mogao predvidjeti ponašanje učenika i to iskoristiti u svrhu strategije za oblikovanje modela podučavanja.

1.3.3. Modul učitelja

Treća komponenta ITS-a, modul tutora ili učitelja je skup aktivnosti podučavanja koje se odvijaju tijekom nastave. Kvalitetan tutor preduvjet je za osposobljavanje učenika u određenom područnom znanju. Modul tutora usko je povezan s modulom učenika koristeći saznanja o stvarnom učeniku i njegovim motivacijskim ciljevima, oblikujući nastavu na optimalan način za pojedinca. Algoritam vođenja podučavanja osim toga sadrži potrebnu razinu znanja iz modula stručnjaka, savjetovanja i predstavljanja različitih aktivnosti tijekom podučavanja i uspješnu povratnu informaciju pri pogreškama učenika. Nwana navodi kako su odluke modula tutora suptilne, a male promjene u organizaciji nastave mogu doprinijeti potpuno različitim iskustvima učenja. U ITS-u se koristi tutorski dijalog za koji su važne komponente teorijskog okvira podučavanja, scenarija učenja i načela za implementaciju tutorskih znanja. Interdisciplinarnost inteligentnih tutorskih sustava najviše dolazi do izražaja upravo u strukturiranju modula učitelja.

Teorijski okvir podučavanja sadrži vrlo širok spektar načela preuzetih iz teorija odgoja i obrazovanja. Stankov, prema hrvatskoj literaturi psihologije obrazovanja, razlikuje tri pristupa teoriji učenja.

Bihevioristički pristup (prema engl. *behaviour* - ponašanje) istražuje ponašanje ljudi u različitim situacijama, pokušavajući uspostaviti zakonitosti, opažanjem i mjerenjem. Zbog toga se bihevioristi manje zanimaju za pamćenje, mišljenje i sl. kognitivne procese, a usmjeravaju se na povezivanje podražaja i iskustva ljudi s njihovim ponašanjima. Prema bihevioristočkom poimanju okolina u potpunosti utiče na ponašanje pojedinca, što objašnjava Wexlerov model učenika kao praznog podskupa baze znanja, ideje čovjeka kao "tabule rase" preuzete iz srodnog empirizma (Locke). Prema biheviorističkim idejama stvoreni su prvi složeniji alati za podučavanje, a takav pristup učenju uveo je u nastavu sistematične elemente poput prikaza ishoda učenja, obradu gradiva u malim koracima, neposrednu povratnu informaciju, ponavljanje, potkrjepljenja u obliku kazne i nagrade te

stupnjevanje napredovanja učenika. Iako je ovaj pristup još uvijek vrlo raširen u ITS-u upravo zbog sistematičnosti, Stankov ističe probleme i ograničenja ovakvih sustava, ponajviše zbog nemogućnosti prilagodbe razini znanja učenika. S obzirom na aktualne trendove u odgoju i obrazovanju, i kod ITS-a najviše uspjeha postižu sustavi bazirani na konstruktivizmu.

Konstruktivizam (prema engl. *construct* - graditi) je pristup podučavanju zasnovan na ideji kako ljudi uče gradeći znanje, spajajući nove informacije koje su kompatibilne svom trenutnom znanju, odnosno kako je svaka nova spoznaja rezultat mentalne konstrukcije na koju utječe vlastito iskustvo. Kognitivistički pristup (prema engl. *cognition* - spoznaja), iz kojeg konstruktivizam potječe, usmjeren je na unutarnje promjene u znanju, psihomotornim vještinama, mišljenju i sl. Iako biheviористi imaju uporište u svojim idejama s obzirom na znanstvenost (izravno promatranje i mjerenje), očito je model podražaj-reakcija prejednostavan za potpuno objašnjavanje vrlo složenog ljudskog ponašanja. Kognitivna teorija nastoji odgonetnuti procese kako ljudi uče, pamte i razmišljaju. Učenje se tako objašnjava kao "proces organiziranja, pohranjivanja i traženja veza među informacijama" (Stankov, 2010.). Konstruktivističke ideje o poučavanju bile su drastično različite od biheviorističkih, koje su donedavno u tom području bile dominantne. Znanje se gradi, a ne prenosi, a učenje je aktivan proces umjesto pasivnog i jedinstven je za svakog pojedinca. Aktiviranjem učenika javlja se povećana motivacija za učenjem u obliku znatiželje, postignuća, i drugih oblika, čak i zabave. Npr. Nwana ističe kako je u lekciji ponekad učinkovitije dati učeniku vremena da sam istraži problematiku nego prerano intervenirati, iako dobar učitelj nikad ne može potpuno uništiti motivaciju ako ona postoji. Samokontrola učenja je važna komponenta konstruktivizma, poticana idejom da se od učenika ne traži slijepo prihvaćanje gledišta učitelja, nego se učenje odvija kroz komunikaciju. Stoga učenici razvijaju i socijalne i komunikacijske vještine te su sposobni aktivno izgraditi znanje i vještine za rješavanje problema u novim, nenaučenim situacijama. Ben-Ari 1998. g. prvi uvodi kratke smjernice konstruktivizma u informatici, a konstruktivizam na slikovit način vidi kao rekurzivan proces koji bi se kao takav mogao implementirati i u ITS-u. Smatra kako bi se s primjenom učenja programiranja trebalo početi nakon usvajanja osnovnih pojmova i izgradnje održivog mentalnog modela računala (naglašava kako mora biti eksplicitno, razumljivo podučen), s obzirom da se učenici prečesto zasite velikim brojem koncepata. Strogi je protivnik samostalnog učenja programiranja po modelu pokušaj-pogreška. Ističe važnost vođenog podučavanja i grupnog

rada s obzirom da učenička poimanja o informatičkim konceptima lako mogu biti krivo shvaćena - postati miskoncepcije, koje se prema konstruktivizmu nazivaju alternativne koncepcije ili okviri (jer možda u nekom području nisu pogrešni) te se puno teže ispravljaju nego znanje koje nedostaje. Ben-Ari naglašava kako klasične knjižnice pogrešaka u ovom slučaju nisu učinkovit način za njihovo ispravljanje jer ne daju dostatno objašnjenje i ideju o dopunjavanju da bi promijenili trenutno znanje učenika. Također, navodi kako efikasnost u rješavanju problema ne garantira razumijevanje i potiče učenje kroz modeliranje dijelova gradiva te konstruiranje općenitog modela, pomoću primjerice semantičkih mreža. Kritičari konstruktivizma ističu težinu organizacije ovakvog oblika nastave, procjenu nužne razine zrelosti i sposobnosti učenika za uspjeh. Istaknuti su i ekstremni zaključci u filozofiji koji trivijalno proizlaze iz konstruktivizma, poput solipsizma (svijet postoji samo kao vlastito mentalno stanje uma) u kojima iščezava potreba za ikakvom etikom, ili danas popularnih teorija zavjera. Stankov navodi neke smjernice za razvoj ovog modula ITS-a poput jednostavnih procedura učenja, intuitivno oblikovanih sučelja te ne prečeste povratne informacije o radu učenika. Nužne su i jasne i precizne upute za rad, posebice učenicima koji ne shvaćaju ili ne mogu samostalno savladati gradivo.

Pored ovog važnog pristupa učenju razlikuje se još i socijalni pristup, koji povezuje elemente biheviorističkog i kognitivističkog pristupa. Ovaj je pristup poznat kao i učenje po modelu. Zasniva se na ideji kako je ponašanje većinom stečeno te da se oblikuje pod utjecajem i pojedinca i socijalne okoline. Model je, u ovom slučaju, osoba od koje se uči, odnosno osoba koju se pokušava imitirati, a učenik isključivo onaj kojeg se uči. Istaknuta su četiri osnovna načina ovog pristupa učenja (Stankov, 2010., prema Zarevski, 2007.). Učenje modeliranjem podrazumijeva promatranje modela koji pokušava utjecati na znanje učenika. Ako učenik oponaša model, a ne razumije postupak ili to ponašanje radi se o učenju imitacijom. Treći oblik, učenje promatranjem se odvija kad model uopće nema namjeru utjecati na "učenika", ali on uzima u obzir potkrjepljenja i posljedice ponašanja modela te tako mijenja i svoje ponašanje. Konačno, kod simboličkog učenja model samo opisuje obrazac ponašanje učeniku bez pokazivanja stvarnog postupka rješavanja problema.

Svi se ovi pristupi učenju moraju kombinirati u modulu učitelja, s obzirom da se međusobno nadopunjavaju za optimalnu individualizaciju i funkcionalnost. Sam scenarij učenja je planiranje načina i situacija u kojima se učenje odvija, a temelji se na teoretskom

okviru. Najvažniji kriterij scenarija učenja je razina dijaloga učenika i učitelja (Stankov, 2010.). U tradicionalnim sustavima učilo se paradigmom "pravilo-primjer-praksa", gdje učenik nije aktivno sudjelovao u podučavanju. Vođeno poučavanje povremenim, ali nenametljivim savjetima je vrlo popularno i može biti iznimno korisno, u modulu učitelja se oblikuje poput "računala trenera". Stankov također navodi kako postoji čitav spektar paradigmi s obzirom na razinu uključenosti učenika, poput učenja kroz igru, razgovijetnog eksperta (SOPHIE, posebno kasnije inačice), simulatora, do učenja pomoću otkrivanja gdje se učenik vodi vlastitom znatiželjom, kao dijametralno suprotne paradigme od tradicionalnog podučavanja.

Jasno je kako je vođenje poučavanja dinamična vještina utemeljena na znanju i iskustvu učitelja, a u ITS-u se ta vještina mora računalno modelirati. Stankov ističe neka načela za implementaciju tutorskih znanja. Navodi kako Woolf i McDonald (1984.) predlažu izgradnju modula učitelja nakon izgradnje modula stručnjaka i modula učenika jer je tad već uspostavljena procjena razine znanja koju posjeduje učenik. Osim toga, učitelj mora odabrati pristup rukovanja pogreškama, odnosno hoće li istu zanemariti, istaknuti, ispraviti ili voditi učenika prema prepoznavanju i ispravljanju vlastitih grešaka. Prema Stankovu pravila izvođenja su osnova prikaza tutorskih znanja u ITS-u, a oni obuhvaćaju učenikove pogreške, prepoznate situacije u procesu poučavanja, učenikove postupke, znanje koje se poučava i usporedbu između znanja učenika i učitelja. Pravila se preklapaju s načinom dijaloga u scenariju učenja, a uključuju i strategije za organiziranje nastavnih cjelina i jedinica. Stankov ističe kako se metodologija podučavanja, posebno dijela gdje se gradivo uvježbava ili ponavlja, temelji na razlaganju (engl. *decomposition*) procesa u potprocese, što je najviše izraženo kroz simulacije. Pravila se mogu po potrebi dodavati, oblikovati ili odbacivati, čime se u istom tutorskom sustavu može i eksperimentirati s različitim pristupima podučavanja, a u "uvježbanom" tutorskom sustavu, koji bi trebao biti programiran za stalno usavršavanje (Nwana, 1990.) te ih dinamički kombinirati za najbolji pristup reakciji učenika.

1.3.4. Modul komunikacije

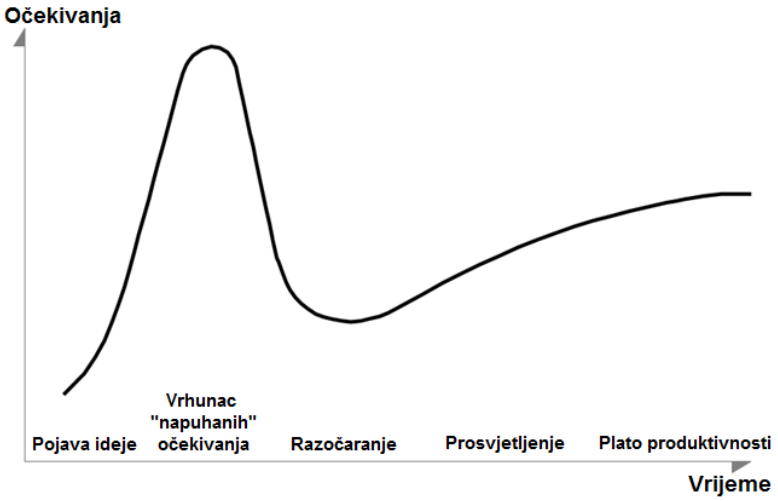
Upravljanje aktivnostima podučavanja koje vode učinkovitom nastavnom procesu zahtijeva i četvrtu, komunikacijsku komponentu između učenika i sustava. Kako je vidljivo na slici 1., radi se o dvosmjernom prevoditelju unutarnjeg prikaza ITS-a i jezika sučelja koji je razumljiv učeniku. Bez obzira koliko ITS bio detaljno razrađen, bez kvalitetne komunikacije ne može biti uspješan, stoga se u novije vrijeme (Wenger, 1987.) ovaj modul izdvaja iz modula učitelja kako posebna komponenta. Nwana navodi kako bi bilo pogrešno modul komunikacije smatrati samo pomoćnim alatom učitelja ITS-a iz dva razloga. Odgovarajuće sučelje može znatno poboljšati predstavljanje problema, a kao konačni oblik prikaza sustava, može biti odlučujuće hoće li ga učenik prihvatiti ili ne. Drugi razlog je ubrzan napredak u medijskoj tehnologiji koji omogućuje kvalitetnije i složenije alate, čime se nameće imperativ za korištenje u inteligentnim tutorskim sustavima. Bitno je organizirati sučelje na način da koristi različite oblike medija kako bi aktivirao više učenikovih osjetila, a ne preopteretio jedno. Zaključci proizlaze iz novijih pedagoških istraživanja podijeljene pažnje (npr. Mayer i Moreno, 1998.), koji su pokazali kako učenici značajno bolje shvaćaju animacije uparene s govorom, a ne tekstom. Stankov ističe osim sučelja i dvije usko povezane funkcije modula komunikacije: sustav vođenja razgovora i generiranje zadataka. Još Carbonell (1970.) ističe važnost komunikacije čovjeka i računala kao dvije različite informacijske strukture, u kojima se ona bez strogog protokola odvija prema ciljevima i potrebama obje strane, slično kao ni što živi učitelj ne bi trebao slijediti slijepo izlaganje, dopuštajući učeniku da sudjeluje u dijalogu. Određeni sustavi nastave pomoću računala imali su mogućnosti grananja u komunikaciji na temelju učenikovih odgovora. Stankov navodi kako se od tutorskog razgovora u ITS-u očekuje fleksibilnost, a razvoj sofisticiranih shema za vođenje razgovora doprinio je i napretku istraživanja ovih sustava. Danas ITS koriste vođeni dijalog organiziran na razne načine. Primjerice, kod pojmovnog stabla pretpostavki sustav odlučuje na kojoj razini komunicirati s učenikom, i kako procijeniti odgovore, a iste funkcije se mogu postići i semantičkim mrežama. DMN mreža (engl. *Discourse Manangement Network*), prema Woolfovoj implementaciji 1984. g. omogućuje, pored već navedenih funkcija, i izražavanje u prirodnom jeziku. Uključuje tri uzastopne razine (pedagošku, strategijsku i taktičku) koje omogućuju hijerarhijsko bistrenje ciljeva podučavanja prema strategiji (npr. za nadopunu znanja) ili dalje prema većoj individualizaciji programa. Iz DMN mreže za vođenje razgovora organizirana je proširena mreža prijelaza (engl. *Augmented Transition Network*, ATN), u kojoj se, također

hijerarhijski organizirani čvorovi i veze koriste za pohranu informacija za kasniju uporabu, a dinamički se uključuju i isključuju ovisno o pravilima i povratnoj informaciji učenika. Stankov zaključuje kako razvoj modela vođenja razgovora ukazuje na potrebno što uže specijalizirano pedagoško znanje za uspješno podučavanje ITS-a (od strategija do znanja o razgovoru). Konačno, samo generiranje zadatka, koje je vrlo srodno samom vođenju razgovora se zasniva na sličnim pretpostavkama.

Rani oblici CAI težili su generiranju novih problema s obzirom na odgovore, najčešće mijenjanju varijabli u osnovnim tipovima zadataka. Pronicanjem dublje strukture problema i korištenjem umjetne inteligencije omogućeno je bogatije predstavljanje znanja iz kojeg se mogu stvarati novi problemi, koje tutor može ne samo riješiti, već i ponuditi smislene povratne veze s obzirom na pogrešne odgovore učenika. Imperativ razvoja budućih inteligentnih sustava upravo je razvijanje ovakvog oblika generiranja problema.

1.4. Budućnost inteligentnih tutorskih sustava

Kao zaključak cjeline i smještaj ITS-a u kontekst današnjih postignuća, očekivanja i potencijala u budućnosti, prikladno je primijeniti Amarin zakon na interes za ITS-om, odnosno objasniti njihov već opisani uspon i pad popularnosti, očekivanja te praktični potencijal. Amara tvrdi kako ljudi uvijek precjenjuju nove tehnologije i potom teže k njihovom dugoročnom podcjenjivanju i zanemarivanju. Dakle, ideje postižu puni tehnološki potencijal tek nakon što prođu fazu najveće popularnosti i očekivanja. Amarin zakon je pak izveden iz ciklusa neumjerene pozornosti[3] (engl. *hype cyle*) kojeg je razradila informacijska tvrtka Gartner. Ciklus opisuje životni put ideje u pet faza s obzirom na prihvaćenost u društvu i očekivanja od te ideje, a grafički je prikazan na slici 2.



Slika 2. Gartnerov ciklus neumjerene pozornosti

Nove ideje postižu medijsku pažnju i privlače ulaganja, čak i prije njihove korisne implementacije. S obzirom na popularnost u porastu, velik broj ljudi zanima nova ideja i njene primjene, bez obzira što su realno, jednim dijelom očekivanja i prenapuhana. Stoga, nakon faze vrhunca neumjerene pozornosti, zanimanje naglo pada, a društvo od nje ne očekuje više mnogo. Nova ulaganja slijede samo ako se ideja dorađuje i njena primjena ostaje ekonomična. Većina ideja potpuno propada, no s vremenom se pokazuje koliko je sama ideja uz popratnu tehnologiju uistinu kvalitetna, kako se shvaćaju i bistre novi korisni

[3]Hrvatski naziv prema http://www.infotrend.hr/clanak/2008/11/nista-nas-ne-smije-iznenaditi,24,542.html

načini njene primjene. Bez obzira radi li se o novim područjima za koje ideja nije ni bila prvobitno osmišljena ili pak o novim saznanjima i doradama koji upotpunjuju korištenje, ona polako napreduje i sazrijeva prema stabilnoj razini prihvaćenosti i društveno očekivanje je realistično kako se ispunjava njen potencijal.

Inteligentni tutorski sustavi, kao i umjetna inteligencija općenito, zasigurno su prošli fazu pretjeranih očekivanja osamdesetih i prije nego su tehnološki postigli potrebnu funkcionalnost, iako su prema Gartneru i kognitivni tutori i strojno učenje još u toj fazi (Panetta, 2017., Groff, 2017.) S obzirom na znatan trud akademske zajednice za stalno poboljšanje inteligentnih tutorskih sustava te sve veću primjenu u raznim područjima može se zaključiti kako se ova ideja, odnosno njena popratna tehnologija tek razvija prema svom punom potencijalu.

2. Analitika učenja

Analitika učenja je postupak mjerenja, prikupljanja, obrade i sistematiziranog prikaza podataka o učenicima koji se koriste zbog razumijevanja i optimizacije procesa učenja.

2.1. Razvoj i definicije

Učenje je interaktivni proces. Ovisno o primijenjenom teorijskom okviru podučavanja, učenici surađuju s učiteljima i tutorima, s nastavnim sadržajem ili s drugim ljudima iz okoline. Mnogi učitelji ulažu znatne napore u oblikovanje nastavnog sadržaja kako bi ovakvim interakcijama poboljšali podučavanje. Bez obzira na pristup, uvijek se javlja niz pitanja u cilju samoprocjene učitelja. Koliko je učinkovit bio sat ili cjelina ili pak kolegij? Ispunjava li potrebe učenika? Kako bolje podržati potrebe učenika? Koje interakcije su učinkovite i kako ih dodatno poboljšati? (Elias, 2011.)

Naravno, uobičajeni pristup odgovaranju na ova pitanja je evaluacija učenika, najčešće uobličena na kraju kolegija, u vidu analize ocjena testova, praćenja napretka i općenitog dojma o učeniku. Ovakvom evaluacijom nužno je patio program zbog ograničenja poput količine podataka koje učenici žele podijeliti na kraju kolegija, upitne kvalitete prijavljenih podataka i znatne odgode primjene intervencije od strane učitelja (barem jedan semestar).

Zanimanje za korištenje ovakvih podataka za poboljšanje učenja i podučavanja je u stalnom porastu pa se iz potrebe stvorila analitika učenja (engl. *learning analytics*) kao znanstvena disciplina unutar pedagogije. Kako ističu Papamitsiou i Economides (2014.) podaci su se tijekom vremena, posebice u zadnjem desetljeću, nagomilali i za konkretne zaključke zahtijevaju nove pristupe njihovoj obradi. Analitika učenja ne uključuje samo obradu podataka (engl. *data mining*, također i hrv., češće u ekonomiji, *rudarenje podataka*), već i njihovu daljnju analizu kako bi nastavnicima bili pomoć pri boljem oblikovanju gradiva. Kod obrazovanja, naravno, pričamo o obradi obrazovnih podataka (engl. *educational data mining*, dalje u tekstu EDM). U brojnim drugim društvenim i prirodnim znanostima duže vrijeme se koriste statistički alati za analizu velikih količina podataka i prepoznavanje uzoraka u njima, kako bi se pronašle zakonitosti i utvrdili zaključci o području istraživanja. Analitika učenja potječe ili je usko vezana uz nekoliko područja istraživanja.

Poslovno obavještavanje (engl. *bussiness intelligence*) je primjerice već odavno razvijano područje istraživanja (Devens, 1865.), tj. prikupljanja informacija iz svih dostupnih izvora, najčešće medija, publikacija i sl. Njihovom učinkovitom analizom donose se predviđanja, odnosno odluke koje mogu unaprijediti poslovanje tvrtke. Danas se poslovno obavještavanje zasniva na računalnoj tehnologiji, bilo za prikupljanje, bilo za bržu obradu podataka, a područje istraživanja postalo je vrlo široko i uključuje skup različitih postupaka koji su upravo potekli iz ove djelatnosti. Primjerice, web analitika se, kao postupak prikupljanja i obrade podataka o posjetiteljima neke internet stranice, koristi za organizaciju i optimizaciju. Kako Elias dodaje, vrlo je djelotvoran način za prikupljanje poslovnih podataka, s obzirom da računala danas s lakoćom računaju trendove za tisuće i milijune korisnika, prema čemu se mogu postaviti hipoteze, vrlo brzo reagirati na promjene temeljene na tim pretpostavkama i vrlo brzo iste testirati.

Goldstein i Katz (2005.) su primjenu načela i nekih alata poslovnog obavještavanja na akademskim istraživanjima opisali izrazom akademska analitika (engl. *academic analytics*). Njihov cilj je bio proučavati tehnološke i upravljačke čimbenike koji utječu na prikupljanje, analizu i korištenje podataka u visokom obrazovanju. Campbell i Oblinger (2007.) su kroz akademsku analitiku direktno proučavali uspjeh studenata, kojeg su nazvali glavnim zadatkom obrazovanja, kroz dva faktora - retenciju (zadržavanje znanja) i prosjek ocjena. Dalje, Elias navodi kako su Norris i sur. (2008.) dodatno naglasili važnost korištenja obrazovnih podataka za predviđanje ishoda podučavanja i pravodobnog djelovanja na njihovo poboljšanje. Ovaj proces su nazvali analitika djelovanja (engl. *action analytics*), a iako često zapostavljen, proces daje novu bitnu svrhu akademskoj analitici, slično kao i kod poslovnog obavještavanja, kombiniranjem dostupnih informacija. Zaključci se primjenjuju na valjane postupke i alate, kako bi se konačno u određenim kolegijima koristili dokazano pouzdani modeli podučavanja. Takve promjene su ne samo djelotvornije u obrazovnom pogledu, već i praktične jer vode k smanjenju troškova ustanova koje ih primijene. Vrlo slično, Arnold (2010.) priča o analitici kao alatu kojeg bi obrazovne ustanove mogle iskoristiti za obavještajne podatke koje vode k promjenama umjesto čistim informacijama o pojedinačnim učenicima ili kolegijima.

S obzirom da se radi o području u razvoju i nema univerzalno prihvaćene definicije poput vrlo općenite, formulirane na početku ovog poglavlja, razni istraživači ovog područja iznose još nekoliko definicija i pogleda na analitiku učenja.

EDM se orijentira na razvoj, istraživanje i primjenu računarnih metoda za prepoznavanje uzoraka u velikoj količini obrazovnih podataka koje bi bez računala bilo vrlo teško ili nemoguće obraditi (Romero i Ventura, 2013.). Cilj ovog područja je također razvijanje i primjena metoda koje bi dovele do zaključaka o obrazovanju općenito (Avella i sur., 2016.).

S druge strane, analitika učenja povezuje navedena područja istraživanja u primjeni kod obrazovanja, s naglaskom na akcijsku analitiku, prema kojoj se donose zaključci i djeluje u skladu s njima (Elias, 2011.). Prema definiciji prve međunarodne konferencije analitike učenja i znanja 2011. g. (engl. *Learning Analytics and Knowledge, LAK*), "analitika učenja je mjerenje, prikupljanje, analiza i predstavljanje podataka o učenicima", zbog "razumijevanja i optimiziranja učenja i okruženja u kojima se ono odvija"[4].

Campbell i sur. (2007.) navode kako analitika općenito spaja velike skupine podataka sa statističkim tehnikama i modelima predviđanja (engl. *predictive modeling*). Može se promatrati kao obrada dostupnih podataka za stvaranje podataka o akcijskom znanju.

Papamitsiou i Economides (2014.) šire definiciju LAK-a, a analitiku učenja i EDM nazivaju ekosustavom metoda kojima se podaci stalno prikupljaju, obrađuju, predstavljaju zaključci i u skladu s njima neprestano djeluje na napredak obrazovnog okruženja i procesa učenja. Naglašavaju holistički pristup analitike učenja za razliku od EDM-a koji redukcionistički promatra pojedine komponente u potrazi za trendovima i uzorcima, bez kojih se pak ne može zaključiti o potpunoj slici problema, zbog čega su ova područja istraživanja komplementarna.

I Avella (2016.) naglašava kako analitika učenja koristi modele predviđanja koji pružaju mogućnosti donošenja akcijskih informacija, u čemu se analitika najviše razlikuje od EDM-a. Navodi kako se i drugi istraživači slažu s multidisciplinarnošću analitike učenja te kako je njena svrha stvoriti edukacijske prilike u kojima je moguće individualizirati učenikove potrebe i sposobnosti kroz aktivno djelovanje, dok EDM oprečno samo učeniku pruža automatiziran odgovor. Analitika učenja koristi razvijene metode EDM-a za pomoć u prepoznavanju i otklanjanju grešaka učenja te organiziranju obrazovnog sustava, dok se EDM, kako je spomenuto, bavi otkrivanjem novih metoda, čime se ponovno zatvara krug na još jednoj razini, a obe su discipline očigledno usko povezane i nerazdvojne, a velikim dijelom se i preklapaju.

[4] http://tekri.athabascau.ca/analytics/

Baker i Siemens (2013.) smatraju očekivanom pomutnju u definiciji ova dva područja koja se bave sličnom problematikom, ali ističu bitne razlike. Istraživači EDM-a se orijentiraju na automatizirane metode otkrivanja pravilnosti u podacima, modeliranju konstrukta i veza među njima, što može, na čemu autori inzistiraju, pomoći u modelima predviđanja. Istraživači analitike učenja ljudskim resursima proučavaju podatke uz holistički pristup vezanim konstruktima, čime se bolje i razumljivije može objasniti određeni fenomen. Prema novijim istraživačima, nadodaju, oba područja, a posebno analitika, se usko povezuju uz pedagoške teorije, čime je naglašen odmak od paradigme tehnologije kao glavnog alata za obradu podataka.

2.2. Procesi i metode u analitici učenja

Također, ne postoji ni sveobuhvatna sistematizacija analitičkih procesa jer su se razvijali nezavisno u raznim područjima prije nego su primijenjeni u analitici učenja, ovisno o potrebama istraživača. Prema definiciji Papamitsioua i Economidasa iz prošlog poglavlja, koja zapravo navodi procese analitike učenja, kao i prema drugim istraživačima, može se zaključiti vrlo općenito kako je prvi korak mjerenje i prikupljanje podataka kao priprema za njihovu obradu još tijekom obrazovnih aktivnosti. Zatim se podaci analiziraju i interpretiraju kako bi učenicima, učiteljima i ustanovama približili postignuća i ispunjavanje ciljeva, a u konačnici olakšali donošenje odluka svih sudionika učenja na svim razinama. Elias navodi još nekoliko ključnih ideja za razmatranje i primjenu u analitici učenja. Te ideje nisu nove, već su preuzete iz drugih, najčešće filozofskih i upravo obrazovnih područja.

Kontinuum znanja, kojeg primjerice Baker (2007.) navodi kao početni koncept u poslovnom obavještavanju, postoji kao filozofski pojam dvjestotinjak godina. Najniža razina kontinuuma su podaci, koji su sami po sebi neupotrebljivi bez uspostave međusobnih odnosa među njima, nakon čega postaju informacije. Informacija može odgovoriti na jednostavna deklarativna pitanja. Analizom i sintezom skup informacija postaje znanje, a znanje već ima dublje odgovore o problemu. Baker zaključuje kako praktičnom primjenom znanja ono postaje mudrost, a u postupku posljednjeg prelaska ključnu ulogu ima predikcijska analiza koja stvara akcijsko znanje (engl. *actionable knowledge*). Dakle, detaljnom obradom osnovnih, sirovih podataka mogu se donijeti ne samo smisleni, već i u budućnosti upotrebljivi zaključci o problemu. Bez obzira, Elias ističe problem obrade prikupljenih podataka u većini akademskih zajednica. Navodi kako u najboljem slučaju, njihovom obradom oni postaju informacije o uspjehu učenika koji se skupno prikazuju kao mjerilo uspjeha sveučilišta. Već je spomenuto kako je analitika učenja kontekstualno mnogo šira od obrade obrazovnih podataka (EDM), stoga je, kako zaključuje, i razvoj akcijskog znanja sveden na početnu fazu hrpe beskorisnih podataka.

Campbell i Oblinger (2007.) su akademsku analitiku opisali kao uređaj za odlučivanje i vođenje postupaka koji radi u ciklusima od pet koraka: prikupljanje, prikaz, predviđanje, djelovanje i bistrenje. Slično kao i kod kontinuuma znanja, a u skladu s pedagoškim teorijama stjecanja znanja (npr. spomenuti Carricov prikaz modula stručnjaka u ITS-u) ciklus započinje prikupljanjem podataka koji se tek nakon obrade prikazuju kao smislene

informacije, kako bi se omogućilo predviđanje zasnovano na znanju i promišljenom djelovanju. Dodatkom posljednjeg koraka, bistrenja, odnosno pročišćavanja stečenog znanja i postupaka, ciklus se iznova ponavlja.

Dron i Anderson (2009.) svojim modelom zajedničke primjene (engl. *Collective Application Model*), koji koriste i kod analitike učenja, također naglašavaju važnost ciklusa otkrivanja znanja. Njihov se model sastoji od pet razina podijeljenih u tri cikličke faze (prikupljanje informacija, njihova obrada i konačno prikaz). Razlika između čiste obrade podataka i analitike je upravo u prikazu podataka na način koji utječe na poboljšanje sustava. Znanje se otkriva upravo spomenutom akcijskom analitikom. Elias također u kasnijim odjeljcima navodi kako ovi istraživači analitiku učenja vide kao sustav ljudskih i računalnih komponenti gdje se naglašava kognitivni, tehnički, ali i socijalni aspekt analitike. Slični zaključci nekih istraživača ovog područja ne čude jer je jasno kako su, primjerice, programska i tehnička podrška korisne onoliko koliko je korisnik spretan u njihovoj upotrebi.

Kao zaključak svog rada, Elias donosi sveobuhvatni model analitike učenja koja koristi četiri vrste tehnoloških resursa potrebnih za zaokruživanje ciklusa u tri faze stalnog bistrenja znanja za učenje i podučavanje. Ovim postupkom kvalitetni podaci, najnovije pedagoške teorije i tehnološke inovacije zajedno pomažu u ispunjavanju potreba cjelokupne obrazovne zajednice. Razvoj analitike teži k usavršavanju procesa učenja, poboljšanju uspjeha učenika te sustava kroz individualizirano planiranje i savjetovanje o najboljim metodama kroz cjelokupno školovanje. Alati za neke neprimijenjene ideje postoje stoga je budući izazov analitike učenja olakšati i podržati promjene koje su potrebne na svim razinama edukacije.

Prema Bakeru i Siemensu (2013.) **predikcijske metode** su ključni dio obrade obrazovnih podataka (EDM). Njima je cilj razviti model kojim se može zaključiti o određenim varijablama među podacima, na temelju drugih dostupnih podataka. Primjerice, mogu se skupljati podaci o učenicima koji su napustili studij, potom razviti model predviđanja napuštanja studija, provjeriti ga na novim skupinama učenika (na kojima se nije prvobitno formirao taj model) te konačno model s eventualnim preinakama koristiti kod predviđanja odustajanja nekih novih studenata. Općenito, u analitici se koriste tri vrste predikcijskih modela. Kod klasifikatora (engl. *classifiers*) varijabla koja se predviđa može se svrstati u neku od unaprijed poznatih kategorija, bilo stablima odlučivanja, pravilima i sl. Regresori ili regresijski modeli (engl. *regressors*) uključuju konstantnu varijablu koja se predviđa,

npr. neki broj, tražeći veze s drugim varijablama. Model latentne procjene (engl. *latent knowledge estimation*), kako i naziv govori, uključuje latentnu varijablu, odnosno onu koja ne može biti direktno opažana. Primjerice, znanje učenika u skupu sposobnosti se procjenjuje s obzirom na uzorke u odgovorima pri provjeri traženih sposobnosti. Modeli koji se koriste u online učenju stalno procjenjuju trenutačno znanje učenika i u tome se razlikuju od psihometrijskih modela u pisanim provjerama znanja. Najrašireniji algoritmi u ovom modelu su danas *Bayesian Knowledge Tracing (*BKT, Corbett i Anderson, 1995.) i *Performance Factors Analysis* (PFA - Pavlik, Cen i Koedinger, 2009.).

Algoritmi otkrivanja strukture u podacima su zapravo oprečni metodama predikcije, jer se koriste kad te strukture nisu unaprijed poznate. Istraživači pokušavaju analizom podataka naći pravilnosti u podacima i općenitije zakonitosti. Baker i Siemens izdvajaju četiri metode otkrivanja strukture. Analiza podataka grupiranjem[5] (engl. *clustering*) je svrstavanje podataka u prirodne, intuitivne skupine, kako bi se olakšala daljnja obrada, a s obzirom da podaci u svakoj skupini imaju više međusobnih sličnosti nego podaci iz različitih skupina, može se zaključiti o karakteristikama objekata kojima pripadaju. Faktorska analiza (engl. *factor analysis*) je slična metoda, koja ide i korak dalje, grupirajući podatke u skupine latentnih faktora. Ima široku primjenu, a u analitici učenja i EDM-u se najčešće koristi za smanjenje broja izmjerenih varijabli zamjenom za latentne faktore koji ih obuhvaćaju. U analizu društvene mreže (engl. *social network analysis*) spadaju modeli veza (interakcija i odnosa) među pojedincima, njihove promjene, kao i uzoraka koji se prepoznaju iz društvene strukture. Društvena mreža se može vizualizirati sociogramom, grafičkim prikazom čvorova (pojedinci) i veza. Model otkrivanja strukture domene (engl. *domain structure discovery*) u analitici se učenja odnosi na domenu cjelokupne edukacije. Tako se npr. proučava kako je nastavni sadržaj povezan sa znanjem ili vještinama učenika, kako bi se oblikovale skupine zadataka i nastavne cjeline. U inteligentnim tutorskim sustavima ovaj se model često upotrebljava za testiranje podataka i za praćenje napretka u učenju.

U EDM-u se još vrlo često koristi metodologija **obrade odnosa među podacima** (engl. *relationship mining*) koja ima slične zadatke kao i neke navedene metode otkrivanja strukture, uz kombiniranje s predikcijskim metodama, s obzirom da idejno vodi do novih otkrića, hipoteza i potreba za novim istraživanjima. Ova skupina metoda se strogo

[5]Prihvaćen naziv, primjerice prema Špišić, V. i Štedul, I. "*Clustering with Open Source Tools*", Veleučilište u Karlovcu, 2012., http://hrcak.srce.hr/87439

orijentira na odnose među varijablama umjesto cjelokupne strukture podataka, odnosno k boljem razumijevanju sitnijih dijelova cjeline. Baker ponovno razlikuje četiri skupine metoda u kojima se najčešće proučavaju izražene veze među podacima. Obrada pravila asocijacije (engl. *association rule mining*) je još jedna metoda koja potječe iz poslovnog obavještavanja. Zasnovana je na traženju *if-then* pravila izvedenih iz veza među podacima, tj. skupa varijabli koje imaju određene vrijednosti prema kojima će najvjerojatnije i neka druga varijabla biti određena. Primjerice, ako kupac ujutro u trgovini kupi somun, kajmak i luk, vjerojatno će za ručak jesti ćevape. Ova metoda je vrlo korisna i u predikcijskim modelima inteligentnih tutorskih sustava. U obradi korelacije (engl. *correlation mining*), kako i samo ime govori, testiraju se hipoteze o povezanosti dvaju čimbenika. Obrada uzastopnih uzoraka (engl. *sequential pattern mining*) proučava privremene veze među događajima (skup uzastopnih uzoraka), primjerice kakav način suradnje studenata vodi k uspješnijim grupnim projektima (vrlo uspješan primjer Perera i sur., 2009., prema Baker, 2013.). Vrlo srodna metoda je kauzalna obrada podataka (engl. *causal data mining*) u kojoj se traže uzročno-posljedične veze među događajima, konstruktima te vremenski promjenjivim varijablama.

Sve ove metode se kombiniraju kako bi se došlo do otkrića, a posebno su česti slučajevi zaključivanja o predikcijskim modelima preko drugih predikcijskih modela ili kombinacije predikcijskih modela s obradom odnosa među podacima. Baker navodi i koncept inženjerstva znanja (engl. *knowledge engineering*), prema Feigenbaumu i McCorducku (1983.) gdje se model problema racionalno donosi ljudskom inteligencijom, a ne predikcijskom analizom.

U analitici učenja može se izdvojiti još jedna metoda koja ne pripada u nijednu od ovih skupina, bistrenje prikaza podataka, s obzirom da se koristi za čovjeku lako razumljiv prikaz podataka i zaključaka o njima na kraju istraživanja. Brz i učinkovit prikaz mogu znatno olakšati rješavanje problema kao što su brzi popravci pedagoških aktivnosti. Pored raznih tipova grafova, mapa i krivulja koji se koriste u analitici učenja, zanimljiv je koncept prikaza promjene znanja (poželjno napretka) pojedinog učenika - *learnogram,* prema kojemu se mogu procijeniti trendovi uspješnih i manje uspješnih učenika.

2.3. Analitika učenja i inteligentni tutorski sustavi

Iz dosadašnjih razmatranja i primjera nije teško zaključiti kako su obrada obrazovnih podataka i analitika učenja korisna područja i u razvoju inteligentnih tutorskih sustava. Ideja ITS-a je, kako je u prošloj cjelini objašnjeno, stvaranje sustava koji će biti jednako perceptivan kao iskusni pedagog i promišljen kao iskusni učitelj, koristeći odgovarajuće strategije. Takav sustav ne samo da podržava učenje već se "trudi" da učenik stekne što bolje znanje, uvažavajući individualne razlike. Iako su mnogi sustavi dosad postigli zavidne rezultate, Baker u svom radu *"Stupid Tutoring Systems, Intelligent Humans"* iz 2016. g. naglašava jaz između ideja i postignuća ITS-a. Kroz istraživanja su se brojni koncepti pokazali kao vrlo djelotvorni, ali sustavi koji se koriste diljem svijeta nisu reprezentativni u bogatstvu ideja koje istraživači promiču. Primjerice, modeliranje učenika u teoriji je vrlo razrađeno složenim pristupima poput teorije prostora znanja (kombinacija mogućih stanja u kojem se znanje pojedinca može nalaziti) ili specijaliziranim algoritmima (spomenuti BKT i PFA algoritmi su dobar primjer) koji se u praktičnim izvedbama vrlo rijetko koriste. Baker navodi najčešći oblik modeliranja jednostavnom heuristikom, promatrajući ukupan broj točnih odgovora ili broj uzastopnih točnih odgovora. Slično, moduli učitelja na kojima se idejno inzistira mogućnost samopoboljšavanja tijekom podučavanja su testirani samo u manjim okruženjima.

Baker problem prepoznaje u inertnosti prilagodbe u primjeni naprednijih modela od teorije do testiranja među istraživačima, preko testiranja u učionicama do široke primjene. Također, jasan je i ekonomski faktor problema. Razvoj složenih sustava zahtijeva pozamašne resurse i mnogo vremena. Stoga donosi ideju o mogućoj strukturi obrazovne tehnologije u budućnosti, gdje nije potreban vrlo sofisticirani ITS, već onaj sličan današnjem, "glupi", kako ga Baker naziva, uz povremenu evaluaciju klasičnog učitelja. Tutorski sustav je tada očito moćan alat za rasterećenje učitelja. Iako se to čini kao korak unatrag prema alatima za podučavanje koji su u obrazovanju već odavno prisutni, Baker svoj stav opravdava činjenicom kako su danas podaci ključni za razvoj dizajna i ljudskih odluka u mnogo većoj mjeri nego automatiziranih sustava odlučivanja te da se ITS čitavo vrijeme zapravo uspješno koristi kao pomoć, iako je idejno zamišljen kao zamjena za živog učitelja. Obrada obrazovnih podataka (EDM) korisna je za primjene u automatskoj prilagodbi, primjerice podržavanju učenika u korištenju računalnih tutorskih sustava, dok se analitika učenja orijentira prema učitelju, bilo u obrazovnom, bilo u komunikacijskom

(odgojnom) aspektu. Baker dodaje kako su očito oba područja korisna u odgojno-obrazovnom smislu, a njihove razlike potječu iz povijesnog razvoja istraživača različitih sfera pedagogije (teoretičari i praktični predavači).

Bienkovski i sur. (2012.) analitiku učenja vide kao ključan dio ITS-a, nužan u svakom od četiri njegova modula. Proučavajući današnje tutorske sustave, ova skupina istraživača ih naziva adaptivnim sustavima za učenje, u kojem je dopuštena intervencija živog učitelja ako je to potrebno učeniku. U sustav uključuje i bazu podataka o učeniku (osobni podaci, ne razina znanja koja je sadržana u modulu učenika) koja se rijetko mijenja, a služi za svrstavanje u skupine kao dodatne varijable za proučavanje kod obrade podataka. Prediktivni model, koji se izdvaja kao posebna komponenta njegovog sustava, kombinira te podatke s tekućim napretkom učenika kako bi procijenio buduća ponašanja i faktore rizika. Izlaz modela se koristi za prikaz podataka, odnosno u komunikacijskom modulu u sistematizaciji ovog rada, ali također i u modulu tutora, za oblikovanje nadolazećeg nastavnog sadržaja kako bi se optimizirao nastavni proces.

Analitika učenja primijenjena na podacima iz ITS-a se pokazala vrlo uspješnom pri izvještavanju i prikazu zaključaka o stanju znanja učenika, koji se primjerice u modelu otvorenog učenika (engl. *open learner model*) predstavlja samom učeniku. U drugim sustavima se može predstavljati samo učiteljima, roditeljima, savjetnicima i sl. Važno je naglasiti kako se ne radi o pukoj povratnoj informaciji, već i o modelu predviđanja uspjeha, ranih znakova odustajanja od kolegija ili čitavog studija (faktori rizika, koncept preuzet iz poslovnog obavještavanja) kako bi cjelokupna zajednica mogla pomoći u otklanjanju problema pri savladavanju zahtjeva, čak i prije nego se problemi manifestiraju, što uključuje i individualna savjetovanja i ideje o promjenama u nastavnom programu.

3. Analitika učenja u inteligentnom tutorskom sustavu AC-ware Tutor

Analitika učenja koristi se za poboljšanje procesa učenja i podučavanja. U sustavu e-učenja AC-ware Tutor se detaljno evidentiraju parametri rada učenika/studenta na sustavu, što omogućuje primjenu analitike učenja na tim podacima.

3.1. AC-ware Tutor

Model prilagodljivog stjecanja znanja u sustavu e-učenja AC-ware Tutor (prema engl. *Adaptive Courseware Tutor),* (Grubišić, 2012.,), je u inteligentnim tutorskim sustavima model računalno oblikovanog, automatskog, dinamičkog i, kako sam naziv ističe, prilagodljivog generiranja nastavnog sadržaja.

Kako je spomenuto u prvoj cjelini ovog rada, tijekom posljednjih dvadesetak godina razvijeno je nekoliko verzija sustava TEx-Sys za oblikovanje ITS-a u bilo kojem područnom znanju (Stankov, 2010.). Zbog istraživanja njihovog učinka u procesu učenja i podučavanja razvijena je i metodologija vrednovanja (Grubišić, 2007.). Iako su rezultati pokazali određeni uspjeh sustava, ukazala se potreba za preinakama i poboljšanjima kako bi se učinkovitost približila klasičnom podučavanju jedan na jedan. Ta poboljšanja su predložena u nekoliko smjerova, od kojih jedan predstavlja AC-ware Tutor sustav, orijentiran na automatsko i dinamičko generiranje prilagodljivog nastavnog sadržaja izmjenama modula učitelja i učenika.

Sustav se temelji na stereotipovima učenika prema Bloomovoj taksonomiji i Bayesovim mrežama (Grubišić, 2012.). Dok se većina inteligentnih tutorskih sustava prilagođava učeniku prema stilu učenja, AC-ware Tutor se orijentira na prilagodbu prema njegovom trenutačnom znanju. Kao posljedica prilagodbe sučelja prema stilu, neki ITS-ovi otvaraju mogućnost preskakanja pojedinih elemenata gradiva, što povećava vjerojatnost pogrešnog redoslijeda savladavanja koncepata ili potpunog zanemarivanja ključnih pojmova. Stoga se u ovom sustavu kreira proces vođenog podučavanja, pri čemu se prikladan opseg gradiva predstavlja učeniku uvažavajući njegovo kognitivno stanje. Ovakvim automatskim i

dinamičkim generiranjem gradiva sustav zanemaruje promjene izgleda sučelja i orijentira se na stjecanje znanja problematike koja se obrađuje.

AC-ware tutor sadrži module stručnjaka, učenika i učitelja, dok se modul komunikacije generički ostvaruje predstavljanjem gradiva i upitnika u tekstualnom obliku. U modulu stručnjaka AC-ware Tutor sustava područno je znanje bazirano na ontologiji, u sklopu koncepata, pojmova i njihovih međusobnih odnosa, a predstavljeno matematičkim metodama iz teorije grafova. Ovaj način prikaza odabran je zbog egzaktnosti u odnosu na semantičku analizu, koja može višeznačno ovisiti o interpretacijama stručnjaka područnog znanja. Pomoću tog znanja generira se ulazni test koji uključuje sve koncepte i veze relevantne za određeno područno znanje.

Nakon ulaznog testa slijedi inicijalizacija modela učenika, čije se procijenjeno znanje svrstava, prema Bloomovoj taksonomiji, u jednu od pet kategorija. Model se ponaša poput svih nas, stvara modele ljudi prema dostupnim informacijama i svrstava ih u kategorije skupa karakteristika - stereotipove. Naš model o nekoj osobi je skup uvjerenja opažanjem podataka, od kojih smo neke potpuno zanemarili, a neki su bili okidači za svrstavanje u određeni "koš". Na isti način se gradi model učenika u AC-ware Tutoru. Grubišić navodi kako su okidači (engl. *trigger*) "događaji čija prisutnost označava prikladnost određenog stereotipa" (Grubišić, 2012.), odnosno tip odgovora prepoznaju kao znanje određene razine. Model učenika se također formulira opisom neizvjesnih događaja Bayesovom teorijom vjerojatnosti. Zaključci o varijablama se mogu prikazati grafički probabilističkom mrežom, u ovom slučaju rezultati učenika se njome povezuju s njegovim postupcima i procijenjenom stanju.

Nakon daljnjih testiranja stereotip se modificira ovisno o postignutim rezultatima, a modul učitelja shodno prilagođava nastavni sadržaj. Nastavni sadržaj AC-ware Tutor-a je oblikovan prema hrvatskoj pedagoškoj praksi, prema nastavnim cjelinama, temama i jedinicama (kod učenja), uz odgovarajuće testove pri provjeri znanja. (Grubišić, 2012.) Niži stereotipovi generiraju sadržaj za podučavanje unutar nastavnih jedinica, dok se kod najviših povezuju pojmovi čitave cjeline.

Ovaj rad je orijentiran na grafički prikaz stereotipova. Prva interakcija učenika sa sustavom odvija se ulaznim testom koji sadrži tri ciklusa pitanja. Kako je spomenuto, nakon toga se gradi model tog učenika, odnosno pretpostavlja stereotip koji opisuje njegovo znanje. Slijedi ciklus učenja. Potom se postupak ponavlja, najčešće po tri ciklusa prije novog učenja. U sustavu postoji ukupno pet stereotipova prema razini znanja prikazanih na tablici

1. Inicijalni test odvija se na razini 2 - osrednji, a u idućem ciklusu, ovisno o točnom i netočnom odgovoru generiraju se pitanja iz više odnosno niže težinske kategorije. Naravno, analogno se postupa i za daljnja testiranja. Stereotip prema kojem će se odviti sljedeći ciklus učenja određuje se prema najvećem broju točnih odgovora neke težinske kategorije. Okidači za stereotip najniže razine su više odgovora "Ne znam". U ovom sustavu nisu predviđena pitanja za razinu "0 - novak", ali ako se učenje i testiranje odvija na toj razini stereotipa generiraju se početnička pitanja (razina 1).

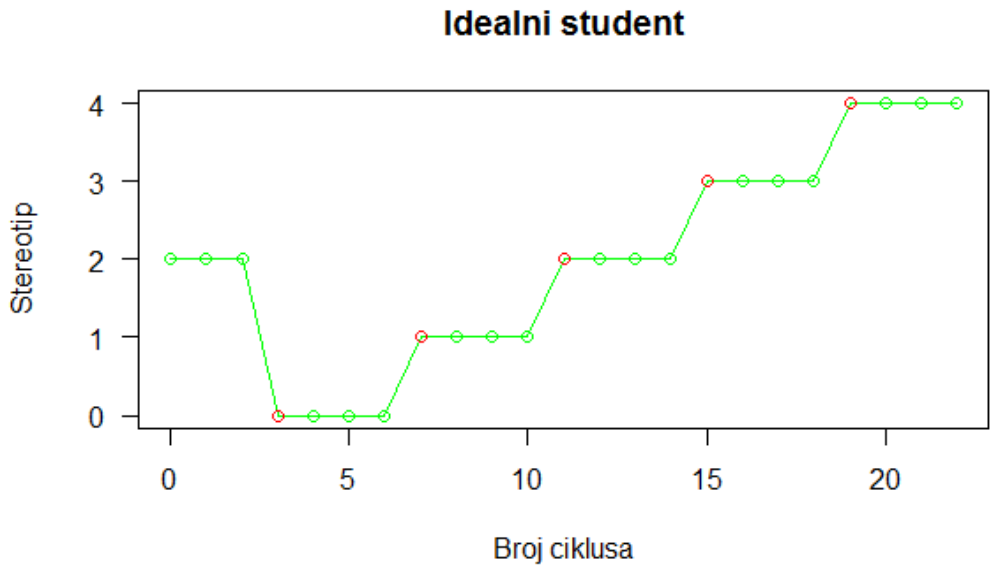
Razina znanja	Stereotip prema znanju	Razina znanja (Bloom)	Bodovi
0	Novak	Nema znanja	(Nema pitanja)
1	Početnik	Reprodukcija	0, 1
2	Osrednji	Razumijevanje	0, 2
3	Napredni	Primjena	0, 3
4	Stručnjak	Analiza, sinteza i vrednovanje	0, 4

Tablica 1. Stereotipovi u sustavu AC-ware Tutor (prema Grubišić, 2012.)

Iako predznanje može pomoći u bržem savladavanju gradiva, učenici-studenti koji pristupaju radu na ovom sustavu ne moraju biti upućeni u problem za uspješno savladavanje gradiva. Stereotip "2 - osrednji" se na početku ulaznog testa dodjeljuje samo radi bolje procjene znanja ispitanika. Kako se ipak radi o sustavu podučavanja, poželjno je znanje studenta obogatiti, odnosno održati kontinuiran rast u razini predstavljenog gradiva nakon ciklusa testiranja.

Iz tog razloga zamislimo studenta koji ne zna baš ništa o području podučavanja. Nakon ulaznog testa, stereotip će biti "novak - 0", biti će mu predstavljeno gradivo na najnižoj razini kroz ciklus učenja i imati će dodatna tri ciklusa testiranja. Nazovimo to jednom skupinom ciklusa. U optimalnom okruženju, svakim ponavljanjem ovog procesa student će prelaziti na višu razinu dok ne utvrdi znanje podučavanog područja na razini "4 - stručnjak". Idealni bi student stoga trebao savladati gradivo i postići razinu stručnjaka kroz pet skupina ciklusa, odnosno ukupno dvadesetak ciklusa učenja i testiranja, računajući i ulazno testiranje. Putanja kretanja njegovog stereotipa se nalazi na slici 3. Crveni kružići

predstavljaju ciklus učenja, dok su zeleni testovi. Vidljiva su tri ciklusa ulaznog testa s osrednjim stereotipom, nakon čega je sustav utvrdio znanje novaka koje se sigurno i bez oscilacija gradilo prema znanju stručnjaka područja podučavanja. Naravno, bit inteligentnih tutorskih sustava je i prilagodba prema vlastitom tempu napredovanja. Kao i u formalnom, unificiranom obrazovanju, jednom učeniku može trebati mnogo više vremena za savladavanje gradiva nego li drugom, ali na koncu mogu biti jednako uspješni. Stoga se proces učenja smatra uspješnim ako se kroz praćenje rada vidi napredak, dok ovaj model samo ilustrativno prikazuje taj trend na najosnovnijem primjeru.



Slika 3. Dijagram kretanja stereotipa za opisani model idealnog studenta

3.2. Jezik R

R je programski jezik prvenstveno namijenjen statističkim izračunima i grafičkom prikazu podataka, najčešće korišten u istraživanju obrade podataka (engl. *data mining*). Nastao je kao implementacija S jezika, razvijenog u Bell Laboratories (Murray Hill, savezna država New Jersey, SAD, tim Johna Chambersa). Jezik omogućuje širok spektar statističkih alata, uključujući i visoku razinu individualizacije grafičkih prikaza podataka. R je besplatna programska podrška otvorenog koda (engl. *open source software*), najviše korišten na UNIX platformama, a verzije postoje i za Windows te MaxOS.

3.2.1. S i R

Komercijalni statistički računalni programi do sredine sedamdesetih nisu postigli značajan uspjeh u istraživačkoj zajednici. Ispisivalo bi se sve što je za problem relevantno, umjesto interakcije s podacima u skladu s tekućim dostignućima obrade podataka. Osim toga, programi su bili veoma ograničeni, a uglavnom ih korisnik nije mogao mijenjati. Za obradu podataka su se često koristili posebno kreirani programi za svaki izračun, u čemu istaknuto mjesto tog vremena ima knjižnica SCS (*Statistical Computing Subroutines*) u programskom jeziku Fortranu (Becker).

Zbog povećane potrebe za statističkim izračunima razvijen je S jezik, 1976. g. u Bell Telephone Laboratories, tada u vlasništvu AT&T kompanije, kori je javno izdan 1976. g. Iako napisan u Fortranu koristeći SCS knjižnicu, sučelje S-a je bilo interaktivno i lakše za korištenje istraživačima koji su manje vješti u programiranju. Osmišljen je kao program koji će uz pomoć funkcija olakšati statističku i analitičku obradu podataka. Osim spomenute Fortranove knjižnice statističkih funkcija, ideja je bila dopustiti korisnicima dodavanje i stvaranje novih funkcija ovisno o potrebama istraživanja. Iako sve ideje nisu implementirane u prvoj verziji, postupno su dodavane u kasnijim inačicama. Usporedno s razvojem operacijskog sustava UNIX, u S-u se dodaju dijelovi napisani u C-u, a 1980. g. izlazi prva javna inačica ovog statističkog jezika. Program postiže svjetski uspjeh, a potpuno je rekonstruiran u C-u 1988. g. kao inačica 3.0, s uključenim prevoditeljem za "staru" verziju. Neke mogućnosti poput makroa su zamijenjene funkcijama i nisu se mogle direktno prevoditi. Suprotno očekivanjima, promjena nije pozitivno prihvaćena jer se prijelaz ipak pokazao "pretežak" za većinu korisnika. Pojavom drugih alata poput

Microsoft Excela za manje statističke izračune, kao i drugih specijaliziranih statističkih programskih paketa široke primjene, S gubi na popularnosti sljedećih godina. U određenim krugovima akademske zajednice ipak se nastavlja koristiti, što je rezultiralo izdavanjem više priručnika za S te nadogradnjama poput S-PLUS i R jezika. Sam S se nije mijenjao od 1998. g., kad je John McKinley Chambers, autor tog jezika objavio posljednji priručnik za S, *"Green Book - Programming with Data: A Guide to the S Language"*. Iste godine je dobio i prestižnu "*Software System Award"*, koju izdaje ACM (*Association for Computing Machinery*, hrv. *Udruženje za računalne strojeve*) [6].

R je moderna implementacija S-a koja uključuje leksičku semantiku preuzetu iz jezika Scope (dijalekt Lisp-a). Iako postoje neke razlike, većina sintakse je vrlo slična, kako navodi Peng (2016.). U razvoju je još od 1992. g., a javna beta verzija objavljena je 2000. g. Otada je postao posebno popularan u krugovima istraživača akademske analitike i obrade obrazovnih podataka (EDM). Neke od istaknutih značajki ovog jezika su, osim definiranja vlastitih funkcija i povezivanje blokova koda iz raznih jezika (C, C++, Fortran, HTML i dr.) s objektima iz R-a. U grafičkom prikazu podataka uključena je podrška matematičkih simbola i izvoz u format LaTex-a. Zajednica programera u R-u stvorila je stoga knjižnice statističkih tehnika specifičnih za mnoge probleme, dostupnih za javno korištenje.

[6] prema Kiš, M., *Englesko-hrvatski i hrvatsko-engleski informatički rječnik*, Zagreb, Naklada Ljevak, 2000.

3.2.2. Istaknuti nedostaci

Kako nijedan sustav nije savršen, tako i R svakako ima ponekih nedostataka. On koristi tehnologiju staru pedesetak godina, otkad je razvijen sustav S.

Često se kao nedostatak R-a navodi pohranjivanje objekata u fizičkoj memoriji (npr. Peng, 2016.). Iako je uistinu R memorijski zahtjevniji od ostalih statističkih paketa, ubrzani razvoj računala doprinio je da se danas većina podataka koji se uobičajeno obrađuju mogu obrađivati na kućnim računalima (tablični podaci do nekoliko milijuna redaka). Također, napravljene su brojne preinake u arhitekturi samog jezika kako bi izvršavanje bilo brže na 64-bitnim sustavima.

R je poprilično težak za učenje, posebno za ne-programere (Muenchen, 2014.), iako nedostatak dokumentacije u posljednje vrijeme nadopunjuje kvalitetna literatura i sama online zajednica, kao i službena podrška. Prema nekim pionirima ovog jezika za njegovo uspješno korištenje doduše nije potrebno ikakvo predznanje u programiranju (Peng, 2016.).

3.2.3. RStudio

RStudio je integrirano programsko okruženje (engl. *integrated development environment*, IDE) za jezik R. Projekt je razvio J. J. Allaire, koji je i današnji vođa RStudio tima. RStudio je pisan u jeziku C++ i koristi Qt framework za grafičko korisničko sučelje (engl. *graphical user interface*, GUI). Relativno je nov alat, prva javna beta verzija je izdana u veljači 2011. g (verzija 0.92). Verzija 1.0 je izdana u studenom 2016. g.

Dostupan je u dvije verzije. RStudio Desktop je osnovna verzija dostupna za Windows, OS X i Linux sustave dok RStudio Server omogućuje pristup web browserom aplikaciji na serverima s najraširenijim Linux distribucijama (Debian, Ubuntu, Red Hat i sl.) RStudio je također dostupan za obe verzije kao open source i komercijalna verzija.

RStudio se sastoji od konzole, preglednika i uređivača teksta (engl. *text editor*) s provjerom i označavanjem sintakse koji je također izravno izvršan, radnog prostora s pregledom varijabli koje koristimo te preglednika povezanih datoteka, dijagrama, paketa i sl. Paketi se odnose na funkcije statističkih tehnika koje dijeli online zajednica, a u najnovijim verzijama se mogu automatski instalirati iz knjižnice pri pokretanju kôda.

3.3. Pisanje kôda u R-u

Naredbe u R-u se mogu izvršavati u konzoli ili se mogu spremati u skripte za kasniji rad na njima. Osnovne skripte napisane u R-u imaju ekstenziju .r, a RStudio uvodi i povezivanje u projekte te korištenja dodatnih mogućnosti u .rmd (*R Markdown*) datotekama.

Pridruživanje vrijednosti varijablama u R-u izvodi se na način:

```
x <- vrijednost
```

gdje je x varijabla kojoj je vrijednost pridružena osnovnim operatorom pridruživanja <-

Pridruživanje je u ovom jednostavnom primjeru podjednako valjano izvesti i operatorom =, kojim se vrijednost pridružuje varijabli u okruženju u kojem se nalazi. Osnovni operator pridruživanja pak uvijek varijablu čini globalnom i ima hijerarhijsku prednost nad operatorom =. Važno je napomenuti kako se u R-u relacijski operator jednakosti označava dvostrukim znakom jednakosti (kao u C-u i srodnim jezicima).

U R-u se razlikuje šest osnovnih tipova podataka (engl. *atomic classes*): *character* (tekst), *numeric* (realni brojevi), *integer* (cijeli brojevi), *complex* (kompleksni broj), *logical* (Booleov tip podataka) i *raw* (sirovi bajtovi podataka).

Podatke tipa *character* referiramo u kodu između navodnika, bilo jednostrukih (`'tekst'`) ili dvostrukih (`"tekst"`) kako bi se razlikovali od ostalih objekata programa (varijabli, funkcija i sl).

Složenije varijable mogu sadržavati nizove podataka. Vektor, kao tipični objekt u R-u (Peng, 2016.) je uređeni niz podataka istog osnovnog tipa, za razliku od liste koja može sadržavati i podatke različitih tipova.

R, kao programski jezik prilagođen obradi podataka može učitavati tablične tekstualne podatke. Uobičajeno se u statistici koriste CSV (engl. *comma-separated value*) datoteke, a njih učitavamo funkcijom `read.csv()`. Tablica podataka (engl. *data frame*) je zapravo lista vektora jednakih duljina, a učitavanjem se može spremiti u varijablu za daljnju manipulaciju. Podskup vektora, matrica ili liste vektora koji odgovara zadanim uvjetima možemo dohvatiti funkcijom:

```
subset(x, uvjet)
```

gdje je x varijabla čiji podskup po uvjetu trebamo dohvatiti.

3.4. Primjena

Cilj je izraditi dijagram kao na slici 3., koji intuitivno dočarava kretanje razine znanja studenta u sustavu AC-ware Tutor. Podaci o radu po ciklusima i cjelinama su dohvaćeni u CSV tablici. Učitamo tablicu podataka i spremamo je u varijablu `podaci`. Izborni argument `header` uzima prvi redak tablice kao imena vektora u odgovarajućim stupcima, a izborni argument `sep = ";"` označava razgodak podataka u CSV tablici, koji je po zadanoj vrijednosti običan zarez, a u našem slučaju tablice sačuvane preko Microsoft Excela generira se kao točka sa zarezom. Varijable `x` i `y` su tekstualni podaci koje korisnik treba sam unijeti ovisno o kojoj osobi treba podatke. S obzirom na varijable `x` i `y` stvaramo podskup varijable `podaci` koji sadrže ime i prezime promatrane osobe. Po tim varijablama pretražujemo vektore `FirstName` i `LastName` u varijabli `podaci` za određivanje podskupa koji sadrži podatke samo promatrane osobe. Spremamo podskup u novu varijablu `osoba`. Slijedi dosad objašnjeni dio kôda.

```
#Komentari se u R-u pišu nakon znaka #
#Ne mogu kao u nekim drugim jezicima prelaziti u novi redak
#Učitavamo tablicu:
podaci <- read.csv("tablica.csv", header = TRUE, sep=";")
#Korisnik unosi ime:
x <- "Ime"
#I prezime:
y <- "Prezime"
#Podaci o toj osobi:
osoba <- subset(podaci, FirstName==x &LastName==y)
```

Varijabla `osoba` je također tablica podataka (engl. *data frame*) koja ima ista imena vektora kao i svi podaci. Treba grafički prikazati kretanje stereotipa tijekom testiranja i učenja te osobe. Osnovna funkcija za crtanje x-y funkcije u R-u je:

```
plot(x, y, ...)
```

gdje su x koordinate točaka apscise, a y ordinate. Ostali argumenti funkcije nisu obvezni, već se dodaju po potrebi. Kao i svi argumenti u R-u, odvajaju se zarezom.

U ovom primjeru potrebno je uzeti sve vrijednosti brojača ciklusa odabrane osobe (`osoba$Counter` predstavlja vektor `Counter` iz varijable `osoba`) te ih povezati s pripadajućim stereotipom (`StereotypeID`).

Najčešće korišten izborni argument `type` opisuje vrstu grafa, a može poprimiti devet različitih vrijednosti, npr.:

`type = "p"`, za iscrtavanje točaka (kružića), dodijeljena vrijednost i ako se ne koristi.

`type = "l"`, za iscrtavanje linija, ili

`type = "o"`, za crtanje kružića preko linija (engl. *overplot*), što je prikladno za naš problem.

Izbornim argumentom `xlab` i `ylab` (prema engl. *label*) na identičan način određujemo naziv koji se na grafu prikazuje uz odgovarajuće osi, a argumentom `main` određujemo glavni naslov dijagrama.

Za naslov je prikladno ispisati ime i prezime promatranog studenta, gdje možemo iskoristiti vrijednosti unesenih varijabli. Tekstualne varijable se u R-u lijepe funkcijom paste:

```
paste(x, y, ..., sep = "")
```

gdje su `x, y, ...` tekstualne varijable koje trebamo zalijepiti, a `sep` određuje proizvoljni tekstualni razmak između njih.

Boja iscrtanog dijagrama može se odabrati parametrom `col` koji poprima vrijednosti naziva boja u engleskom jeziku. S obzirom da je objašnjeno kako postoje ciklusi u kojima se testira znanje, a potom uči na novoj razini - stereotipu zanimljivo ih je grafički istaknuti različitim bojama. U parametar `col` može se učahuriti provjera radi li se o učenju (vektor `IsLearning` u skupu podataka `osoba` jednak 1) ili testiranju (isti vektor jednak nuli). Ovisno o vrijednosti odredimo boju, npr. crvenu i zelenu.

S obzirom kako se vrijednosti oznaka na osima automatski ispunjavaju ovisno o veličini prozora u koji se graf iscrtava, a nama za stereotip trebaju samo cjelobrojne vrijednosti (0-4), iz estetskih razloga možemo osi nacrtati ručno ili ručno odrediti mjesta oznaka na

osima. Osi se ručno crtaju dodavanjem parametra funkciji plot `axes = FALSE`, kako ih ta funkcija ne bi iscrtala automatski. Potom sami određujemo način crtanja na posljednji iscrtani graf, a kod se piše van funkcije `plot()`, gdje se X-os dodaje na posljednji graf s `axis(1)`, a y-os s `axis(2)`. Funkcija `box()` uokviruje čitav graf, a za svaku od funkcija mogući su i dodatni parametri, poput mjesta gdje će se oznake crtati - `at`, ili grafičkog parametra `las`, koji određuje uobičajenu, vertikalnu (1) ili horizontalno polegnutu (0) orijentaciju brojčanih oznaka. Važno je napomenuti kako je ovakav način stilski prilagodljiviji, ali se iscrtava tek nakon grafa pa ga nije moguće fiksirati na točno određen raspon vrijednosti.

Za rješenje problema stoga je odabran način ručnog određivanja oznaka unutar funkcije `plot()` kako bi uvijek imali vidljivo kretanje stereotipa na istom rasponu 0 - 4. Parametrom `ylim` određujemo stroge granice oznaka na y-osi, koje postavljamo na vektor koji sadrži 0 i 4, naše granične vrijednosti stereotipa.

Spomenuti grafički parametar `las` i u funkciji `plot` za y-os ima zadanu vrijednost 0 pa je radi estetike mijenjamo u 1. Slijedi kôd za iscrtavanje dijagrama .

```
#Crtanje grafa, glavni dio:
plot(osoba$Counter, osoba$StereotypeID,
     xlab= "Broj ciklusa", ylab = "Stereotip",
     main = paste(x, y, sep = " "),
     col = ifelse(osoba$IsLearning > 0, "red", "green"),
     type = "o", ylim=c(0, 4), las = 1)
```

Broj bodova koji je student ostvario u procesu testiranja (vidi Tablica 1.) u dostupnim podacima sustava nalazi se u drugoj CSV tablici, s obzirom da se boduje odgovor na svako pitanje. Podatkovna tablica je sličnog formata, s istim imenima stupaca koje koristimo kao vektore imena i prezimena. Analognim postupkom kao i kod podataka iz prve tablice uzimamo podskup promatrane osobe, a funkcijom `mtext()` poslije iscrtavanja grafa dodajemo u podnaslov (izborni parametar `side = 3`) ukupan broj bodova koje je student ostvario. Zbrojimo sve vrijednosti iz vektora `score` funkcijom `sum()`, osim negativne vrijednosti -1 koja predstavlja pitanje koje nije ni bilo ponuđeno studentu. Za kombinirani

ispis teksta i brojke ponovno možemo učahuriti funkciju paste(). Slijedi kôd za dodavanje podnaslova kakav nam treba na iscrtani graf:

```
bodovi <- read.csv("bodovi.csv",header = TRUE,sep=";")

bod <- subset(bodovi,FirstName==x & LastName==y)

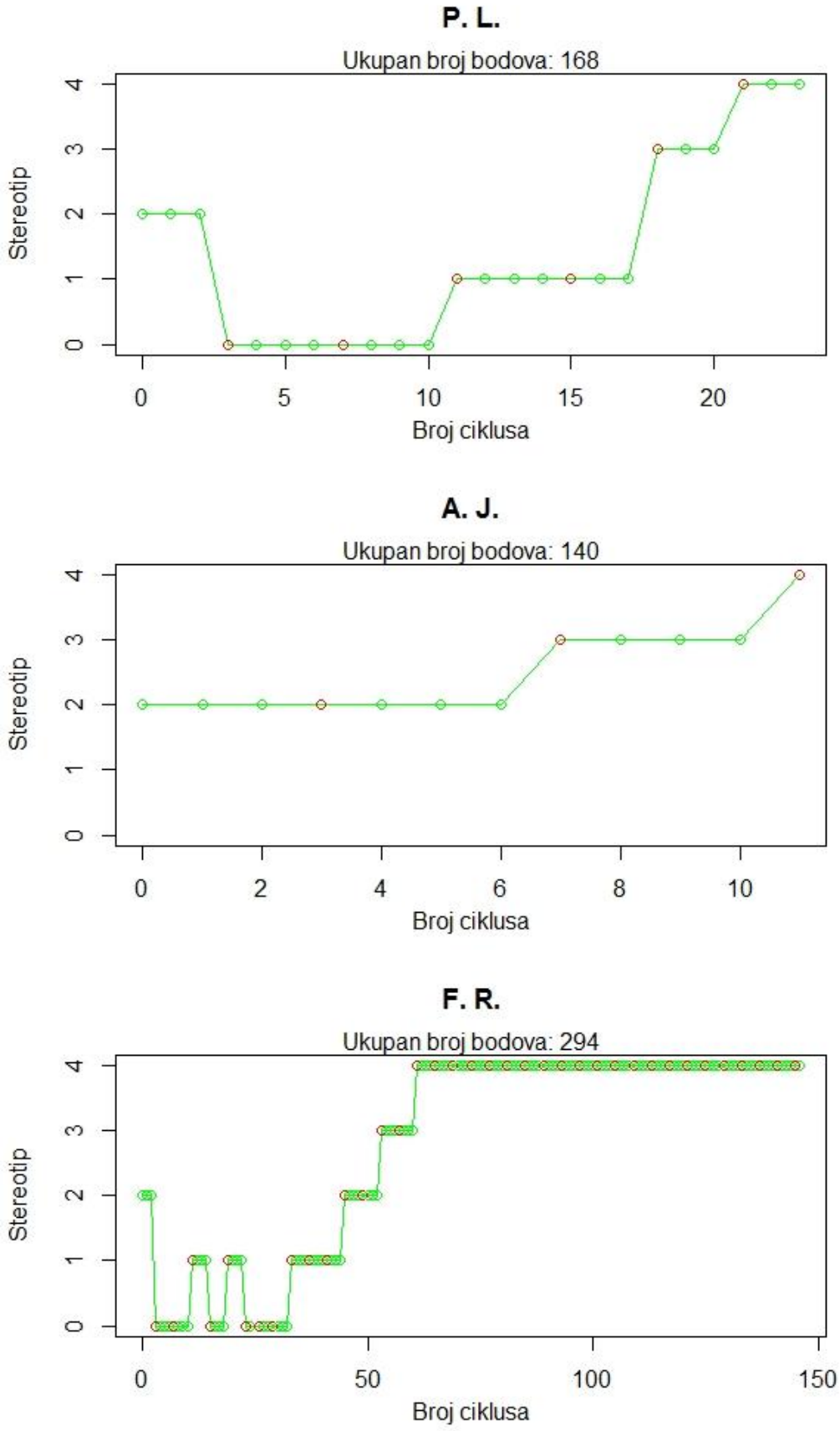
mtext(

paste("Ukupan broj bodova:", sum(bod$score[bod$score>0]),

sep = " "), side = 3 )
```

U RStudiju postoji opcija *Knit document* za izvoz skripte i njenog izlaznog rezultata u formate prikladne za pregled i prezentaciju, poput HTML, PDF i MS Word dokumenata, a uz spomenute mogućnosti ubacivanja blokova kôda HTML-a u skriptu ova opcije je iznimno korisna za oblikovanje prikaza rezultata. U privitku ovog rada nalazi se cjelokupan kôd i izlaz za jednog studenta. Dio studenata u dostupnim podacima je samo isprobao sustav kroz nekoliko ciklusa pa takvi podaci nisu reprezentativni ni vrijedni analize. U nastavku je objašnjeno kretanje procijenjenog znanja za tri odabrana studenta, a rezultat iscrtavanja dijagrama koristeći objašnjeni kôd je prikazan na slici 4.

Student P. L. je prošao 24 ciklusa testiranja (zeleni kružići) i učenja (crveni kružići) u sustavu i postigao najvišu razinu znanja iako je modul stručnjaka nakon ulaznog testa procijenio kako student nije imao nikakvog znanja o područnom gradivu. Student je kroz rad ostvario 168 bodova od ukupnih 348 bodova koji odgovaraju točnim odgovorima svih 87 pitanja četvrte razine. Iako je učenje na dvije najniže razine kod P. L. trajalo nešto duže, može se zaključiti kako je put njegovog učenja blizak modelu idealnog studenta (v. sliku 3.). Vidljiv je skokovit napredak nakon osamnaestog ciklusa, kad je nakon testova procijenjenog znanja stereotipa "1 - početnik" ponuđeno učenje na razini "3 - napredni", što je kasnijim dostizanjem najviše razine i opravdano. Takav napredak se može objasniti razumijevanjem i navikavanjem na način rada sustava ili pak mentalnim procesom bistrenja primljenih informacija koje je dovelo do dubljeg povezivanja i razumijevanja konteksta gradiva, savladavajući apstraktni odnosno tzv. threshold koncept (Eckerdal i sur., 2006.).



Slika 4. Primjeri izlaznih dijagrama za tri studenta

Student A. J. je na sustavu proveo samo 12 ciklusa, s tim da je očito bio upućen u područno gradivo. Već nakon ulaznog testa procijenjena razina znanja je bila "2 - osrednji", a nakon svake skupine ciklusa (skupina od ciklusa učenja i tri testiranja) je rasla do najviše razine. Student je nakon učenja na najvišoj razini odustao od daljnjeg testiranja, vjerojatno smatrajući kako je učenje završeno.

Student F. R. je na sustavu proveo čak 145 ciklusa testiranja i učenja. Njegovo procijenjeno znanje je fluktuiralo između dvije najniže razine prvih četrdesetak ciklusa, dok se nije ustalilo na razini "1 - početnik". Na toj razini student je prošao još tri skupine ciklusa učenja i testiranja, kako je vidljivo na slici 4. Nakon toga F. R. je proveo još po dvije serije ciklusa učenja i testiranja na idućim razinama bez pada u procijenjenom stereotipu da bi nakon ukupno šezdesetak ciklusa postigao najvišu razinu "4 - stručnjak". Na toj razini je proveo još devedesetak ciklusa učenja i testiranja, po čemu je očito kako je riječ o upornom i vrijednom studentu kojeg je sustav dodatno intrigirao za učenje.

Sporiji, ali siguran put napretka ovog studenta pokazuje ne samo kako je sustav uspješan, već i važnost individualizacije obrazovnog programa. Visoka razina individualizacije u klasičnoj učionici je uvjetno rečeno nemoguća, ali inteligentnom tutorskom sustavu najveća snaga upravo leži u višestrukoj mogućnosti individualizacije.

Zaključak

Inteligentni tutorski sustavi se danas koriste u svim okruženjima, prvenstveno kao pomoć pri podučavanju. Ključna ideja ovakvih sustava je visoka razina prilagodbe potrebama učenika za optimalni ishod učenja. S obzirom na vrhunac popularnosti u edukaciji krajem minulog stoljeća, inteligentni tutorski sustavi nisu dostigli dovoljnu razinu autonomnosti za ispunjenje nametnutih očekivanja. Ovakvi sustavi su vrlo kompleksni, zahtjevni i skupi za izradu. Bez obzira, zasigurno se u budućnosti može očekivati njihov daljnji razvoj i češće korištenje kao potpuna zamjena za ljudskog učitelja, posebice u formalnom obrazovanju gdje je nastavu teško ili nemoguće organizirati.

Cilj analitike učenja također je poboljšati ishod učenja proučavanjem dostupnih podataka. Radi se o pedagoškoj akademskoj disciplini u ranoj fazi sistematizacije metodologije, preuzete iz utvrđenih analitičkih procesa u drugim područjima, posebno ekonomije. Povećanjem potrebe za učinkovitijim obrazovnim sustavom posljednjih je godina analitika učenja na temelju prikupljenih podataka o napretku učenika nezamjenjiva pomoć svim sudionicima edukacijskog procesa, a najviše učiteljima i pedagozima u načinu individualizacije i oblikovanju budućeg gradiva.

Ovaj rad orijentiran je na deskriptivno praćenje kretanja stereotipa studenata prema dvjema serijama dostupnih podataka inteligentnog tutorskog sustava AC-ware Tutor, koji se temelji na prilagođavanju gradiva prema procjeni stereotipa, kao razine trenutačnog znanja učenika.

Gotovo svi promatrani studenti koji su prošli više od tridesetak ciklusa dostigli su najvišu razinu znanja, neovisno o predznanju. Primijećene su značajne individualne razlike u brzini savladavanja gradiva, što najbolje ilustrira važnost prilagodljivosti sustava učeniku. Putanja učenja razmatranih studenata je uglavnom bila isključivo uzlazna, što odgovara modelu idealnog studenta raspravljenom na početku treće cjeline ovog rada. Također, kod većine studenata primijećen je skokovit prijelaz s nižih na više stereotipove, odnosno "Eureka efekt", koji se može objasniti shvaćanjem načina rada sustava, ali i stvaranjem vlastitog mentalnog konstrukta veza među pojmovima koji odgovara područnom znanju. U metodici informatike poznati su tzv. threshold koncepti koji su teški za razumijevanje, a ključni za čitavo područje na koje se odnose. Učenici moraju uložiti značajan trud za potpuno razumijevanje apstraktnog pojma s obzirom na dugotrajnost procesa mijenjanja

vlastitog mentalnog konstrukta koji bi i dalje podržao sve veze među već utvrđenim konceptima. Jednom shvaćeni, threshold koncepti se podrazumijevaju u širokom kontekstu, što se manifestira kao skokovit pomak u znanju.

Za analizu uzroka primijećenih fenomena, kao i za detaljnije zaključke, potrebna su daljnja istraživanja. Primjerice, moglo bi se pratiti koliko je vremena učenik proveo na svakom skupu pitanja (ciklusu). Osim shvaćanja kognitivnih procesa, to bi olakšalo i bistrenje podataka za obradu, s obzirom kako su neki studenti samo isprobali rad na sustavu nasumično i brzo odgovarajući na pitanja te nisu valjani uzorci bilo kakve analize. Iz didaktičkog aspekta može se proučiti jesu li studentima neki koncepti bili značajno teži od ostalih, kako bi se u budućnosti bolje oblikovao modul učitelja te olakšalo stjecanje znanja kao cjeline.

Literatura

[1] ANDERSON, J. R., "*The Expert Module*", dio zbornika "*Foundations of Intelligent Tutoring Systems*", Lawrence Erlbaum Association Inc., 1988.

[2] ARNOLD, K. E., "*Signals: Applying Academic Analytics*", objavljeno u "*Educause Quarterly 33*", 2010.

[3] AVELLA, J. T. i sur., "*Learning Analytics Methods, Benefits, and Challenges in Higher Education: A Systematic Literature Review*", University of Phoenix, SAD, 2016.

[4] BAKER, R. S., *"Stupid Tutoring Systems, Intelligent Humans"*, Teachers College, Columbia University, New York, USA, 2016.

[5] BAKER, R. S., SIEMENS, G., *"Educational Data Mining and Learning Analytics"*, Teachers College - Columbia University, Athabasca University, SAD, 2013.

[6] BECKER, R. A., *"A Brief History of S"*, AT&T Bell Laboratories, Murray Hill, New Jersey, SAD, ?

[7] BEN-ARI, M., "*Constructivism in Computer Science Education*", Weizmann Institure of Science, Israel, 1998.

[8] BIENKOWSKI, M. i sur., "*Enhancing Teaching and Learning Through Educational Data Mining and Learning Analytics: An Issue Brief*", U.S. Department of Education, SAD, 2012.

[9] BLOOM, B. S., "*The Two-Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring*", članak iz zbornika "*Educational Researcher*", br. 13, 1984.

[10] CAMPBELL, J. P. i sur., "*Academic Analytics - A New Tool for A New Era*", Educause, Washington, SAD, 2007.

[11] CAMPBELL, J. P., OBLINGER, D. G., *"Academic Analytics"*, Educause, Washington, SAD, 2007.

[12] CLANCEY,W.J., "*Knowledge-Based Tutoring: The GUIDON Program.*", MIT Press, Cambridge, MA, SAD, 1987.

[13] CORBETT A. T., ANDERSON, J. R., "*LISP intelligent tutoring system: Research in skill acquisition*", 1989.

[14] CORBETT A. T., KOEDINGER, K. R., I ANDERSON, J. R. "*Intelligent tutoring systems*", članak iz zbornika: "*Handbook of human-computer interaction*", Amsterdam: Elsevier., Nizozemska, 1997.

[15] CORBETT, A. T., ANDERSON, J. R., "*Knowledge Tracing: Modeling the Acquisition of Procedural Knowledge. User Modeling and User-Adapted Interaction*", 1995.

[16] CROWDER, N. A., "*Automatic tutoring by means of intrinsic programming*", dio zbornika "*Automatic teaching: The state of the art*", Wiley, 1959.

[17] DEVENS, R. M., HAMILTON, S., *"Cyclopædia of Commercial and Business Anecdotes"*, D. Appleton and Co., New York, SAD, 1865.

[18] DRON, J., ANDERSON, T., "*On the design of collective applications"*, iz zbornika zaključaka međunarodne konferencije računarstva ICCSE, 2009.

[19] ECKERDAL, A. i sur., "*Putting Threshold Concepts into Context in Computer Science Education*", 2006.

[20] ELIAS, T., "*Learning Analytics: Definitions, Pocesses and Potential*", Athabasca University (?), SAD, 2011.

[21] GOLDSTEIN, P. J., KATZ, R. N., "*Academic Analytics: The Uses of Management Information and Technology in Higher Education*", Educause, Washington, SAD, 2005.

[22] GROFF, J. S., "*Personalized Learning: The State of the Field & Future Direction*", Center for Curriculum Redesign, 2017.

[23] GRUBIŠIĆ, A., "*Model prilagodljivoga stjecanja znanja učenika u sustavima e-učenja*", doktorska disertacija, Sveučilište u Zagrebu, veljača 2012.

[24] GRUBIŠIĆ, A., "*Vrednovanje učinka inteligentnih sustava e-učenja*", magistarski rad, Sveučilište u Zagrebu, 2007.

[25] HARTLEY, J. R., SLEEMAN, D., "Towards more intelligent teaching systems", objavljeno u "*International Journal of Human-computer Studies*" i "*International Journal of Man-machine Studies*", 1973.

[26] LARKIN, J., CHABAY, R., "*Computer Assisted Instruction and Intelligent Tutoring Systems: Shared Goals and Complementary Approaches*", Hillsdale, New Jersey: Lawrence Erlbaum Associates, 1992.

[27] LAWSON, D. R., "*Who thought of it first*", članak iz zbornika "*Using Programmed Instruction*", Educational Technology publications, 1973.

[28] MAYER, R.E. I MORENO R., "*A Split-Attention Effect in Multimedia Learning: Evidence for Dual Processing Systems in Working Memory*", dio zbornika "*Journal of Educational Psychology Vol. 90-2*", 1998.

[29] MUENCHEN, R. A., "*Why R is hard to learn*", 2014. http://r4stats.com/articles/why-r-is-hard-to-learn

[30] NKAMBOU, R. i sur., "*What are Intelligent Tutoring Systems*", dio zbornika "*Advances in Intelligent Tutoring Systems*", 2010.

[31] NWANA, H. S., "*Intelligent Tutoring Systems: an overview*", University of Liverpool, dio zbornika "Artificial Intelligence Review", 1990.

[32] PANETTA, K., "*Top Trends in the Gartner Hype Cycle for Emerging Technologies*", 2017. http://www.gartner.com/smarterwithgartner/top-trends-in-the-gartner-hype-cycle-for-emerging-technologies-2017

[33] PAPAMITSIOU, Z., ECONOMIDES, A., "*Learning Analytics and Educational Data Mining in Practice: A Systematic Literature Review of Empirical Evidence*", objavljeno u "*Educational Technology & Society, 17, 49. - 64.*", 2014.

[34] PAVLIK, P. I., CEN, H., KOEDINGER, K. R., *"Performance Factors Analysis, A New Alternative to Knowledge Tracing"*, iz zbornika zaključaka međunarodne konferencije o umjetnoj inteligenciji u obrazovanju, AIED, 2009.

[35] PENG, R. D., "*R programming for data science*", Leanpub, 2016.

[36] PRESSEY, S. L., "*A third and fourth contribution toward the coming "industrial revolution" in education*", objavljeno u "*School and Society 36*", 1932.

[37] ROMERO, C., VENTURA, S., "*Data Mining in Education. WIREs Data Mining and Knowledge Discovery*", John Wiley & Sons., 2013.

[38] ROSIĆ, GLAVINIĆ, STANKOV, "*Conceptualization of Intelligent Tutoring Systems*", dio zbornika "*Intelligent Learning Infrastructure for Knowledge Intensive Organizations: A Semantic Web Perspective*", 2006.

[39] SHORE, K., "*The Overly Dependent Student*", Education World, http://www.educationworld.com/a_curr/shore/shore047.shtml

[40] SHUTE, V. J., PSOTKA, J., "*Intelligent Tutoring Systems: Past, Present, and Future*", Human resources directorate manpower and personnel research division, 1994.

[41] STANKOV, S., "*Inteligentni tutorski sustavi – teorija i primjena*", Sveučilište u Splitu, listopad 2010.

[42] VERZANI, J., "*Simple R*", College of Staten Island, 2002.

[43] WENGER, E., "*Artificial intelligence and tutoring systems*", Morgan Kaufmann, Los Altos, Kalifornija, SAD, 1987.

[44] WOOLLEY, D. R., *"PLATO: The Emergence of Online Community*", 1994. http://thinkofit.com/plato/dwplato.htm

Napomena: Sve web lokacije su ponovno provjerene i dostupne u rujnu 2017.

Sažetak

Ovaj diplomski rad započinje pregledom razvoja, ideja i nekih ostvarenih inteligentnih tutorskih sustava, nakon čega je detaljno opisana njihova danas široko prihvaćena arhitektura u vidu četiri modula. Nastavlja se pogledom na analitiku učenja iz perspektive vodećih istraživača tog područja, danas neizostavne pedagoške znanstvene discipline te se zaključuje o njenoj primjeni u inteligentnim tutorskim sustavima, ističući povezanost ova dva područja istraživanja. Predstavljene su osnovne zamisli inteligentnog tutorskog sustava AC-ware Tutor, koji posebnu pažnju pridaje modulu učenika i koristi se na splitskom Prirodoslovno-matematičkom fakultetu od 2012. g. Podaci o radu studenata iskorišteni su za prikaz podataka o napretku za vrijeme njihovog učenja. Za obradu se koristi programskom paketu RStudio, integriranom programskom okruženju za jezik R, najpopularnijeg alata u analitici učenja, čije su osnove također metodički objašnjene kroz konkretne primjere, a dane su i idejne smjernice za buduća istraživanja ovog sustava u analitici učenja.

Ključne riječi:

e-učenje, inteligentni tutorski sustav, analitika učenja, obrada podataka, prilagodljivi računalom oblikovani nastavni sadržaj, stereotip, jezik R, RStudio

Summary

This thesis starts with the overview of development, ideas and evaluation of intelligent tutoring systems, after which widely accepted architecture of its four modules is described in detail. Paper is continued with a synthesis of learning analytics from its prominent researcher's point of view, concluding with application in intelligent tutoring systems while highlighting synergy of these two academic disciplines. Basic ideas of intelligent tutoring system AC-ware Tutor are presented. This system is focused on student module and came into use in 2012 on Faculty of science, University of Split. Student data from this system were logged and now used for representation of their learning progress. Data was processed in RStudio, an integrated development environment for R language, the most popular tool in learning analytics. Its basics are also methodically explained through the examples. Some ideas for direction of future research on this system in learning analytics are also given.

Keywords:

e-learning, intelligent tutoring system, learning analytics, data mining, adaptive courseware, stereotype, R language, RStudio

Skraćenice

ACM	Association for Computing Machinery	Udruženje za računalne strojeve
ACT	*Assumptions and related principles for a Computer-implemented Tutor*	
ATN	*Augmented Transition Network*	Proširena mreža prijelaza
BKT	*Bayesian Knowledge Tracing*	
CAI	*Computer-Assisted Instruction*	Nastava pomoću računala
CSV	*Comma-Separated Value*	
DMN	*Discourse Manangement Network*	Mreža upravljanja raščlanjivanja
EDM	*Educational Data Mining*	Obrada obrazovnih podataka
GUI	Graphical User Interface	Grafičko korisničko sučelje
ICAI	*Intelligent Computer-Assisted Instruction*	Intelig. nastava pomoću računala
IDE	Integrated Development Environment	integrirano programsko okruženje
ITS	*Intelligent Tutoring System*	Inteligentni tutorski sustav
PFA	*Performance Factors Analysis*	
SCS	*Statistical Computing Subroutines*	

Privitak

Slijedi R skripta za objašnjeno iscrtavanje dijagrama iz podataka dostupnih u dvije CSV tablice, kao izvješće kreiran opcijom K*nit document* u programskom paketu RStudio.

```
#Ucitavamo tablicne podatke:
podaci <- read.csv("tablica.csv", header = TRUE, sep=";")
bodovi <- read.csv("bodovi.csv", header = TRUE, sep=";")

#Unosimo ime i prezime:
x <- "Pxxx"
y <- "Lxxx"

#Podskup podataka o jednoj osobi:
osoba <- subset(podaci, FirstName==x & LastName==y)
bod <- subset(bodovi, FirstName==x & LastName==y)

#Iscrtavanje dijagrama:
plot(osoba$Counter, osoba$StereotypeID, xlab = "Broj ciklusa",
      ylab="Stereotip", main = "P. L.", #main = paste(x, y, sep = " "),
      col = ifelse(osoba$IsLearning > 0,"red","green"),
      type = "o", ylim=c(0,4), las = 0)

#Ispis podnaslova - bodovi:
mtext(paste("Ukupan broj bodova:", sum(bod$score[bod$score>0]), sep = "
"), side = 3)
```

