



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Ani Grubišić

**MODEL PRILAGODLJIVOGA STJECANJA  
ZNANJA UČENIKA U SUSTAVIMA E-UČENJA**

DOKTORSKI RAD

Zagreb, 2012



UNIVERSITY OF ZAGREB  
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ani Grubišić

**ADAPTIVE STUDENT'S KNOWLEDGE  
ACQUISITION MODEL IN E-LEARNING SYSTEMS**

DOCTORAL THESIS

Zagreb, 2012

---

Doktorski rad je izrađen na Sveučilištu u Zagrebu, Fakultetu elektrotehnike i računarstva, Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave.

Dio istraživanja je obavljen na Prirodoslovno-matematički fakultet, Sveučilište u Splitu, te na Fakultetu prirodoslovno-matematičkih i odgojnih znanosti, Sveučilište u Mostaru.

Mentor: prof.dr.sc. Slavomir Stankov (Prirodoslovno-matematički fakultet, Sveučilište u Splitu)

Doktorski rad ima: 279 stranica

Doktorski rad br.: \_\_\_\_\_

---

---

Povjerenstvo za ocjenu i obranu doktorske disertacije:

1. Dr.sc. Vlado Glavinić, redoviti profesor, Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva
2. Akademik dr.sc. Leo Budin, redoviti profesor emeritus (u mirovini), Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva
3. Dr.sc. Robert Manger, izvanredni profesor, Sveučilište u Zagrebu Prirodoslovno-matematički fakultet

Datum obrane disertacije: 24. veljače 2012.

---

---

*Za Staru Baku*

---

---

## Zahvala

Željela bih se zahvaliti svima koji su pomogli u nastanku ove doktorske disertacije, ali i na različite načine oblikovali moj život proteklih godina. Sada, kada je cijeli ovaj „posao“ završen, riječi zahvale zvuče mi prazno u usporedbi s onim što osjećam. Nadam se, da ću vam svima, svoju potpunu zahvalnost izraziti u godinama koje slijede.

Na prvome mjestu, zahvalnost za izradu ove disertacije, ali i za moj cjelokupni profesionalni razvoj, dugujem svom mentoru, prof.dr.sc. Slavomiru Stankovu. Od dana kad mi je odlučio biti mentor na diplomskom radu i od kada mi je ponudio posao na Zavodu za informatiku, prof. Stankov je pokazivao iznimnu vjeru u mene. U radu mi je omogućio samostalnost i kreativnu slobodu, a svojim savjetima uspio me usmjeriti na pravi put. Njegovi konkretni stručni savjeti i sposobnost da identificira problem, uvijek su mi bili velika pomoć u radu. U njemu sam uvijek imala osobu kojoj sam se mogao obratiti u vezi s bilo kojim problemom, znajući da ću dobiti pomoć. Profesore, velika Vam hvala na svemu!

Docentu dr.sc. Branku Žitku, kolegi s kojim surađujem od kada sam se zaposlila i mom "supatniku" s poslijediplomskog studija, dugujem veliku zahvalnost na stručnoj pomoći tijekom izrade ove doktorske disertacije. On je "čovjek koji zna znanje" i ovim putem mu se srdačno zahvaljujem.

Hvala mojoj obitelji i prijateljima na moralnoj podršci i razumijevanju, hvala što su tu uz mene, usprkos tome što sam ih često zapostavljala zbog posla. Žao mi je zbog toga i nastojat ću to ispraviti.

Naposljetku, najveću zahvalnost ipak dugujem mom Fabjanu i našim anđelima Mislavu i Rozi koji su najbolje iskusili sve muke mog profesionalnog razvoja. Hvala vam što ste me uvijek podržavali i vjerovali u mene. Bez vas ne bi bilo ove disertacije, niti bilo kojeg mog uspjeha. Hvala na strpljivom podnošenju zanemarivanja, ohrabrenju koje ste mi je pružali kad bih posustala i radovanju svakom mom uspjehu.

---

---

## Sažetak

Model automatskog i dinamičkog generiranja prilagodljivog računalom oblikovanog nastavnog sadržaja u sustavima e-učenja, AC-ware Tutor, predstavlja sinergiju tradicionalne strukture inteligentnih tutorskih sustava, stereotipova, Bloomove taksonomije znanja i Bayesovih mreža. U današnje vrijeme, kada se najviše uočava problematika prilagođavanja znanju učenika, ovakav model donosi rješenja koja olakšavaju učiteljima dugotrajan i iscrpljujući proces oblikovanja nastavnih sadržaja i testova. Naime, automatizam u generiranju elemenata računalom oblikovanog nastavnog sadržaja prebacuje spomenuti zadatak s živih učitelja na računalnog tutora, koji samostalno oblikuje, odabire i niže elemente računalom oblikovanog nastavnog sadržaja potpuno prilagođene razini znanja učenika. Razina znanja učenika je određena njegovim stereotipom, a profinjena je Bayesovim modelom učenika. Ovakav računalom oblikovani nastavni sadržaj nije statičan, već se dinamički mijenja nakon svakog provjeravanja znanja, prilagođavajući sljedeći element courseware-a za učenje aktualnoj razini znanja učenika. Učenik se uči i poučava onoliko dugo koliko mu je potrebno da postigne željenu razinu znanja. Poželjno je da svaki učenik završi cijeli proces učenja i poučavanja na razini stručnjaka, ali kako to nije ostvarivo ni u okruženju tradicionalne nastave, tako se i u ovom pristupu omogućava učeniku završavanje procesa učenja i poučavanja na nižim razinama znanja. Još jednom naglašavamo da model AC-ware Tutor prilagođava sadržaj, a ne sučelje, znanju učenika, a ne stilovima učenja, što ga čini jedinstvenim primjerkom među prilagodljivim sustavima e-učenja.

### Ključne riječi:

Inteligentni tutorski sustav, prilagodljivi računalom oblikovani nastavni sadržaj, automatsko generiranje pitanja, Bloomova taksonomija znanja, Bayesova mreža predikcija, stereotip

---

---

## Abstract

A model of automatic and dynamic generation of adaptive courseware in e-learning systems, AC-ware Tutor, represents a synergy between traditional intelligent tutoring systems structure, stereotypes, Bloom's knowledge taxonomy and Bayesian networks. Nowadays, when student's knowledge adaptation problematic emerges the most, this model brings solutions to teachers easing long and exhausting process of designing courseware and tests. Namely, automatism in generation of courseware elements is, in this model, transferred from teachers to computer tutor, who designs, selects and sequences courseware elements adapted to student's knowledge level. Student's knowledge level is determined by his or her stereotype, and refined by Bayesian student model. Courseware designed in this model is not static, but it changes dynamically after every knowledge test, adapting next courseware element to current student's knowledge level. Student is involved in learning and teaching process as long as it is necessary for him or her to acquire wanted knowledge level. It is desirable that each and every student finishes learning and teaching process as expert, but as it is not achievable in traditional classroom, this model enables student to finish learning and teaching process at lower knowledge levels. We emphasise once more that AC-ware Tutor model adapts content, not interface, to student's knowledge, not learning styles, what makes it unique exemplar among adaptive e-learning systems.

### Key words:

Intelligent tutoring systems, adaptive courseware, automatic question generation, Bloom's knowledge taxonomy, Bayesian probability network, stereotype

---



---

## Prošireni sažetak

S obzirom da je ovoj doktorskoj disertaciji prethodilo dugogodišnje istraživanje o učinkovitosti sustava e-učenja, a posebice inteligentnih tutorskih sustava, logično se nametnula ideja o definiranju modela prilagodljivog stjecanja znanja učenika kojim bi se povećala učinkovitost spomenutih sustava. Stoga su u ovoj doktorskoj disertaciji objašnjene najvažnije značajke pristupa realizacije prilagodljivosti u sustavima e-učenja.

Učeničko znanje, kao glavni razlog postojanja sustava e-učenja, ovisno je o područnom znanju te se razvija tijekom procesa učenja i poučavanja. Upravo zato je nemoguće, na temelju usvojenosti jednog područnog znanja, pouzdano procijeniti učenikove performanse na drugom područnom znanju, ali je to moguće napraviti unutar istog područnog znanja. Upravo se na ovoj činjenici temelji prilagodljivost koja je realizirana u novom modelu prilagodljivog stjecanja znanja u sustavima e-učenja. Uzimajući u obzir aktualnu razinu znanja učenika, prilagođava se sadržaj, korištenjem Bloomove taksonomije znanja, Bayesovih mreža i stereotipova učenika.

Naš pristup ispunjava sve zahtjeve koji se postavljaju pred realizaciju prilagođavanja: svaki učenik uči svojim ritmom; prilagođavanje se događa često; svaki učenik može uspješno završiti proces učenja i poučavanja; kada je nešto uspješno naučio, učenik nastavlja dalje; učenika se ne poučava ono što već zna.

Novi model automatskog i dinamičkog generiranja i prilagodljivog odabira, nizanja i prezentiranja računalom oblikovanog nastavnog sadržaja u inteligentnim tutorskim sustavima uvažava aktualnu razinu znanja učenika i njegove kognitivne značajke koje određuju složenost, odnosno razinu elemenata računalom oblikovanog nastavnog sadržaja koji će se učeniku prezentirati. Automatsko generiranje računalom oblikovanog nastavnog sadržaja u novom modelu označava da elemente za učenje i provjeravanje znanja stvara sam sustav na temelju ontologije područnog znanja. Dinamičko generiranje računalom oblikovanog nastavnog sadržaja označava da se on stvara u trenutku izvođenja. Prilagodljivi odabir, nizanje i prezentiranje se odvija automatski i dinamički u skladu s modelom učenika (stereotip i Bayes) uporabom predložaka za tvrdnje.

Prednost našeg pristupa je u tome što se računalom oblikovani nastavni sadržaj prilagođava učeniku prema njegovom znanju, a ne prema stilu učenja. Također prilagođava se sam sadržaj, a ne sučelje, što je karakteristika za sustave prilagodljive hipermedije. Sustavi prilagodljive hipermedije, kao najpoznatiji predstavnici sustava koji se prilagođavaju učeniku, temelje svoj mehanizam za prilagođavanje na statičkim sadržajima kojima je najmanji element raspoloživ za prilagođavanje tek dio stranice. Prilagođavanje ne zadire u mijenjanje

---

---

samog sadržaja stranica, već može jedino prikazati ili ne prikazati neke njene dijelove. Naime, sustavi prilagodljive hipermedije ne mogu sami stvarati nove stranice, već samo mogu na stranicama određene dijelove prikazati ili ne prikazati učeniku. Dakle, ono što prikazuju (dio stranice) uvijek ima istu formu i sadržaj koji ovisi o autoru stranice.

Tvrdimo da prilagođavanje opisano u novom modelu pridonosi boljem stjecanju znanja jer je najmanji element raspoloživ za prilagođavanje koncept znanja, dakle, „atomska čestica“ znanja i u tom smislu nedjeljiva. Uspješnost prilagodljivosti je obrnuto proporcionalna najmanjem elementu raspoloživom za prilagođavanje. Pošto smo se u novom modelu dotakli najmanjeg elementa za prilagođavanje, smatramo da našim pristupom realiziramo maksimalnu prilagodljivost.

Pošto ovaj model prilagodljivog stjecanja znanja zadire u sve komponente inteligentnog tutorskog sustava, objasnili smo na koji način smo modificirali spomenute komponente tako da inteligentni tutorski sustav bude prilagodljiv učeniku prema njegovom znanju. Zbog toga smo u modul stručnjaka, temeljen na formalnom ontološkom opisu područnog znanja, definirali reprezentaciju područnog znanja, u modulu učenika, temeljenom na Bayesovim mrežama predikcije znanja, smo definirali stereotipove prema Bloomovoj taksonomiji znanja, a modulu učitelja smo definirali predloške tvrdnji i pitanja temeljem Bloomove taksonomije znanja prilagođenoj hrvatskoj pedagoškoj praksi. Modul komunikacije u ovom modelu ne definiramo jer se ovakav pristup može realizirati u bilo kojem ITS-u koji ima znanje zasnovano nad ontologijom, neovisno o načinu komunikacije između učenika i sustava.

Navedena tri modula su u interakciji da bi se realizirala prilagodljivost na različitim aspektima procesa učenja i poučavanja: prilagođava se struktura računalom oblikovanog nastavnog sadržaja (odabir i nizanje) na temelju učenikova stereotipa, prilagođava se prezentiranje na temelju Bayesove mreže i stereotipa (predlošci za tvrdnje), prilagođava se provjeravanje znanja učenika na temelju njegovog stereotipa (predlošci za pitanja).

Prikazali smo detaljno strukturu novog modela prilagodljivog stjecanja znanja Adaptive Courseware Tutor Model (AC-ware Tutor). Za svaku komponentu sustava navodimo teorijsku podlogu nad kojom se ona zasniva, kao i formalizaciju koja opisuje specifičnosti našeg pristupa. Zamisao predloženog modela se dekomponira do razine modula i obuhvaća: modul stručnjaka (područno znanje), modul učitelja (računalom oblikovanog nastavnog sadržaja) i modul učenika (modeliranje učenika). Detaljno smo objasnili proces generiranja elemenata računalom oblikovanog nastavnog sadržaja, odabira i nizanja, te prezentiranja elemenata, pojedinačno kroz komponente sustava.

---

---

## Extended abstract

Given that this doctoral dissertation was preceded by many years of research on the effectiveness of e-learning systems, especially intelligent tutoring systems, it is logical to impose the idea of defining an adaptive student's knowledge acquisition model that would increase the effectiveness of these systems. Therefore, in this doctoral dissertation, the most important features of an adaptivity realization approaches are explained.

Student's knowledge, as the main reason for the existence of e-learning systems, depends on the domain knowledge and develops during the learning and teaching process. That is why it is impossible, based on acquisition of one domain knowledge, to reliably estimate student's performance on another domain knowledge, but is it possible within the same domain knowledge. Adaptivity realized in adaptive student's knowledge acquisition model in the e-learning systems is based on this fact. Taking into account the current student's knowledge level, the content is adapted, using Bloom's knowledge taxonomy, Bayesian networks and student stereotypes.

Our approach meets all the demands that are placed before the adaptation realization: each student learns on his or her rhythm; adaptation happens frequently; each student can successfully complete the teaching and learning process; when students has learned something successfully, the student continues with learning and teaching process; students are not taught what they already know.

A new model of automatic and dynamic generation and adaptive selection, sequencing and presentation of courseware in intelligent tutoring systems, takes into account the current student knowledge level and his or her cognitive characteristics that determine the complexity or level of courseware element that will be presented to the student. Automatic generation of courseware in the new model indicates that the courseware elements for learning and testing knowledge are created by model itself, based on the domain knowledge ontology. Dynamic generation of courseware indicates that the courseware is created in the moment of execution. The adaptive selection, sequencing and presentation of the courseware is done automatically and dynamically in accordance with a student model (stereotype and Bayesian) using the templates for statements.

The advantage of our approach is that the courseware is adapted to the student's knowledge, not the learning style. To the content itself is adapted, not the interface, which is characteristic of adaptive hypermedia. Adaptive hypermedia systems, as the best known representatives of adaptive e-learning systems, base their adaptation mechanism on the static content whose smallest adaptation element is just a page part. Adaptation does not

---

---

change the contents of the page, but can only show or hide some of its parts. Namely, the adaptive hypermedia systems cannot create new pages themselves, but can only display or not display certain parts of the pages to the student. So, what they show (a part of the page) has the same form and content which depends on the page author.

We claim that the adaptation process described in the new model contributes to a better knowledge acquisition because its smallest element available for adaptation is the knowledge concept of, ie, "atomic particle" of knowledge and in this sense inseparable. The success of adaptivity is inversely proportional to the smallest element available for adaptation. As, in the new model, we have reached the smallest element for adaptation, we believe that our approach realizes maximum adaptivity.

Since this adaptive student's knowledge acquisition model interferes with all components of an intelligent tutoring system, we have explained how we have modified these components so that an intelligent tutoring system is adaptive to the student's knowledge. Therefore, in the expert module, based on a formal ontological description of domain knowledge, we have defined a representation of domain knowledge, in the student module, based on Bayesian networks, we have defined stereotypes according to Bloom's knowledge taxonomy, in the teacher module, we have defined templates for statements and questions based on Bloom's knowledge taxonomy adapted to the Croatian pedagogic practice. The communication module in this model is not defined because presented adaptation approach can be realized in any intelligent tutoring system which has ontology-based knowledge, regardless of the manner of communication between students and the system.

The above three modules have to interact in order to realize the adaptivity in different aspects of learning and teaching process: the structure of the courseware (selecting and sequencing) is adapted on the basis of student stereotype, the presentation of courseware is adapted based on Bayesian network and stereotypes (templates for statements), the student's knowledge testing is adapted on the basis of student stereotype (templates for questions).

We have shown in detail the structure of the new adaptive student's knowledge acquisition model Adaptive Courseware Tutor Model (AC-ware Tutor). For each component of the system we describe its theoretical background, as well as the formalization that describes the specifics of our approach. The idea of the proposed model is decomposed into modules: the expert module (domain knowledge), the teacher module (courseware) and student module (student modeling). We thoroughly explain the process of generating courseware elements, the selection and sequencing of courseware elements, the presentation of courseware elements, throughout each system component.

---

# Sadržaj

<b>1</b>	<b>UVOD.....</b>	<b>1</b>
<b>2</b>	<b>ZAMISAO MODELA ZA PRILAGODLJIVO STJECANJE ZNANJA U SUSTAVIMA E-UČENJA.....</b>	<b>5</b>
2.1	MOTIVACIJA .....	7
2.2	PRILAGODLJIVOST U SUSTAVIMA E-UČENJA .....	9
2.3	ODREDNICE MODELA .....	12
2.4	ELEMENTI STRUKTURE .....	16
2.4.1	<i>Modul stručnjaka.....</i>	<i>16</i>
2.4.1.1	Teorijska podloga modula stručnjaka.....	16
2.4.1.2	Formalizacija modula stručnjaka u sustavu <i>AC-ware Tutor</i> .....	17
2.4.2	<i>Modul učenika .....</i>	<i>22</i>
2.4.2.1	Teorijska podloga modula učenika.....	23
2.4.2.2	Formalizacija modula učenika u sustavu <i>AC-ware Tutor</i> .....	32
2.4.3	<i>Modul učitelja.....</i>	<i>34</i>
2.4.3.1	Teorijska podloga modula učitelja.....	34
2.4.3.2	Formalizacija modula učitelja u sustavu <i>AC-ware Tutor</i> .....	35
<b>3</b>	<b>STANJE ISTRAŽENOSTI .....</b>	<b>50</b>
3.1	RAZVOJ SUSTAVA ZA GENERIRANJE PRILAGODLJIVOG RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA.....	50
3.1.1	<i>Analiza sustava za generiranje prilagodljivog računalom oblikovanog nastavnog sadržaja .....</i>	<i>60</i>
3.2	MODELIRANJE UČENIKA U SUSTAVIMA ZA GENERIRANJE PRILAGODLJIVOG RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA .....	65
3.2.1	<i>Primjeri sustava s Bayesovim modelom učenika .....</i>	<i>66</i>
3.2.1.1	ANDES .....	70
3.2.1.2	CAPIT .....	73
3.2.2	<i>Primjeri sustava sa stereotipnim modelom učenika .....</i>	<i>75</i>
3.2.2.1	GRUNDY .....	76
3.2.2.2	UNIX Consultant .....	78
<b>4</b>	<b>MODEL SUSTAVA AC-WARE TUTOR .....</b>	<b>82</b>
4.1	PODRUČNO ZNANJE I REPREZENTACIJA PODRUČNOG ZNANJA .....	82
4.1.1	<i>Struktura područnog znanja .....</i>	<i>82</i>
4.1.2	<i>Graf područnog znanja .....</i>	<i>85</i>
4.1.2.1	Komponente i cjeline grafa područnog znanja.....	88
4.1.3	<i>Reprezentacija područnog znanja .....</i>	<i>95</i>
4.2	GENERIRANJE, ODABIR, NIZANJE I PREZENTIRANJE ELEMENATA RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA .....	98
4.2.1	<i>Elementi za učenje.....</i>	<i>100</i>
4.2.2	<i>Elementi za provjeravanje znanja.....</i>	<i>108</i>
4.2.2.1	Predlošci pitanja.....	111
4.2.2.2	Ocjenjivanje poznavanja povezanosti .....	116
4.2.2.3	Izvedivost i generiranje testa .....	120
4.2.3	<i>Odabir i nizanje elemenata računalom oblikovanog nastavnog sadržaja prilagođeno stereotipovima .....</i>	<i>124</i>
4.2.4	<i>Prezentiranje elemenata računalom oblikovanog nastavnog sadržaja prilagođeno stereotipovima</i> <i>131</i>	
4.2.4.1	Predlošci tvrdnji za prezentiranje elemenata računalom oblikovanog nastavnog sadržaja.....	131
4.2.4.2	Algoritmi za prezentiranje elemenata računalom oblikovanog nastavnog sadržaja .....	134
4.2.5	<i>Dinamičko generiranje, odabir, nizanje i prezentiranje računalom oblikovanog nastavnog sadržaja</i> <i>137</i>	

---

4.3	MODELIRANJE UČENIKA .....	139
<b>5</b>	<b>PRIKAZ ARHITEKTURE I VREDNOVANJE PROTOTIPA SUSTAVA.....</b>	<b>146</b>
5.1	ARHITEKTURA PROTOTIPA SUSTAVA .....	146
5.1.1	<i>Osnovni UML dijagrami .....</i>	<i>147</i>
5.1.2	<i>Organizacija skupova podataka .....</i>	<i>149</i>
5.1.3	<i>Komponente sustava .....</i>	<i>152</i>
5.1.3.1	Podsustav AC-ware-ITS.....	152
5.1.3.2	Aplikacija AC-ware-Web i baza podataka ZapisUcenika .....	153
5.1.4	<i>Implementacija i postavljanje prototipa sustava .....</i>	<i>157</i>
5.1.5	<i>Način korištenja prototipa.....</i>	<i>157</i>
5.2	VREDNOVANJE PROTOTIPA SUSTAVA.....	160
5.2.1	<i>Prvo prototipno testiranje .....</i>	<i>160</i>
5.2.1.1	Analiza anketnog upitnika .....	161
5.2.2	<i>Drugo prototipno testiranje.....</i>	<i>164</i>
5.2.2.1	Analiza anketnog upitnika .....	164
5.2.3	<i>Zaključak vrednovanja prototipa.....</i>	<i>168</i>
<b>6</b>	<b>ZAKLJUČAK.....</b>	<b>169</b>
<b>7</b>	<b>LITERATURA.....</b>	<b>175</b>
<b>8</b>	<b>PRILOZI .....</b>	<b>187</b>
8.1	PRIKAZ SUSTAVA ZA GENERIRANJE RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA.....	187
8.2	PODRUČNO ZNANJE - RAČUNALO KAO SUSTAV .....	207
8.2.1	<i>Popis koncepata i relacija u ontologiji „Računalo kao sustav“ .....</i>	<i>207</i>
8.2.2	<i>Ontologija "Računalo kao sustav" prikazana u sustavu Cmap Tools COE – grafički prikaz.....</i>	<i>209</i>
8.2.3	<i>Ontologija "Računalo kao sustav" prikazana u sustavu Cmap Tools COE – LifeMap prikaz.....</i>	<i>210</i>
8.3	SEMANTIČKO ODREĐIVANJE REPREZENTACIJE ANALIZOM KONCEPATA I RELACIJA U PODRUČNOM ZNANJU .....	212
8.4	OBLIKOVANJE I STRUKTURIRANJE ELEMENATA RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA .....	219
8.4.1	<i>Konstruiranje elemenata računalom oblikovanog nastavnog sadržaja u područnom znanju „Računalo kao sustav“ .....</i>	<i>219</i>
8.4.2	<i>Nastavne cjeline, teme i jedinice u područnom znanju „Računalo kao sustav“.....</i>	<i>229</i>
8.4.2.1	<i>Strukture statičkog računalom oblikovanog nastavnog sadržaja za svaki od stereotipova prema znanju</i>	<i>241</i>
8.5	INICIJALIZIRANA BAYESOVA MREŽA ZA PODRUČNO ZNANJE „RAČUNALO KAO SUSTAV“ .....	270
8.6	DETALJNI DIJAGRAMI KLASA IMPLEMENTIRANOG PROTOTIPA AC-WARE TUTOR .....	271
8.7	ANKETNI UPITNIK KORIŠTEN U OKVIRU PROTOTIPNOG TESTIRANJA SUSTAVA AC-WARE TUTOR .....	276
<b>9</b>	<b>ŽIVOTOPIS.....</b>	<b>279</b>

---

# Popis slika

Slika 2.1. KOMPONENTE ITS-A (PREMA (BURNS & CAPPS, 1988)) .....	6
Slika 2.2. OKRUŽENJE DINAMIČKOG I PRILAGODLJIVOG GENERIRANJE RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA U INTELIGENTNIM TUTORSKIM SUSTAVIMA.....	13
Slika 2.3. STRUKTURA NOVOG MODELA.....	14
Slika 2.4. INTERAKCIJA U MODELU PRILAGODLJIVOG STJECANJA ZNANJA.....	15
Slika 2.5. OD PODRUČNOG ZNANJA DO GRAFA PODRUČNOG ZNANJA.....	18
Slika 2.6. CJELINA U PODRUČNOM ZNANJU I NAJVEĆI PUT .....	19
Slika 2.7. REPREZENTATIVNI PODGRAF PODRUČNOG ZNANJA .....	22
Slika 2.8. GRAFIČKA STRUKTURA BAYESOVE MREŽE .....	28
Slika 2.9. ODNOS PODRUČNOG ZNANJA I RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA .....	35
Slika 2.10. OPĆA ARHITEKTURA SUSTAVA KOJI PODRŽAVAJU NIZANJE SADRŽAJA (MODIFICIRANO PREMA .....	35
Slika 2.11. ELEMENT RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA ZA UČENJE.....	36
Slika 2.12. PRIMJER ELEMENATA RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA ZA UČENJE RAZLIČITIH RAZINA .....	37
Slika 2.13. ODREĐIVANJE POČETNOG STEREOTIPA UČENIKA .....	42
Slika 2.14. REALIZACIJA PROVJERAVANJA ZNANJA.....	43
Slika 2.15. GENERIRANJE TVRDNJI ZA UČENJE I POUČAVANJE .....	46
Slika 2.16. PREZENTIRANJE TVRDNJI ZA UČENJE I POUČAVANJE UČENIKU .....	46
Slika 2.17. REALIZACIJA UČENJA I POUČAVANJA .....	47
Slika 2.18. CIKLUSI U PROCESU UČENJA I POUČAVANJA.....	47
Slika 2.19. „POPRAVLJANJE“ ZAVRŠNOG STEREOTIPA (DO MAX. STRUČNJAK).....	48
Slika 3.1. BAYESOVA MREŽA .....	67
Slika 3.2. BAYESOVA MREŽA UPOTPUNJENA VJEROJATNOSTIMA .....	68
Slika 3.3. BAYESOVA MREŽA SUSTAVA ANDES .....	72
Slika 3.4. „MALA“ BAYESOVA MREŽA.....	74
Slika 3.5. „VELIKA“ BAYESOVA MREŽA.....	75
Slika 3.6. STEREOTIPOVI U SUSTAVU GRUNDY (RICH, 1979).....	76
Slika 3.7. STEREOTIP SPORTAŠ U SUSTAVU GRUNDY (RICH, 1979) .....	77
Slika 3.8. OKIDAČI U SUSTAVU GRUNDY (RICH, 1979) .....	77
Slika 4.1. UVJETNA VJEROJATNOST ČVORA <i>PROGRAMSKA PODRŠKA</i> .....	144
Slika 4.2. UVJETNA VJEROJATNOST ČVORA <i>MASOVNA MEMORIJA</i> .....	144
Slika 4.3. UVJETNA VJEROJATNOST ČVORA <i>NEGACIJA</i> .....	144
Slika 4.4. ISJEČAK IZ BAYESOVE MREŽE ZA ČVOR <i>NEGACIJA</i> .....	144
Slika 5.1. DIJAGRAM SLUČAJEVA KORIŠTENJA .....	147
Slika 5.2. TESTIRANJE ZNANJA .....	148
Slika 5.3. GENERIRANJE TVRDNJI ZA UČENJE .....	148
Slika 5.4. DIJAGRAM KLASA ZA UČENJE U MODULU UČITELJA .....	149
Slika 5.5. DIJAGRAM KLASA ZA TESTIRANJE U MODULU UČITELJA.....	149
Slika 5.6. PAKETI .....	150
Slika 5.7. KLASA MODULA UČITELJA.....	150
Slika 5.8. KLASA MODULA STRUČNJAKA.....	150
Slika 5.9. RELACIJSKA STRUKTURA ZA BAZU PODATAKA PODRUČNOZNANJE .....	151
Slika 5.10. KLASA MODULA UČENIKA.....	151
Slika 5.11. RELACIJSKA STRUKTURA ZA BAZU PODATAKA ZAPISUČENIKA .....	151
Slika 5.12. DIJAGRAM KOMPONENTI PROTOTIPA SUSTAVA .....	152
Slika 5.13. DIJAGRAM KLASA PODSUSTAVA AC-WARE-ITS.....	153
Slika 5.14. DIJAGRAM KLASA APLIKACIJE AC-WARE-WEB .....	154
Slika 5.15. PRIJAVA U SUSTAV .....	155

---

SLIKA 5.16. ODABIR PODRUČJA I PODPODRUČJA I POZDRAVNA PORUKA .....	155
SLIKA 5.17. UČENJE I POUČAVANJE .....	155
SLIKA 5.18. DIO PITANJA IZ TESTA .....	156
SLIKA 5.19. POV RATNA INFORMACIJA O TOČNOSTI ODGOVORA NA TESTU .....	156
SLIKA 5.20. PODSKUP PODRUČNOG ZNANJA ZA UČENJE .....	158
SLIKA 5.21. PRVI CIKLUS PROVJERAVANJA ZNANJA ZA STEREOTIP NOVAK S PITANJIMA IZ PRVE TEŽINSKE KATEGORIJE .....	158
SLIKA 5.22. DRUGI CIKLUS PROVJERAVANJA ZNANJA ZA STEREOTIP NOVAK S PITANJIMA IZ DRUGE TEŽINSKE KATEGORIJE .....	159
SLIKA 5.23. MOGUĆNOST ZAVRŠAVANJA S PROCESOM UČENJA I POUČAVANJA NA KAO STEREOTIP KOJI NIJE STRUČNJAKA .....	159
SLIKA 8.1. REPREZENTACIJA ZA STRUČNJAKE UZ PRAG PRIHVAĆANJA OD 50% .....	216
SLIKA 8.2. REPREZENTACIJA ZA STUDENTE UZ PRAG PRIHVAĆANJA OD 50% .....	217
SLIKA 8.3. REPREZENTACIJA ZAJEDNIČKA ZA STRUČNJAKE I STUDENTE UZ PRAG PRIHVAĆANJA OD 50%.....	218
SLIKA 8.4. USPOREDBA REPREZENTACIJA DOBIVENIH SEMANTIČKOM ANALIZOM I MATEMATIČKIM PRISTUPOM .....	218

---



# Popis tablica

TABLICA 2.1. EKSPERIMENTI UKLJUČENI U META-ANALIZU .....	7
TABLICA 2.2. STRUKTURA BLOOMOVE TAKSONOMIJE ZNANJA .....	30
TABLICA 2.3. STRUKTURA REVIDIRANE BLOOMOVE TAKSONOMIJE ZNANJA .....	31
TABLICA 2.4. STEREOTIPOVI PREMA ZNANJU .....	32
TABLICA 2.5. ODNOS STEREOTIPOVA UČENIKA I RAZINA ZNANJA .....	33
TABLICA 2.6. RAZINE ZNANJA, STEREOTIPOVI I VRSTE PITANJA S PREDLOŠCIMA .....	38
TABLICA 2.7. TEŽINSKE KATEGORIJE PITANJA I STEREOTIPOVI.....	41
TABLICA 2.8. STEREOTIPOVI I NAJVEĆA RAZINA ELEMENATA RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA .....	43
TABLICA 2.9. PRAVILA ZA DODAVANJE ELEMENATA ZA PROVJERAVANJE ZNANJA .....	44
TABLICA 2.10. PREZENTIRANJE ZNANJA PREMA STEREOTIPOVIMA.....	45
TABLICA 2.11. STEREOTIPOVI I KOLIČINA I RAZINA ZNANJA KOJA SE PREZENTIRA.....	45
TABLICA 2.12. RAZINE ZNANJA, STEREOTIPOVI I VRSTE TVRDNJI S PREDLOŠCIMA .....	46
TABLICA 3.1. REZULTAT PRETRAŽIVANJA U BAZI SCIENCE DIRECT .....	59
TABLICA 3.2. PREGLED PRONAĐENIH SUSTAVA I NJHOVA CITIRANOST .....	59
TABLICA 3.3. USPOREDNA TABLICA .....	64
TABLICA 3.4. TABLICA UVJETNE VJEROJATNOST ČVORA PRIMJENA PRAVILA .....	72
TABLICA 3.5. TABLICA UVJETNE VJEROJATNOST ČVORA PROPOZICIJE .....	73
TABLICA 4.1. TEŽINSKE KATEGORIJE PITANJA U KVIZU .....	110
TABLICA 4.2. BLOOMOVA TAKSONOMIJA I VRSTE PITANJA .....	110
TABLICA 4.3. PREDLOŠCI I KONCEPTI NA KOJE SE ODOSE.....	117
TABLICA 4.4. PODSKUPOVI PODRUČNOG ZNANJA I ODGOVARAJUĆE RAZINE ZNANJA PREMA BLOOMOVOJ TAKSONOMIJI .....	120
TABLICA 4.5. STEREOTIPOVI ODREĐENI SKUPOM KOJI IMA NAJVIŠE ELEMENATA.....	123
TABLICA 4.6. UVJETNE VJEROJATNOSTI .....	139
TABLICA 5.1. DINAMIČKE WEB STRANICE APLIKACIJE AC-WARE-WEB .....	154
TABLICA 5.2. PITANJA I ODGOVORI PRVOG DIJELA ANKETNOG UPITNIKA .....	161
TABLICA 5.3. PITANJA I ODGOVORI DRUGOG DIJELA ANKETNOG UPITNIKA.....	162
TABLICA 5.4. PITANJE I ODGOVOR TREĆEG DIJELA ANKETNOG UPITNIKA .....	163
TABLICA 5.5. PITANJA I ODGOVORI PRVOG DIJELA ANKETNOG UPITNIKA .....	164
TABLICA 5.6. PITANJA I ODGOVORI DRUGOG DIJELA ANKETNOG UPITNIKA.....	166
TABLICA 5.7. PITANJE I ODGOVOR TREĆEG DIJELA ANKETNOG UPITNIKA .....	166
TABLICA 8.1. REZULTATI SEMANTIČKE ANALIZE STRUČNJAKA.....	212
TABLICA 8.2. REZULTATI SEMANTIČKE ANALIZE OD STRANE STUDENATA .....	213
TABLICA 8.3. ELEMENTI RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA "RAČUNALO KAO SUSTAV" .....	230
TABLICA 8.4. ELEMENTI RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA "RAČUNALO KAO SUSTAV" - ABECEDNO.....	236
TABLICA 8.5. STRUKTURA STATIČKOG RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA ZA STEREOTIP NOVAK .....	241
TABLICA 8.6. STRUKTURA STATIČKOG RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA ZA STEREOTIP POČETNIK.....	246
TABLICA 8.7. STRUKTURA STATIČKOG RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA A ZA STEREOTIP OSREDNJI.....	251
TABLICA 8.8. STRUKTURA STATIČKOG RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA ZA STEREOTIP NAPREDNI .....	261
TABLICA 8.9. STRUKTURA STATIČKOG RAČUNALOM OBLIKOVANOG NASTAVNOG SADRŽAJA ZA STEREOTIP STRUČNJAK.....	267

# 1 Uvod

„To improve is to change, to be perfect is to change often.“

Winston Churchill

Kroz povijest računala mnogo se pažnje posvetilo ideji o uporabi računala kao inteligentnog suradnika, odnosno osobnog učitelja, koji će nam otkriti i objasniti zamršena područna znanja. Složenost tog zadatka je uočena nakon što se za konačni cilj u realizaciji te ideje uzelo *inteligentno poučavanje* (eng. intelligent tutoring). *Umjetna inteligencija* (eng. artificial intelligence, AI), kao područje koje spaja računala i inteligentno ponašanje, javlja se krajem 50-tih i početkom 60-tih godina prošlog stoljeća sa začetnicima Alanom Turing-om, Marvinom Minsky-m, Johnom McCarthy-iem i Allenom Newell-om (Urban-Lurain, 1996). AI je u suštini orijentirana na prikaz znanja (eng. knowledge representation), razumijevanje prirodnog jezika (eng. natural language understanding), te rješavanje problema (eng. problem solving), a sve to je jednako važno za razvoj koncepta inteligentnog poučavanja (Beck, Stern, & Haugsjaa, 1996).

Šezdesetih godina prošlog stoljeća znanstvenici su razvili brojne sustave za realizaciju *nastave pomoću računala* (eng. Computer Assisted Instructional, CAI) (Uhr, 1969). U uporabi je i termin *vježbanje temeljeno na računalu* (eng. Computer Based Training, CBT) koji obuhvaća sve sustave koji omogućavaju „neinteligentni“ način učenja i poučavanja. Spomenuti CAI sustavi bi učenika upoznali s problemom, primili i zapamtili učenikov odgovor. Oni se nisu pretjerano bavili problematikom kako učenici uče, već su smatrali da ako je učeniku dana informacija, on će je usvojiti. Veći problem tada su bili tehnički izazovi u programiranju na skupim i glomaznim „mainframe“ računalima.

Kasnih 60-tih i ranih 70-tih godina prošlog stoljeća znanstvenici su prešli s pukog prezentiranja zadataka učenicima na uvažavanje učenika, kao faktora koji utječe na proces učenja i poučavanja (Suppes, 1967, prema (Uhr, 1969)). Neki od razvijenih sustava mijenjali su način prikazivanja nastavnih sadržaja ovisno o učenikovim odgovorima. Programeri su stoga morali unaprijed znati sve moguće učenikove odgovore i odlučiti koju će informaciju prezentirati sustav. Ovo predstavlja početak modeliranja učenika, iako je u ovom razdoblju promatrano samo učenikovo ponašanje, a nije modelirano znanje učenika.

Godine 1982., Sleeman i Brown su u (Sleeman & Brown, 1982) dali pregled posljednjih dostignuća u području nastave potpomognute računalima i prvi put su uporabili složenicu *inteligentni tutorski sustavi* (eng. Intelligent Tutoring Systems, ITS) da bi opisali nove sustave i da bi ih mogli razlikovati od starih CAI sustava. Učenje se kod ovih novih sustava vrši na principu „učim kroz rad“ (eng. learning-by-doing). Sleeman i Brown su klasificirali postojeće

ITS-ove u četiri klase: (1) nadziratelji rješavanja problema (eng. problem-solving monitors), (2) poučavatelji (eng. coaches), (3) laboratorijski instruktori (eng. laboratory instructors) te (4) savjetnici (eng. consultants). Rickel (Rickel, 1989) navodi da su „inteligentni“ CAI sustavi, odnosno ICAI sustavi, oni CAI sustavi koji imaju područno znanje stručnjaka i znanje pedagoga, te ih smatra jednakima ITS-ovima. Kao jedan od prvih istraživača u području učenja uz pomoć računala i od računala, Carbonell je spojio CAI i AI implementirajući sustav SCHOLAR, koji se smatra jednim od prvih inteligentnih tutorskih sustava (Carbonell, 1970).

Inteligentni tutorski sustavi spadaju u kategoriju sustava temeljenih na znanju jer prikazuju znanje o područnom znanju kojeg poučavaju, znanje o učeniku koga poučavaju i znanje o tutorskim metodama koje koriste za poučavanje učenika (Self, 1974).

Novi pojam koji se javlja krajem 90-tih godina prošlog stoljeća je *e-učenje*. *E-učenje* predstavlja presjek svijeta informacijske i komunikacijske tehnologije (eng. information and communication technology, ICT) i svijeta obrazovanja (Stankov, Grubišić, & Žitko, 2004). U usporedbi s tradicionalnom nastavom u razredu koja u središte stavlja učitelja i njegovu kontrolu nad razredom, nastavnim sadržajem i procesom učenja i poučavanja, e-učenje u središte stavlja učenika kojem omogućava interaktivno učenje vlastitim ritmom, u jednostavnom, fleksibilnom, distribuiranom okruženju za učenje (Khan, 2001). Najčešće korištena definicija e-učenja jest da je e-učenje skup aplikacija i procesa, kao što su učenje temeljeno na Web-u (eng. Web-based learning), učenje temeljeno na računalu (eng. computer-based learning), virtualni razredi (eng. virtual classrooms) i digitalna suradnja (eng. digital collaboration), koji omogućavaju pristup nastavnim sadržajima pomoću različitih elektroničkih medija (CD-ROM, Internet, intranet, extranet, audio i video, satelit, itd.) (ASTD, 2001).

Istraživanje u okviru doktorske disertacije predstavlja nastavak dugogodišnjeg istraživanja, razvoja i primjene modela Tutor-Expert System (TEx-Sys) za oblikovanje inteligentnih tutorskih sustava za po volji odabrana područna znanja (Stankov, 1997), (Stankov, Rosic, Žitko, & Grubisic, 2008). Tijekom posljednjih petnaest godina razvijene su tri verzije sustava TEx-Sys koje se razlikuju po arhitekturi, implementiranim funkcionalnostima i podržanim sudionicima. U svim verzijama sustava TEx-Sys prikaz znanja je temeljen na semantičkoj mreži s okvirima. Učenje se realizira navigacijom kroz mrežu čvorova semantičke mreže, a testiranje se provodi dinamičkim kvizom. Kviz u modelu TEx-Sys dinamički generira pitanja i odgovore temeljem strukture područnog znanja uz prisutnu prilagodbu težine pitanja za učenika. Jedino u prvoj verziji sustava je implementiran pristup konstruktivističkog provjeravanja znanja u kojem učenik nastoji nadopuniti problemsku mrežu područnog znanja do početne originalne mreže područnog znanja koju je oblikovao stručnjak. Ovakav način testiranja znanja pokazao se pogodnim za dijagnosticiranje nedostajućeg znanja i pogrešnog poimanja učenika.

Nastavak istraživanja je usmjeren prema ostvarivanju prilagođavanja prema znanju učenika na način da se svakom učeniku automatski i dinamički generira nastavni sadržaj prilagođen

isključivo njegovom znanju. Novi model inteligentnog tutorskog sustava se razlikuje od sustava eXtended Tutor-Expert System (xTEEx-Sys) (Stankov, 2003), posljednje izvedenice modela TEx-Sys, po tome što ne sadrži autorsko okruženje za oblikovanje nastavnog sadržaja. Ovu funkcionalnost obavlja algoritam za generiranje i nizanje elemenata nastavnog sadržaja, a na osnovi prethodno oblikovanog područnog znanja. Stručnjak mora koristiti posebni autorski alat kako bi oblikovao područno znanje.

Cilj istraživanja ove disertacije je oblikovanje modela prilagodljivog stjecanja znanja učenika u sustavu e-učenja. Ovakav model sustava e-učenja je poboljšani model inteligentnog tutorskog sustava sastavljen od modula: (i) stručnjaka sa specifikacijom reprezentativnog skupa koncepata i relacija za utvrđivanje početne razine znanja učenika; (ii) učenika sa stereotipovima utvrđenim prema Bloomovoj taksonomiji znanja i Bayesovim mrežama predikcije znanja; (iii) učitelja temeljenog na Bloomovoj taksonomiji i u suglasju s pedagoškom praksom u Hrvatskoj. Očekujemo da ovaj model sustava e-učenja poboljša proces učenja, poučavanja i testiranja znanja učenika za po volji odabrano područno znanje.

Stereotip se koristi kako bi se temeljem njega napravio nastavni sadržaj prilagođen učeniku prema njegovom znanju. Uvažavanje stereotipa učenika u sustavima e-učenja ima za cilj poboljšano stjecanje novih vještina, znanja i/ili ponašanja, u odnosu na učenje bez uvažavanja stereotipa.

U *drugom* poglavlju je opisana zamisao novog modela prilagodljivog stjecanja znanja u sustavima e-učenja. Iznosi se struktura novog modela temeljena na tradicionalnoj modularnoj arhitekturi ITS-a. Osim toga, predlažu se funkcionalnosti svakog od modula kao i pristupi modeliranju podataka nad kojima se realiziraju funkcionalnosti. Prilikom opisivanja svakog od modula navode se i tradicionalni pristupi u njihovom modeliranju.

U *trećem* poglavlju je dano stanje istraženosti. Uz elaboraciju teorijskih postavki, prikazan je povijesni razvoj ideje o prilagođavanju procesa učenja i poučavanja, kao i analiza strukture i funkcionalnosti za ukupno sedamnaest sustava e-učenja koje karakterizira generiranje prilagodljivih nastavnih sadržaja. Za svaki sustav su istaknute funkcionalnosti pojedinih modula, kao i posjedovanje poželjnih karakteristika. Na kraju je napravljena usporedba sa stajališta njihovih općih karakteristika i s obzirom na realizaciju prilagođavanja nastavnih sadržaja prema znanju učenika. Osim toga, naglašene su zajedničke karakteristike uspoređenih sustava e-učenja i vlastitog pristupa realiziranog u okviru modela *Adaptive Courseware Tutor (AC-ware Tutor)*.

U *četvrtom* poglavlju detaljno opisujemo formalni model sustava AC-ware Tutor i njegove komponente. Opisujemo model AC-ware Tutor preko njegovih sudionika i funkcionalnosti. Nadalje je prikazana struktura modela u kojoj se ističu osnovne komponente i skupovi podataka. Slijedi formalna definicija osnovnih skupova podataka temeljena na teoriji skupova i teoriji grafova. Realizacija funkcionalnosti je prikazana opisivanjem metoda za

dobivanje područnog znanja, nastavni sadržaj i inicijalni model učenika, te metoda za učenje i testiranje znanja učenika.

U *petom* poglavlju prikazana je arhitektura prototipa modela AC-ware Tutor temeljenog na opisanom modelu, kao i provedeno istraživanje kojim se vrednovao taj prototip. Opis arhitekture započinje prikazom komponenti koje se nadalje razrađuju do klasa i objekata. Prvo se prikazuje arhitektura sustava AC-ware Tutor, kao i klase modela skupova. Zatim se daje klasni prikaz arhitekture prototipa aplikacije kojom se realizira Web okruženje za učenje i testiranje. Nadalje se opisuju instrumenti, tijek i okruženje za rad prototipnih istraživanja provedenih na dvije skupine ispitanika. Ujedno se interpretiraju rezultati svake skupine kao i informacije prikupljene anketnim upitnikom.

U *šestom* poglavlju se daje osvrt na doktorsku disertaciju po poglavljima i iznose se pojedini znanstveni doprinosi.

Popis korištene literature se nalazi u *sedmom* poglavlju.

Prilozi rada iz *osmog* poglavlja sadrže kronološki prikaz sustava za generiranje računalom oblikovanih nastavnih sadržaja, razvijenu ontologiju područnog znanja namijenjenu poučavanju i testiranju učenika, rezultate istraživanja o semantičkom određivanju reprezentacije područnog znanja analizom koncepata i relacija, detaljan prikaz oblikovanja i strukturiranja elemenata računalom oblikovanih nastavnih sadržaja, detaljne dijagrame klasa implementiranog prototipa sustava AC-ware Tutor i anketni upitnik provedenog prototipnog testiranja.

## 2 Zamisao modela za prilagodljivo stjecanje znanja u sustavima e-učenja

„Svijet se mijenja, a s njime se trebaju mijenjati i ljudi. Oni koji se ne uspiju prilagoditi neće moći opstati.“

Don Quijote

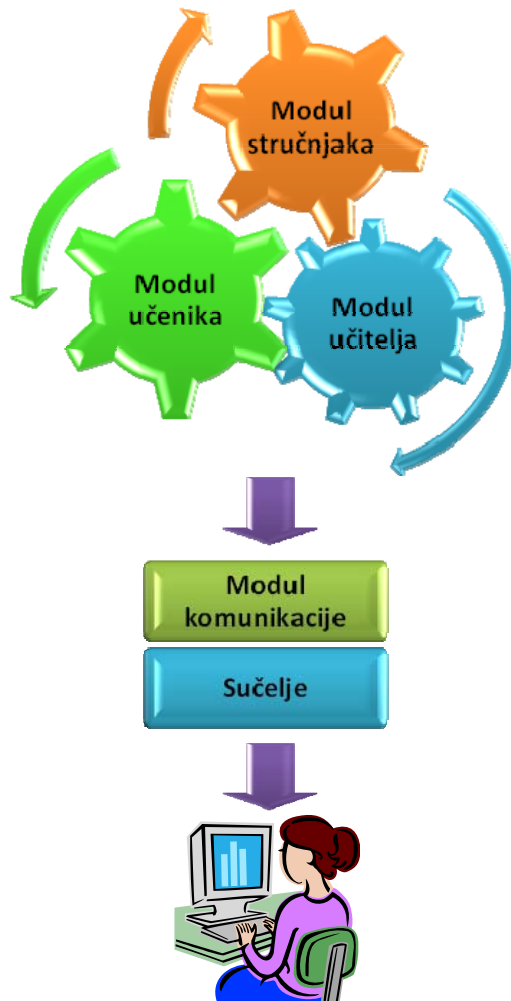
*Sustavi e-učenja* omogućavaju pristup elektroničkim izvorima za učenje bilo kada i bilo kome (Albert, 2001). *Inteligentni sustavi e-učenja* imaju sposobnost prikladnog ponašanja u neizvjesnim situacijama koje se javljaju u procesu učenja i poučavanja. Posebna klasa inteligentnih sustava e-učenja su *inteligentni tutorski sustavi* (ITS).

Više od četrdeset godina razvoja dijeli današnje inteligentne tutorske sustave od prvih pristupa i pokušaja razvoja. Mnoštvo sustava je razvijeno, implementirano i testirano u okviru nastavnog procesa u školi i na fakultetima, te je postignut je jednoglasan dogovor o arhitekturi ITS-ova.

Inteligentni tutorski sustavi su računalni sustavi koji su namijenjeni potpori i poboljšanju procesa učenja i poučavanja u odabranom područnom znanju, poštujući individualnost učenika kao što je to slučaj kod tradicionalnog poučavanja jedan-na-jedan ((Wenger, 1987), (Ohlsson, 1987), (Sleeman & Brown, 1982)). ITS-ovi svojim modelima područnog znanja određuju što će se poučavati, a strategijama poučavanja određuju kako će se poučavati ((Wenger, 1987), (Ohlsson, 1987), (Shute & Psocka, 1996)). Znanje koje sustav ima o područnom znanju, principi pomoću kojih sustav poučava i metode pomoću kojih primjenjuje te principe, te metode i tehnike za modeliranje učenika tijekom stjecanja znanja i umijeća čine okosnicu ITS-ova.

Svaki se ITS sastoji od četiri komponente: područnog znanja, modula učitelja, modula učenika (Self, 1994) i komunikacijskog modula (Woolf, 1992). Ovim je uvjetovana struktura ITS-a (Burns & Capps, 1988): (1) baza područnog znanja (modul stručnjaka), (2) modul za vođenje poduke (modul učitelja), (3) modul za obuhvat i procjenu znanja učenika (modul učenika), (4) modul interakcije učenika, učitelja i područnog znanja (modul komunikacije) - Slika 2.1.

*Modul stručnjaka* izvršava dvije važne funkcije. Prvo, on služi kao izvor znanja koje se predstavlja učeniku. Drugo, on predstavlja standard za ocjenjivanje učenikovih performansi. Modul stručnjaka je okosnica svakog inteligentnog tutorskog sustava jer sadrži bazu područnog znanja, a znanje predstavlja ključ za inteligentno ponašanje (Anderson, 1988).



Slika 2.1. Komponente ITS-a (prema (Burns & Capps, 1988))

Funkcija *modula učenika* je u bilježenju učenikova razumijevanja ili nerazumijevanja područnog znanja, tj. on bilježi učenikovo napredovanje. Sleeman i Brown (Sleeman & Brown, 1982) prvi put spominju model učenika kao pojam koji predstavlja apstraktni prikaz učenika u sustavu. Oni su klasificirali modele učenika u tri klase: (1) model prekrivanja (eng. overlay) – promatra učenikovo znanje kao podskup znanja stručnjaka, (2) model razlike (eng. differential) – promatraju se razlike između znanja učenika i znanja stručnjaka, te (3) model zbrke (eng. perturbation) – promatra se učenikovo pogrešno shvaćanje (eng. misconceptions) u usporedbi sa znanjem stručnjaka.

*Modul učitelja* je usko povezan s modulom učenika te odlučuje o načinu poučavanja svakog učenika ponaosob. Modul učitelja, kao sastavni dio ITS-a koji preuzima ulogu stvarnih učitelja, mora zadovoljiti trima karakteristikama: (1) tutor mora imati kontrolu nad izborom i redoslijedom nastavnih sadržaja, (2) tutor mora odgovarati na učenikova pitanja, te (3) tutor mora prepoznati kada učeniku treba pomoć i kakva pomoć mu treba. Te zadatke modul učitelja rješava pomoću tzv. scenarija učenja. Scenarij učenja je situacija u kojoj se odvija učenje, tj. načini na koje ITS poučava učenika (Rickel, 1989).

*Modul komunikacije* kontrolira interakciju između ITS-a i učenika. Ono predstavlja sučelje učenika i okruženje nastavnog procesa odnosno interakciju učenik-učitelj-znanje. Modul komunikacije upravlja interakcijom sustava s učenikom koja se ostvaruje preko dijaloga i grafičkog korisničkog sučelja.

U nastavku ćemo pobliže objasniti motivaciju i razloge koji su doveli do istraživanja provedenog u okviru ove doktorske disertacije. Ideja s razradom na visokoj razini je također opisana.

## 2.1 Motivacija

Istraživanje u okviru doktorske disertacije predstavlja nastavak dugogodišnjeg istraživanja, razvoja i primjene modela Tutor-Expert System (TEx-Sys) za oblikovanje inteligentnih tutorskih sustava za po volji odabrana područna znanja (Stankov, 1997), (Stankov et al., 2008). Tijekom posljednjih petnaest godina razvijene su tri verzije sustava TEx-Sys koje se razlikuju po arhitekturi, implementiranim funkcionalnostima i podržanim sudionicima. Svaka od verzija donijela je specifičnosti za koje je trebalo utvrditi da li poboljšavaju proces učenja i poučavanja.

Stoga je razvijena metodologija za vrednovanje učinka inteligentnih sustava e-učenja u okviru magistarskog rada (Grubišić, 2007). Na temelju razvijene metodologije u posljednjih šest godina provedeni su brojni eksperimenti s učenicima različite kronološke dobi sa različitim verzijama TEx-Sys modela kojima se vrednovala učinkovitost istog (Stankov, Glavinić, & Grubišić, 2004), (Grubišić, 2007). U meta-analizi napravljenoj nad 11 eksperimenata koji su provedeni u ak.god. 2005/2006 i 2006/2007 s 499 učenika, dobivena je veličina učinka sustava xTEx-Sys od 0,71 (Grubišić, 2010) (Tablica 2.1). Dobivena veličina učinka je velika (prema (Cohen, 1988)), ali ipak je daleko manja od 2-sigma, učinkovitosti tutorskog poučavanja (Bloom, 1984).

Tablica 2.1. Eksperimenti uključeni u meta-analizu

Područno znanje	Veličina uzorka	Trajanje	Veličina učinka
Uvod u računarstvo	80 studenata I. god.	14 tjedana	0,16
Kemija	41 8. raz. O. Š.	10 tjedana	0,60
Fizika - optika	80 8. raz. O. Š.	7 tjedana	0,75
Priroda i društvo	48 2. raz. O. Š.	6 tjedana	0,80
	48 3. raz. O. Š.	6 tjedana	0,83
	40 4. raz. O. Š.	6 tjedana	1,11
Uvod u računarstvo	39 studenata I. god	14 tjedana	0,42
QBASIC - programiranje	39 studenata I. god	14 tjedana	0,45
Matematika	18 6. raz. O. Š.	7 tjedana	1,32
	18 8. raz. O. Š.	7 tjedana	0,15
	48 5. raz. O. Š.	5 tjedana	0,38



Rezultati dobiveni iz navedenih eksperimenata i meta-analize pokazali su da je potrebno uvesti poboljšanja u model TEx-Sys koja bi eventualno dovela do povećanja veličine učinka istog. Nije imalo smisla nastaviti provoditi eksperimente nad istom verzijom modela, već je trebalo uvesti promjene, a onda te promjene vrednovati u sljedećim eksperimentima. Ciklus u kojem se izmjenjuju razvoj i vrednovanje završava u momentu kada sustav kojeg vrednujemo pokaže učinkovitost od najmanje 2-sigma, jer se danas, još uvijek, najučinkovitijim oblikom nastave smatra tutorsko poučavanje – jedan živi učitelj na jednog učenika.

Upravo ova činjenica, o nužnosti uvođenja poboljšanja, je usmjerila istraživanje u okviru doktorske disertacije prema pronalaženju načina kako postići veće veličine učinka. Krenulo se s istraživanjem u kojem modulu ITS-a i na koji način postići poboljšanje koje bi dovelo do povećanja veličine učinka do 2-sigma. Navesti ćemo dva smjera istraživanja koja su se provodila u okviru dvije različite doktorske disertacije.

*Jedan pravac* istraživanja je išao prema obradi kontroliranog jezika nad ontologijom (Žitko, 2010). U svim verzijama modela TEx-Sys prikaz znanja je temeljen na semantičkoj mreži s okvirima. Učenje se realizira navigacijom kroz mrežu čvorova semantičke mreže, a testiranje se provodi dinamičkim kvizom. Nepravilna gramatička struktura dinamički generiranih pitanja je otežavala učeniku razumijevanje onoga što ga se pita. Generiranje kontroliranog jezika u novom modelu inteligentnog tutorskog sustava omogućava učeniku učenje na osnovu prikaza rečenica. Novi pristup je omogućio realizaciju dijaloga jednosmjerne inicijative u kojem računalni tutor postavlja pitanja, a učenik odgovara i dobiva povratnu informaciju na kontroliranom jeziku. Na prototipu Controlled Language Based Tutor (CoLaB Tutor-a) (Žitko, 2010) je provedeno akcijsko istraživanje o osjećaju "zadovoljstva". Ustanovljeno je da su studenti razumjeli nastavni sadržaj i smatraju kako bi primjena CoLaB Tutor-a utjecala na kvalitetu tradicionalne nastave. Također, studenti su bolje razumjeli nastavni sadržaj prezentiran u CoLaB Tutor-u i nego u xTEx-Sys-u i općenito su mu dalji bolju ocjenu.

*Drugi pravac* istraživanja proveden je u okviru ove disertacije. Odlučeno je da će se uvesti poboljšanja u formi *automatskog i dinamičkog generiranja prilagodljivog nastavnog sadržaja temeljenog na stereotipovima učenika, Bayesovim mrežama i Bloomovoj taksonomiji znanja*. Promjene će se najviše očitovati u modulu učitelja i učenika.

S obzirom da smo se odlučili na realizaciju prilagodljivosti nastavnih sadržaja, u nastavku ćemo pobliže objasniti što je to prilagodljivost i kako se ona može ostvariti u sustavima e-učenja.

## 2.2 Prilagodljivost u sustavima e-učenja

Nastavni pristupi i tehnike koje su usmjerene prema potrebama pojedinog učenika nazivaju se *prilagodljiva nastava* (eng. adaptive instruction) (Corno & Snow, 1986, prema (Park & Lee, 1996)). *Prilagodljivi sustavi* (eng. adaptive systems - AS) su sustavi koji mogu mijenjati svoju strukturu, funkcionalnosti ili sučelje da bi se prilagodili različitim potrebama pojedinca ili grupe kao i promjenama njihovih potreba tijekom vremena (Benyon & Murray, 1993).

Prilagodljivi sustavi i prilagodljiva nastava definiraju novu klasu sustava - *prilagodljive obrazovne sustave* (eng. adaptive educational systems - AES) ili *prilagodljive sustave e-učenja* (eng. adaptive e-learning systems) koji prilagođavaju proces učenja, poučavanja i testiranja znanja različitim značajkama učenika. Dva najpoznatija predstavnika prilagodljivih sustava e-učenja su *inteligentni tutorski sustavi* (eng. intelligent tutoring systems - ITS) i *sustavi prilagodljive hipermedije* (eng. adaptive hypermedia systems – AHS).

U tradicionalnoj nastavi u razredu veza između učitelja i učenika je jedan-na-više (1-∞). U tutorskom okruženju ta veza je jedan-na-jedan (1-1). Prilagodljivi sustavi e-učenja koriste vezu više učitelja na jednog učenika (∞-1) kojom omogućava da jedan učenik ima više načina poučavanja. Postoji mnogo aspekata učenika koje različiti sustavi e-učenja žele utvrditi, kao što su znanje, stilovi učenja, kognitivni stilovi, osjećaji, sklonosti, itd. (Lin, 2007). Poblje ćemo objasniti dva najvažnija pristupa realizaciji prilagodljivosti u sustavima e-učenja: prema stilovima učenja i prema kognitivnim značajkama.

Općenito, teorija o stilovima učenja se temelji na spoznaji da ljudi imaju različite pristupe učenju (način na koji usvajaju i obrađuju informacije) koje će biti učinkovito ako im se pruži odgovarajući način poučavanja prilagođen njihovom stilu učenja (Gilbert, 2000). Uvijek postoji mogućnost da učenik ima više stilova učenja za isto područno znanje ili da im u jednom područnom znanju odgovara jedan stil učenja, a u drugom neki drugi stil učenja. Također, stilovi učenja se mijenjaju tijekom procesa učenja i poučavanja, što većina sustava koji se temelje na stilovima učenja ne uzimaju u obzir.

Najpoznatija klasifikacija stilova učenja je Felder-Silvermanov model (Felder & Silverman, 1988) u kojem se spominju načini na koje učenici usvajaju i obrađuju informacije: opazajno i intuitivno, vizualno i auditorno, induktivno i deduktivno, reflektivno i aktivno, sekvencijalno i općenito. Opažanje uključuje promatranje, sakupljanje informacija, dok intuitivno uključuje nagađanje, imaginaciju. Vizualni učenici najbolje zapamte ono što vide, dok auditorni učenici pamte ono što čuju. Indukcija je način zaključivanja koji ide od specifičnog prema općenitom, a dedukcija obrnuto. Pri indukciji se izvode principi, a pri dedukciji se utvrđuju posljedice. Aktivno eksperimentiranje uključuje diskusije ili objašnjavanje ili testiranje informacija u vanjskom svijetu, dok reflektivno promatranje uključuje sagledavanje i manipuliranje informacijama introspektivno. Aktivni učenici dobro rade u grupama, dok reflektivni učenici najbolje rade sami ili u paru. Aktivni učenici teže eksperimentiranju, a reflektivni

teoretiziranju. Neki učenici uče sekvencijalno na način kako im je nešto bilo prezentirano. Neki se učenici danima ili tjednima ne miču s mrtve točke, ne pokazuju osnovno razumijevanje, ne mogu riješiti najjednostavnije probleme, sve dok im se ne „upali lampica“. Tada razumiju poučavane sadržaje tako dobro da ih mogu primijeniti na složenim problemima. Oni uče globalno.

Prilagodljivi sustavi e-učenja koji implementiraju stilove učenja u procesu učenja i poučavanja su *sustavi prilagodljive hipermedije*. Oni prilagođavaju samo način prezentiranja učenikovom stilu učenja, ali ne koriste spoznaju o stilu učenja za odlučivanje o tome što će se prikazivati učeniku ((Brusilovsky, Schwarz, & Weber, 1996), (Beumont & Brusilousky, 1995), (Brusilovsky, 1996)). Sustavi prilagodljive hipermedije koriste prilagodljivu prezentaciju i prilagodljivu potporu navigaciji koja se temelji na auditornom/vizualnom stilu učenja. Prilagodljivom prezentacijom se prilagođava ili sadržaj dokumenta ili stil teksta. Stranice nisu statične, već se prilagodljivo generiraju ili slažu od različitih dijelova za svakog učenika (C-Book i Hypertext) (Brusilovsky, 1998). Prilagodljiva potpora navigaciji podupire učenika u orijentiranju i navigaciji u hiperprostoru mijenjanjem izgleda vidljivih veza (ELM-ART, InterBook, WEST-KBNS i AST) (Brusilovsky, 1998). Osnovne metode prilagođavanje sadržaja su: dodatna objašnjenja (samo za one učenike koji ih mogu razumjeti), preduvjetna objašnjenja (ponavljanje onih koncepata za učenike koji nemaju dovoljno predznanja), usporedna objašnjenja (za objašnjavanje novih koncepata koji su slični već usvojenima), varijante objašnjenja (iste informacije, ali različito prezentirane – tekst, grafika, zvuk), sortiranje (iste informacije uređene svakom učeniku posebno). Osnovne metode prilagođavanje hiperveza su: globalno i lokalno vođenje (učeniku se predlaže putanja kroz sve lekcije ili prva sljedeća lekcija), potpora globalnoj i lokalnoj orijentaciji (na poseban način se označavaju se hiperveze prema naučenim, preporučenim ili zabranjenim čvorovima).

Jedan primjerak ovih sustava je *Arthur* (Gilbert, 2000). Svaka jedinica poučavanja se može prezentirati učeniku na različite načine: auditorno, vizualno i tekstualno. Nakon svake jedinice poučavanja, sustav provjerava učenikovo znanje putem niza kratkih pitanja. Ako učenik pokaže znanje (rezultat bolji od 80%), sustav nastavlja s poučavanjem. Ograničenje od 80% je postavio Bloom u okviru učenja s provjeravanjem (eng. mastery learning) (Bloom, 1976, prema (Gilbert, 2000)). Ako učenik pokaže nerazumijevanje, sustav prezentira nenaučene sadržaje, ali na drugačiji način. Time se ostvaruje element prilagodljivosti u sustavu Arthur.

*AHyCo (Adaptive Hypermedia Courseware)* (Hoic-Bozic & Mornar, 2005) je još jedan obrazovni sustav prilagodljive hipermedije (eng. adaptive educational hypermedia system - AEHS) koji se koristi za stvaranje i ponovno upotrebljavanje prilagodljivog računalom oblikovanog nastavnog sadržaja s naglaskom na potporu prilagodljivoj navigaciji i nizanju lekcija. Ovaj sustav preporučuje učeniku koje stranice treba vidjeti na temelju učenikovog znanja i nekih specifičnih atributa lekcije. Sustav nastoji što manje ograničiti slobodu kretanja

učenika kroz sadržaje za učenje, no istovremeno usmjerava i vodi učenika pri usvajanju znanja. Učenikov put kroz prostor znanja ovisi o njegovoj početnoj i aktualnoj razini znanja.

Drugi pristup prilagođavanja učenicima prema njihovim kognitivnim sposobnostima opisuju Kinshuk, Lin i Patel (Kinshuk, Lin, & Patel, 2005) u svom modelu kognitivnih značajki (eng. Cognitive Trait Model - CTM). Utvrđene su četiri kognitivne sposobnosti: kapacitet radne memorije, sposobnost induktivnog zaključivanja, sposobnost učenja pridruživanjem, brzina obrađivanja informacija. Kapacitet radne memorije se odnosi na veličinu tranzijentne memorije koja se može nakratko aktivirati za ograničenu količinu informacija ( $7\pm 2$  koncepta) koja će se kasnije obraditi (Miller, 1956, Bunting i ostali, 2004, Kane i Engle, 2003, prema (Lin, 2007)). Sposobnost induktivnog zaključivanja se odnosi na sposobnost generaliziranja apstraktnog pojma na temelju njegovih instanci. Sposobnost učenja pridruživanjem se odnosi na stvaranje veza između novih i naučenih konceptata. Ove kognitivne sposobnosti su značajke učenikovog ukupnog kognitivnog kapaciteta te se nazivaju kognitivne značajke (eng. cognitive traits). Prilagođavanje u ovom kontekstu utječe na ne-preopterećivanje kognitivnih kapaciteta učenika jer to dovodi do obeshrabrivanja učenika. Sustav kod kojeg je implementiran CTM je *MCALM (Marginal Costing adaptive learning module)* (Lin, 2007).

Navedeni pristupi prilagođavanju u sustavima e-učenja temelje se na relativno stabilnim karakteristikama učenika koje se mogu primijeniti na različita područna znanja i probleme, čime omogućavaju relativno pouzdanu predikciju učenikovih performansi. Učenikovo znanje, kao glavni razlog postojanja sustava e-učenja, ovisno je o područnom znanju te se razvija tijekom procesa učenja i poučavanja. Upravo zato je nemoguće, na temelju usvojenosti jednog područnog znanja, pouzdano procijeniti učenikove performanse na drugom područnom znanju, ali je to moguće napraviti unutar istog područnog znanja. Upravo se na ovoj činjenici temelji prilagodljivost koju realiziramo u modelu prilagodljivog stjecanja znanja u sustavima e-učenja.

Prilikom određivanja načina prilagođavanja, potrebno je voditi računa o: sustavu prilagođavanja (GDJE), ciljevima prilagođavanja (ZAŠTO); značajkama učenika prema kojima se prilagođava (PREMA ČEMU); tehnologijama prilagođavanja (ŠTO – sadržaj ili navigacija); metodama i tehnikama prilagođavanja (KAKO) (Brusilovsky, 1996).

*Inteligentni tutorski sustavi* su trebali preuzeti vodstvo među prilagodljivim sustavima e-učenja. Međutim, do toga nije došlo zbog njihove nefleksibilnosti i troškova razvoja (Mohan, Greer, & McCalla, 2003). Da bi se uspješno realizirao prilagodljiv, dinamičan, personaliziran računalom oblikovani nastavni sadržaj u ITS-ovima potrebno je definirati arhitekturu koja osigurava jasnu razliku između „stroja za prilagođavanje“ koji dinamički generira računalom oblikovani nastavni sadržaj, od sadržaja od kojih je personalizirani računalom oblikovani nastavni sadržaj generiran (Dagger, Wade, & Conlan, 2005).

Nadalje, postoje brojne klasifikacije kognitivnih stilova i stilova učenja, kao i kontradiktorni dokazi o tome koji je pristup najbolji. Jasno je da učenici pokazuju preference prema jednom

načinu učenja i poučavanja, ali nije jasno koliko je ta preferenca stabilna i pouzdana (Valley, 1997, prema (Papanikolaou, Grigoriadou, Kornilakis & Magoulas, 2003)). Također Brusilovsky, iako veliki zagovornik i začetnik ideje o prilagodljivoj hipermediji, navodi da je informacija o učeniku, dobivena putem provjeravanja znanja učenika, pouzdanija od informacije do koje se došlo zaključivanjem na osnovu navigacije učenika (Brusilovsky, 1996). Stoga smo se odlučili za prilagođavanje prema znanju, jer ono pruža nedvosmislenu i postojanu osnovicu za koju postoji taksonomija znanja kao uporište.

U našem pristupu smo se opredijelili za realizaciju prilagodljivosti u inteligentnim tutorskim sustavima (GDJE) radi povećanja njihove učinkovitosti (ZAŠTO). Uzimajući u obzir aktualnu razinu znanja učenika (PREMA ČEMU), prilagođava se sadržaj (ŠTO), korištenjem Bloomove taksonomije znanja, Bayesovih mreža i stereotipova učenika (KAKO). Objasnimo sada najvažnije značajke pristupa realizacije prilagodljivosti koja je predložena u ovoj doktorskoj disertaciji.

## **2.3 Odrednice modela**

Novi model dinamičkog i prilagodljivog generiranja *računalom oblikovanog nastavnog sadržaja* (eng. *courseware*) u inteligentnim tutorskim sustavima uvažava aktualnu razinu znanja učenika i njegove kognitivne značajke koje određuju složenost, odnosno razinu elemenata računalom oblikovanih nastavnih sadržaja koji će se učeniku prezentirati. Prednost našeg pristupa je u tome što se računalom oblikovani nastavni sadržaj prilagođava učeniku prema njegovom znanju, a ne prema stilu učenja.

Svako područje ljudskog djelovanja može se prikazati skupom pravilno povezanih koncepata. Prezentiranje svakog koncepta, kao i odnosa između koncepata, ne može se prilagoditi svim stilovima učenja. Pošto je učenikovo znanje glavni razlog postojanja sustava e-učenja općenito, usmjerili smo naše istraživanje prema prilagođavanju računalom oblikovanih nastavnih sadržaja u inteligentnim tutorskim sustavima, odnosno proučavamo nizanje sadržaja i složenost prilagođavanja (ŠTO), a ne način prilagodljivog prezentiranja (KAKO), što je karakteristično za sustave prilagodljive hipermedije.

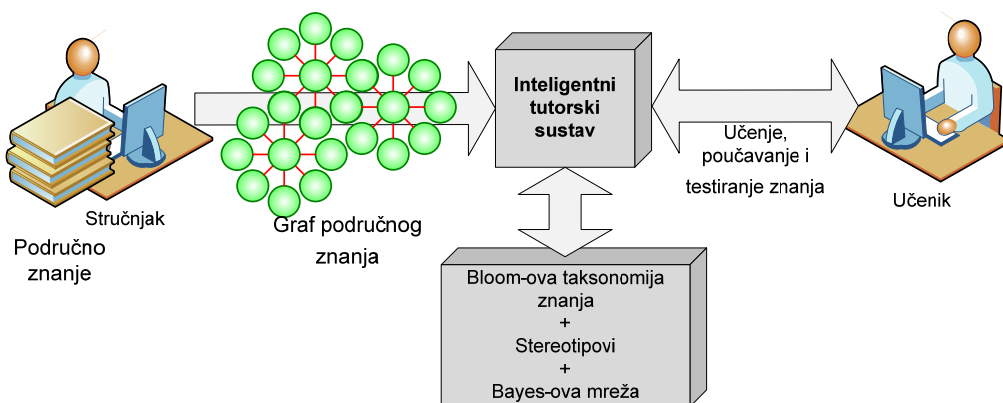
Sustavi prilagodljive hipermedije, kao najpoznatiji predstavnici sustava koji se prilagođavaju učeniku, temelje svoj mehanizam za prilagođavanje na statičkim sadržajima kojima je najmanji element raspoloživ prilagođavanje tek dio stranice. Prilagođavanje ne zadire u mijenjanje samog sadržaja stranica, već može jedino prikazati ili ne prikazati neke njene dijelove. Naime, sustavi prilagodljive hipermedije ne mogu sami stvarati nove stranice, već samo mogu na stranicama određene dijelove prikazati ili ne prikazati učeniku. Dakle, ono što prikazuju (dio stranice) uvijek ima istu formu i sadržaj koji ovisi o autoru stranice.

Tvrdimo da prilagođavanje opisano u novom modelu pridonosi boljem stjecanju znanja jer je najmanji element raspoloživ za prilagođavanje koncept znanja, dakle, „atomska čestica“ znanja i u tom smislu nedjeljiva. Uspješnost prilagodljivosti je obrnuto proporcionalna najmanjem elementu raspoloživom za prilagođavanje. Pošto smo se u novom modelu dotakli najmanjeg elementa za prilagođavanje, smatramo da našim pristupom realiziramo maksimalnu prilagodljivost.

U suglasju s postignutim spoznajama i dogovorima, a u svrhu realizacije što kvalitetnijeg obrazovanja, pristupili smo razvoju ideje o inteligentnom poučavanju temeljenom na *modelu automatskog i dinamičkog generiranja i prilagodljivog odabira, nizanja i prezentiranja* računalom oblikovanih nastavnih sadržaja. Zamisao modela dinamičkog i prilagodljivog generiranja računalom oblikovanih nastavnih sadržaja u inteligentnim tutorskim sustavima opisana je u nastavku poglavlja.

Većina ITS-ova omogućava okruženje za učenje koje se temelji na „slobodnom“ pristupu učenju, tj. učenici samostalno biraju svoju putanju učenja (eng. learning path) u računalom oblikovanom nastavnom sadržaju. Na taj način mogu preskočiti usvajanje pojedinih elemenata računalom oblikovanog nastavnog sadržaja ili učiti nešto pogrešnim redoslijedom ili učiti nešto što nema njima odgovarajuću težinu (ili preteško ili prelagano). To je razlog zašto ITS-ovi moraju učenika sami voditi u procesu učenja i poučavanja i prezentirati im samo onaj računalom oblikovani nastavni sadržaj koji ima prikladni obim i težinu. Stoga ITS-ovi, prilikom generiranja, odabira, nizanja i prezentiranja sadržaja za učenje učeniku, trebaju voditi računa o aktualnom znanju učenika i o složenosti elemenata računalom oblikovanog nastavnog sadržaja. Ovakvim pristupom će se smanjiti kognitivno preopterećivanje te će se omogućiti individualizirano vođenje procesa učenja i poučavanja.

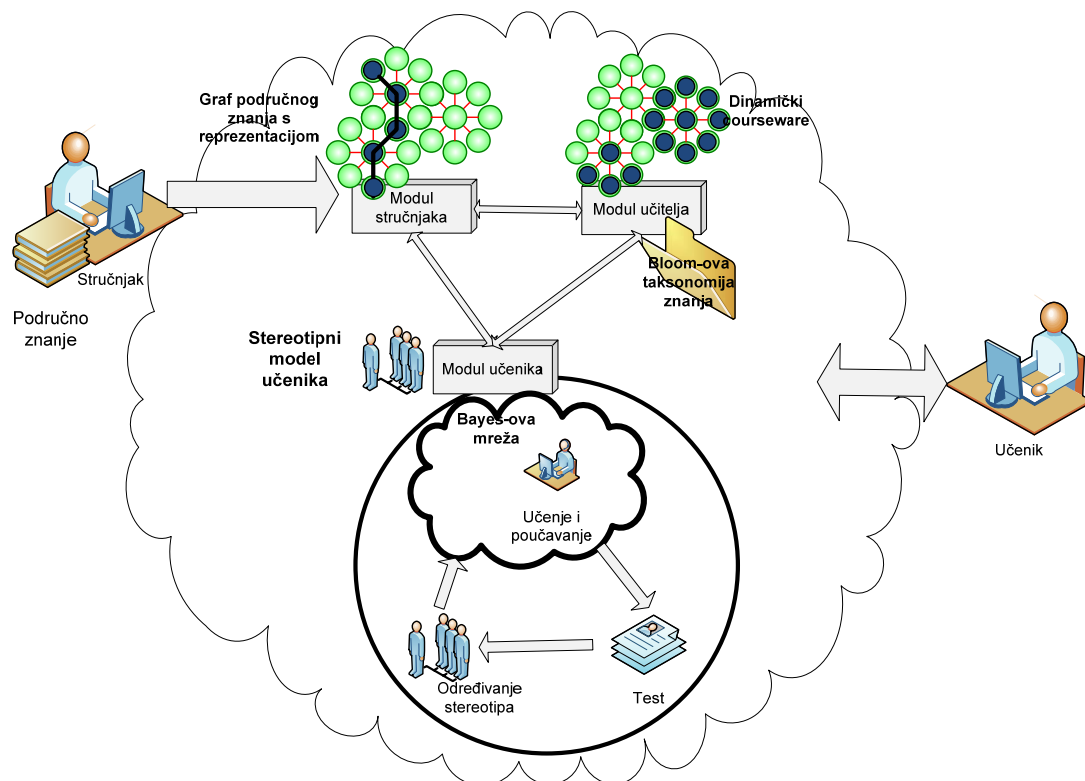
Predloženi model prilagodljivoga stjecanja znanja u ITS-ovima zasniva se na automatskom i dinamičkom generiranju i prilagodljivom odabiru, nizanju i prezentiranju računalom oblikovanog nastavnog sadržaja. Okruženje dinamičkog i prilagodljivog generiranja računalom oblikovanog nastavnog sadržaja u inteligentnim tutorskim sustavima prikazano je na Slika 2.2.



Slika 2.2. Okruženje dinamičkog i prilagodljivog generiranja računalom oblikovanog nastavnog sadržaja u inteligentnim tutorskim sustavima

Automatsko generiranje računalom oblikovanog nastavnog sadržaja u novom modelu označava da elemente računalom oblikovanog nastavnog sadržaja za učenje i provjeravanje znanja stvara sam sustav na temelju ontologije područnog znanja. Dinamičko generiranje računalom oblikovanog nastavnog sadržaja označava da se ti sadržaji stvaraju u trenutku izvođenja. Prilagodljivi odabir, nizanje i prezentiranje računalom oblikovanog nastavnog sadržaja se odvija automatski i dinamički u skladu s modelom učenika (stereotip i Bayes) uporabom predložaka za tvrdnje.

Pogledajmo sada strukturu novog modela koja je prikazana na Slika 2.3. Modul stručnjaka se temelji na formalnom ontološkom opisu područnog znanja koji omogućava definiranje reprezentacije područnog znanja, modul učenika na stereotipovima definiranim prema Bloomovoj taksonomiji znanja (Bloom, 1956) i na Bayesovim mrežama predikcije znanja (Conati, Gertner, Vanlehn, & Druzdzel, 1997), a modul učitelja na Bloomovoj taksonomiji znanja prilagođenoj hrvatskoj pedagoškoj praksi te na predlošcima tvrdnji i pitanja. Modul komunikacije u ovom modelu ne definiramo jer se ovakav pristup može realizirati u bilo kojem ITS-u koji ima znanje zasnovano nad ontologijom, neovisno o načinu komunikacije između učenika i sustava.

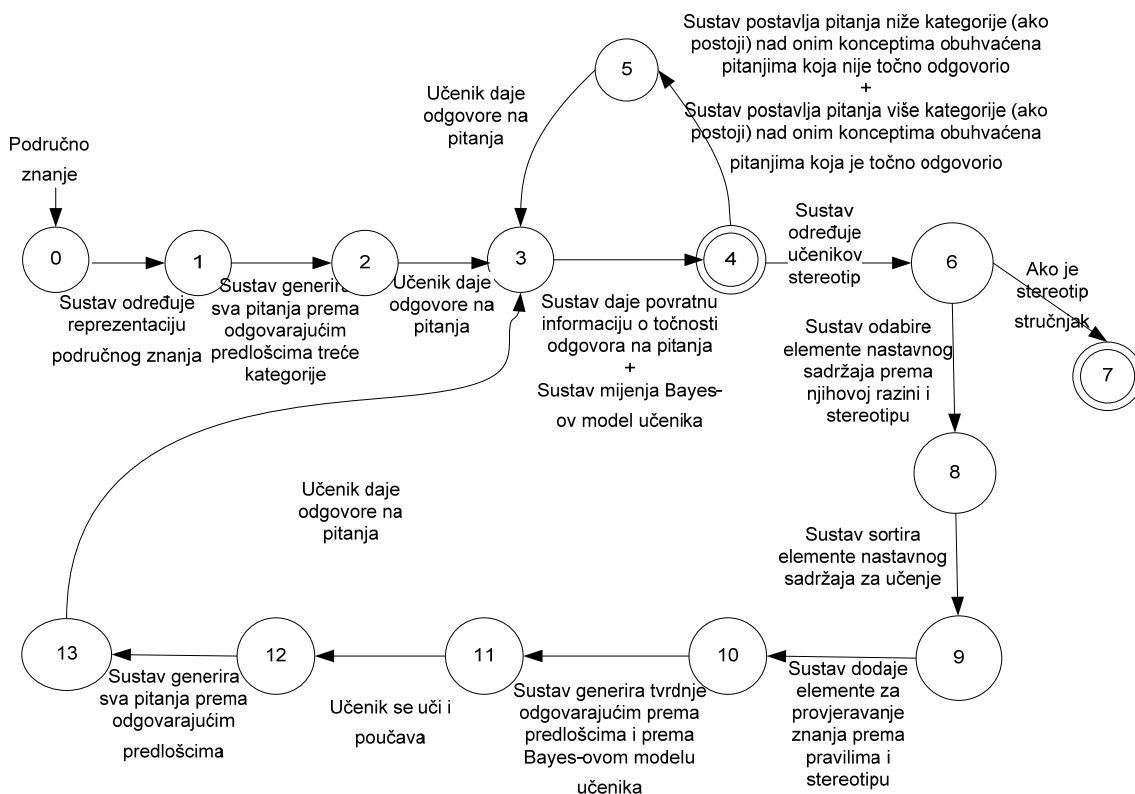


Slika 2.3. Struktura novog modela

U modulu stručnjaka se povezanost koncepata pokazuje relacijama unutar područnog znanja, što je bitna informacija koja modulu učitelja omogućava generiranje i prezentiranje računalom oblikovanog nastavnog sadržaja. Računalom oblikovani nastavni sadržaji je opisan nizom elemenata pri čemu svaki element sadrži povezani podskup koncepata

područnog znanja. Prilikom generiranja elemenata računalom oblikovanog nastavnog sadržaja, modul učitelja proučava zavisnost koncepata te na toj osnovi ih povezuje unutar elementa nastavnog sadržaja. Naposljetku, modul učitelja niže elemente računalom oblikovanog nastavnog sadržaja kako bi se definirala struktura tih sadržaja. Modul učitelja je odgovoran i za vođenje procesa učenja, poučavanja i testiranja znanja, pri čemu se testiranje realizira zadacima objektivnog tipa. Za vrijeme testiranja, modul učitelja surađuje s modulom stručnjaka zbog generiranja pitanja i analiziranja učenikovih odgovora. Također, uz pomoć modela učenika, modul učitelja planira daljnje akcije koje utječu na vođenje procesa učenja, poučavanja i testiranja znanja.

Navedena tri modula su u interakciji (stroj s konačnim brojem stanja Slika 2.4.) da bi se realizirala prilagodljivost na različitim aspektima procesa učenja i poučavanja: prilagođava se struktura računalom oblikovanog nastavnog sadržaja (odabir i nizanje) na temelju učenikova stereotipa, prilagođava se prezentiranje računalom oblikovanog nastavnog sadržaja na temelju Bayesove mreže i stereotipa (predlošci za tvrdnje), prilagođava se provjeravanje znanja učenika na temelju njegovog stereotipa (predlošci za pitanja).



Slika 2.4. Interakcija u modelu prilagodljivog stjecanja znanja

U sljedećem potpoglavlju prikazat ćemo detaljnije strukturu novog modela prilagodljivog stjecanja znanja. Za svaku komponentu sustava navodimo teorijsku podlogu nad kojom se ona zasniva, kao i formalizaciju koja opisuje specifičnosti našeg pristupa u modelu prilagodljivog stjecanja znanja.



## **2.4 Elementi strukture**

Izložena je cjelokupna zamisao modela automatskog, dinamičkog i prilagodljivog generiranja računalom oblikovanog nastavnog sadržaja u inteligentnim tutorskim sustavima zasnovanog na stereotipovima, Bayesovim mrežama i Bloomovoj taksonomiji znanja, namijenjenog učenju, poučavanju i testiranju znanja učenika kojeg smo nazvali *Adaptive Courseware Tutor Model (AC-ware Tutor)*.

U daljnjim podpoglavljima ovog poglavlja navodi se teorijska podloga za implementaciju i postavljanje pojedinih modula kao i njihovih skupova podataka unutar novog modela. S tim u vezi, zamisao predloženog modela se dalje dekomponira do razine modula i obuhvaća: modul stručnjaka (područno znanje), modul učitelja (računalom oblikovani nastavni sadržaj) i modul učenika (modeliranje učenika). Na ovaj način su postavljeni temelji dinamičkog i prilagodljivog generiranja računalom oblikovanog nastavnog sadržaja u sustavu *AC-ware Tutor* zasnovanog na stereotipovima, Bayesovim mrežama i Bloomovoj taksonomiji znanja. Poblje ćemo objasniti proces generiranja elemenata računalom oblikovanog nastavnog sadržaja, odabira i nizanja tih elemenata, te prezentiranja spomenutih elemenata, pojedinačno kroz komponente sustava.

### **2.4.1 Modul stručnjaka**

Osnovne vrste znanja kojim se opisuje određeno područno znanje su proceduralno i deklarativno znanje. Za formalizaciju proceduralnog znanja koriste se produkcijska pravila, dok se za deklarativno znanje koristi formalni jezik simbola ili neka druga shema prikazivanja znanja.

#### **2.4.1.1 Teorijska podloga modula stručnjaka**

Danas se često za formalizirani prikaz znanja u ITS-ovima koristi ontologija (Lee, Hendler, & Lassila, 2001). Ontologijom se opisuje konceptualni model nekog područja, odnosno objekti, koncepti i drugi entiteti za koje se smatra da postoje te relacije među njima (Genesereth i Nilsson, 1987, prema (Gruber, 1993)). Osnovni strukturni elementi konceptualnog modela su koncepti i relacije. Najpoznatija definicija ontologije je: „*An ontology is an explicit specification of a conceptualization*“, odnosno, ontologija je eksplicitna specifikacija konceptualizacije (Gruber, 1993).

Dakle, svako područje ljudskog djelovanja može se prikazati skupom pravilno povezanih koncepata koji odgovara područnom znanju. Ontološki pristup u opisivanju područnog znanja omogućava jednostavnu formalizaciju deklarativnog znanja korištenjem različitih alata koji podržavaju rad s konceptima i relacijama.

Formalni ontološki opis područnog znanja je ulazni skup podataka modela stručnjaka. Na osnovu ovog opisa modul stručnjaka formira područno znanje. U modulu stručnjaka se analizira ontološki opis područnog znanja radi određivanja međusobne povezanosti koncepata.

Da bi bila strojno čitljiva, ontologija mora biti prikazana jezikom koji pruža sustav kodiranja pogodan za strojnu obradu. Jezici koji se koriste u prikazu ontologija se dijele na: jezike koji se temelje na logici prvog reda (KIF - Knowledge Interchange Format), jezike koji se temeljena logici sustava okvira (KL-ONE - Knowledge Language One) i mrežni jezici (OWL - Web Ontology Language). OWL je jezik za semantičko kodiranje razvijen od strane W3C (<http://www.w3.org/TR/owl-ref/>) u svrhu kreiranja i razmjene ontologija na Webu. Standard je namijenjen programskim aplikacijama, koje umjesto jednostavne prezentacije informacija za ljudsko korištenje, trebaju obrađivati sadržaj informacija. OWL omogućava strojnu razmjenu Web sadržaja na razini koja je naprednija od one podržane jednostavnim XML, RDF ili RDFS shemama.

#### 2.4.1.2 Formalizacija modula stručnjaka u sustavu *AC-ware Tutor*

U našem pristupu koristimo područno znanje zasnovano nad ontologijom. Od prethodno spomenutih kategorija jezika odabrali smo OWL DL (Web Ontology Language Description Logic) jer je to standardizirani jezik za opisivanje ontologija na Web-u temeljen na  $\mathcal{SHOIN}^{(D)}$  familiji opisne logike koja posjeduje većinu elemenata predikatne logike prvog reda. Opisna logika ujedinjuje prednosti grafičke notacije s formalnom logikom koja je podskup predikatne logike prvog reda (Žitko, 2010).

Područno znanje je prikazano konceptima i relacijama među njima. Pošto trebamo naznačiti smjer povezanosti koncepata, koristimo pojmove podkoncept i nadkoncept. Da bismo za svaku relaciju u ontologiji nedvosmisleno naznačili koje koncepte povezuje i kakav je odnos između tih koncepata uvodimo sljedeću definiciju.

**Definicija 2.1** Neka je  $E_{KCP} = \{K_1, \dots, K_n\}$ ,  $n \geq 0$ , skup koncepata,  $E_{REL} = \{r_1, \dots, r_m\} \cup \{ima_{podvrstu}, ima_{primjerak}, ima_{dio}, slot, filler\}$ ,  $m \geq 0$ , skup relacija i  $\emptyset_E$  prazni element područnog znanja.

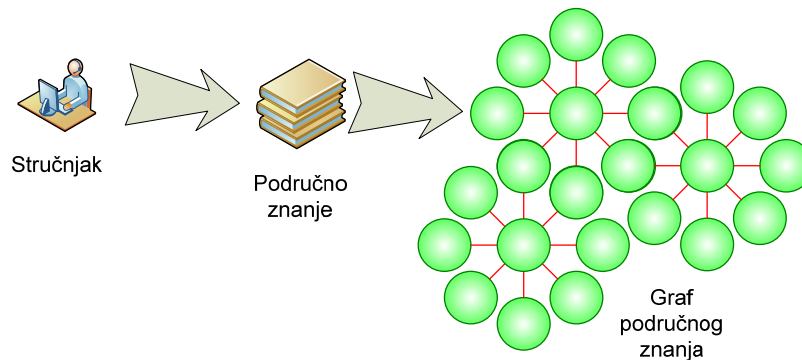
**Područno znanje** je skup uređenih trojki  $(K_1, r, K_2)$  koje definiraju da su koncepti  $K_1$  i  $K_2$  povezani relacijom  $r$ . U ovako definiranoj uređenoj trojci koncept  $K_1$  je **nadkoncept** koncepta  $K_2$ , tj. koncept  $K_2$  je **podkoncept** koncepta  $K_1$ .

Relacija *isa*, odnosno *ima\_podvrstu*, služi za hijerarhijsku kategorizaciju koncepata i predstavlja generalizaciju koja omogućava nasljeđivanje svojstava između nadkoncepta i podkoncepta. Relacija *inst*, odnosno *instance\_of*, odnosno *je\_primjerak*, služi za povezivanje koncepta koji predstavlja klasu s konceptom koji predstavlja instancu te klase. Relacija *part*,

odnosno *part\_of*, odnosno *ima\_dio*, služi za strukturalno raščlanjivanje koncepta na njegove dijelove.

Pošto su osnovni elementi trojki područnog znanja koncepti i relacije među njima, koristimo *teoriju grafova* kao matematičku podlogu koja omogućava upravljanje podskupovima i elementima područnog znanja, kao i vizualizaciju područnog znanja ((Gross & Yellen, 1998), (Veljan, 1989)). Zato definiramo usmjereni *graf područnog znanja* (Definicija 2.2) za kojeg vrijede sve zakonitosti iz teorije grafova.

**Definicija 2.2** Za područno znanje PZ definiramo usmjereni **graf područnog znanja**  $GPZ = (V, A)$  gdje je skup vrhova  $V = E_{KCP}$ , a skup bridova  $A = \{(K_1, K_2) | \exists (K_1, r, K_2) \in PZ, r \neq \emptyset_E, K_1 \neq K_2\}$  jednak skupu svih uređenih parova onih konceptata iz područnog znanja koji su povezani nekom relacijom.



Slika 2.5. Od područnog znanja do grafa područnog znanja

Ako postoji brid  $br=(K_1, K_2)$  između konceptata  $K_1$  i  $K_2$ , onda to kraće pišemo  $K_1K_2$  ili samo  $br$ . Navodimo nekoliko definicija kojima pobliže određujemo graf područnog znanja.

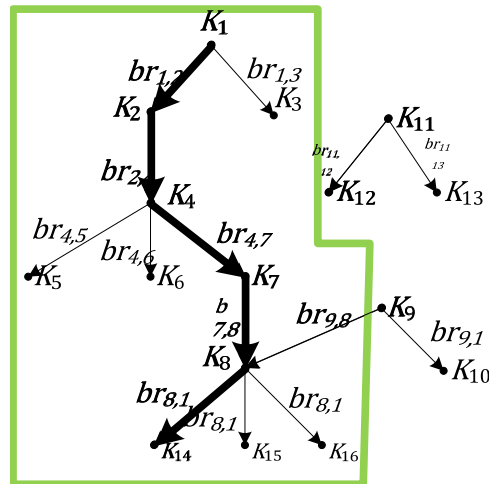
**Definicija 2.3** Skup **nadkonceptata** koncepta  $K_x$  je skup  $NadK_x = \{K \in E_{KCP} | \exists (K, r, K_x) \in PZ, K \neq K_x, r \neq slot, filler, \emptyset_E\} = \{K \in V | \exists (K, K_x) \in A, K \neq K_x\}$ . **Broj nadkonceptata** koncepta  $K_x$   $nK_x$  je broj elemenata skupa  $NadK_x$ .

**Definicija 2.4** Skup **podkonceptata** koncepta  $K_x$  je skup  $PodK_x = \{K \in E_{KCP} | \exists (K, r, K_x) \in PZ, K \neq K_x, r \neq slot, filler, \emptyset_E\} = \{K \in V | \exists (K_x, K) \in A, K \neq K_x\}$ . **Broj podkonceptata** koncepta  $K_x$   $pK_x$  je broj elemenata skupa  $PodK_x$ .

**Definicija 2.5** Vrh grafa područnog znanja nazivamo **korijen** ako nema nadkonceptata, a ima podkonceptata.

**Definicija 2.6** Definiramo **cjelinu**  $C_i$  u grafu GPZ kao najveći podgraf koji ima samo jedan korijen, tj. cjelina je graf čiji skup vrhova sadrži samo jedan korijen. **Put**  $K_{C_i}MaxVrh_{C_i}$  je najveći put u nekoj cjelini, a **Maxlen** $_{C_i}$  je duljina tog puta. **MaxVrh** $_{C_i}$  je korijen cjeline koji se naziva **centralni vrh cjeline**  $C_i$ .

Centralni vrh nema prethodnika i od njega vodi put do svih ostalih vrhova. Bilo koji vrh u cjelini, osim centralnog vrha, ima jednog ili više neposrednih prethodnika. Cjelina je, dakle povezani podgraf u kojem je svaki vrh povezan s centralnim vrhom (osim njega samog) najmanje jednim putem. U cjelini nema izoliranih vrhova. Dakle, najmanji podskup područnog znanja je cjelina koja je definirana korijenom. Odnosno, svaki korijen u područnom znanju definira jednu cjelinu (Slika 2.6.).



Slika 2.6. Cjelina u područnom znanju i najveći put

Pošto područno znanje može biti obimno, potrebno je definirati takav podskup područnog znanja koji će dostojno reprezentirati cijelo područno znanje. Dakle, reprezentacijom područnog znanja obuhvaćamo sve koncepte i relacije koji su relevantni za neko područno znanje. Nadalje ćemo pristupiti definiranju reprezentacije područnog znanja kao skupa relevantnih koncepata i relacija.

### **Specifikacija reprezentacije područnog znanja**

Reprezentacija područnog znanja će omogućiti inicijalizaciju modela učenika bez koje ne bi bilo moguće realizirati prilagodljivost prema znanju. Naime, bez inicijalizacije modela učenika koje će se realizirati ulaznim testom znanja nad reprezentacijom područnog znanja, sam proces učenja i poučavanja bi bio upitan. To je kao da liječnik pacijentu da lijek bez da je prvo utvrdio simptome njegove bolesti (Glaser & Nitko, 1970).

Postoje dva pristupa u određivanju reprezentacije područnog znanja: određivanje reprezentacije temeljem semantičke analize koncepata i relacija u područnom znanju, te određivanje reprezentacije temeljem matematičkih metoda iz teorije grafova. Oba pristupa imaju svoje prednosti i nedostatke.

Ako se odlučimo za *semantičku analizu*, onda nam za određivanje reprezentacije treba određeni broj stručnjaka iz nekog područnog znanja koji će odrediti koji koncepti i relacije upravo zbog svog značenja moraju pripadati reprezentaciji tog područnog znanja. Ovaj

pristup se temelji na heuristikama i stručnosti onih osoba koje smo odabrali za ovaj posao. Odabir reprezentacije je „životniji“ i „stvarniji“ i bolje ogovara situaciji jer se odabir radi drugačije za svako područno znanje, ali su problemi upravo u nejedinstvenosti.

Da bismo to pokazali, pristupili smo istraživanju o promišljanjima 7 stručnjaka iz područnog znanja „Računalo kao sustav“ o tome što bi bila reprezentacija tog područnog znanja. Stručnjaci su odabrali reprezentacije od 7 do 31 koncepta bez obzira na to da li su povezani ili nisu. Utvrdili smo da su se samo dva koncepta našla u svih 7 reprezentacija, četiri koncepta su se našla u njih 5, sedam konceptata u 4 reprezentacije. Cijela analiza istraživanja može se vidjeti u Prilogu 8.3.

Upravo je raznolikost semantičkog viđenja reprezentacije pokazala nužnost za pronalaženjem mehanizama za ne-semantičko *matematičko određivanje* reprezentacije neovisno o područnom znanju. Zato u našem pristupu ne promatramo semantiku konceptata i relacija, već isključivo koristimo matematičke metode iz teorije grafova. Ovakve reprezentacije djeluju prilično „krute“ i „ograničene“ upravo zato što se ista metoda odabira koristi za bilo koje područno znanje. U protivnom bismo za svako novo područno znanje koje uvodimo u sustav e-učenja, morali prvo napraviti višestruku semantičku analizu konceptata i relacija od strane određenog broja stručnjaka iz tog područnog znanja, a zatim u više navrata ponavljati odabir reprezentacije dok se ne dobije ona koja će zadovoljiti sve stručnjake.

Pošto smo utvrdili da zbog fleksibilnosti prema različitim područnim znanjima, u našem pristupu definiranju reprezentacije područnog znanja nećemo uzimati u obzir semantiku konceptata već isključivo njihov položaj u grafu područnog znanja o kojem ovisi da li će oni pripadati reprezentaciji ili ne, potrebno je odrediti matematičke metode kojima ćemo dobiti takvu reprezentaciju.

Naša ideja za definiranje reprezentativnog podgrafa područnog znanja je potvrdu našla u područja društvenih mreža, mreža citiranosti i komunikacijskih mreža upravo zbog njihove veličine. U ovim područjima se pristupa tzv. „uzorkovanju“ *grafa* (eng. graph sampling) ili  *smanjenju grafa* (eng. graph reduction), tj. pristupa se definiranju reprezentativnog uzorka grafa. Dva autora govore o ovom problemu i svaki od njih ima svoju podjelu metoda. Zajedničko i jednom i drugom autoru je to što sličnu metodu smatraju najuspješnijom.

*Prvi autor* (Leskovec & Faloutsos, 2006) dijeli metode za uzorkovanje na tri grupe: metode koje se temelje na slučajnom odabiru vrhova, slučajnom odabiru bridova i na tehnike istraživanja koje simuliraju slučajne šetnje ili širenje virusa.

Pošto u našem pristupu ne želimo slučajno odabirati vrhove i bridove jer želimo da reprezentativni podgraf bude povezan, odlučili smo se na uzorkovanje istraživanjem. Uobičajen pristup kod uzorkovanja istraživanjem je slučajan odabir prvog vrha, a zatim istražiti vrhove u njegovoj blizini.

U ove metode spadaju: *susjedstvo slučajnog vrha* (eng. Random Node Neighbor (RNN)), *slučajna šetnja* (eng. Random Walk (RW)), *slučajni skok* (eng. Random Jump (RJ)) i *šumski požar* (eng. Forest Fire (FF)). Metoda *susjedstvo slučajnog vrha* slučajno odabire vrh i sve njegove podvrhove. Kod *slučajne šetnje* slučajno biramo početni vrh i simuliramo slučajnu šetnju po grafu dok ne postignemo potpuno povezani uzorak željene veličine. *Slučajni skok* je sličan slučajnoj šetnji jer kreće od slučajno odabranog vrha od kojeg započinjemo šetnju, ali omogućava i „skakanje“ na drugi slučajno odabrani vrh od kojeg nastavljamo šetnju sve dok ne postignemo uzorak željene veličine. Kod ove metode uzorak nije potpuno povezan. Kod *šumskog požara* slučajno biramo početni vrh od kojeg započinjemo s „paljenjem“ svih izlaznih bridova prema njegovim podvrhovima na kojima nastavljamo s „paljenjem“ bridova.

Istraživanja su pokazala da se najbolje metode uzorkovanja temelje na *slučajnim šetnjama* (eng. random walk) i „*šumskim požarima*“ (eng. forest fire) koje definiraju reprezentativne uzorke veličine 15% originalnog grafa. Slučajne šetnje se koriste kad želimo da uzorak ima slična svojstva kao i originalni graf (eng. scale-down goal) i kada želimo da uzorak bude povezan. „Šumske požare“ koristimo kada želimo da uzorak bude sličan onome kakav je originalni graf bio kada je bio veličine uzorka (eng. back-in-time goal), odnosno promatramo vremenski razvoj grafa (Leskovec & Faloutsos, 2006).

*Drugi autor* (Krishnamurthy et al., 2005) metode za smanjenje grafa dijeli u tri kategorije: (i) metode brisanja (eng. deletion methods) koje brišu vrhove ili bridove iz grafa dok se ne postigne željena veličina, (ii) metode kontrakcije (eng. contraction methods) koje kontrahiraju susjedne vrhove dok se ne postigne željena veličina, (iii) metode istraživanja (eng. exploration methods) prelaze preko određenog broja vrhova zadržavajući na taj način svojstva originalnog grafa.

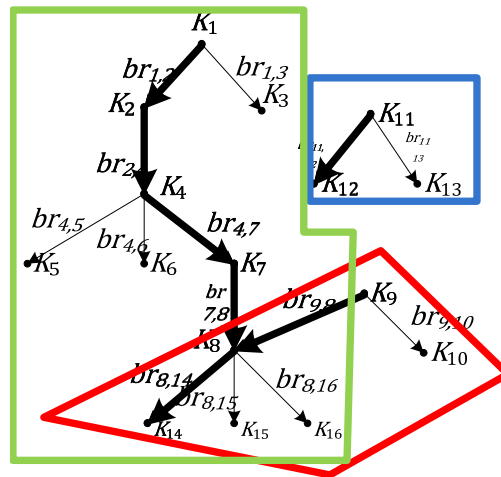
Pošto u našem pristupu želimo sačuvati svojstva originalnog grafa i želimo da reprezentativni podgraf bude povezan, odlučili smo se na metode istraživanja (primijetimo da smo i kod prvog autora odabrali uzorkovanje istraživanjem). Uobičajen pristup kod metoda istraživanja je slučajan odabir prvog vrha, a zatim prelazimo preko grafa prema zadanoj metodi istraživanja.

U ove metode spadaju: *istraživanje po širini* (eng. Exploration by Breadth First Search (EBFS)) i *istraživanje po dubini* (eng. Exploration by Depth First Search (EDFS)). Metoda *istraživanje po dubini* vrši pretraživanje po dubini od slučajno odabranog početnog vrha sve dok ne postignemo uzorak željene veličine. Metoda *istraživanje po širini* vrši pretraživanje po širini od slučajno odabranog početnog vrha (u širinu pretražuje po slučajno odabranom neposječenom bridu) sve dok ne postignemo uzorak željene veličine.

Istraživanja su pokazala da se ovim algoritmima može smanjiti veličina grafa za 70% vrhova uz zadržavanje svojstava originalnog grafa (Krishnamurthy et al., 2005).

Pošto želimo da nam uzorak bude povezani podgraf koji ima slična svojstva kao cijeli graf područnog znanja (reprezentira područno znanje), koristimo kombinaciju *slučajne šetnje* i

*istraživanja po dubini*. Neka je  $G$  graf područnog znanja koji ima  $n$  vrhova. Cilj je stvoriti reprezentativni podgraf  $S$  sa  $n_0$  vrhova,  $n_0 < n$ , koji će biti najbliži grafu  $G$ , tj. želimo da  $S$  ima slična svojstva kao i  $G$ . U našem pristupu tražimo da *slučajna šetnja* bude najduža šetnja od odabranog početnog vrha (*istraživanje po dubini*). Za svaku cjelinu grafa područnog znanja odabiremo njen centralni vrh  $MaxVrh_{C_i}$  kao početni vrh *slučajne šetnje*, odnosno za svaku cjelinu kombinacija *slučajne šetnje* i *istraživanja po dubini* će rezultirati sa  $PutK_{C_i}MaxVrh_{C_i}$  (Definicija 2.6).



Slika 2.7. Reprezentativni podgraf područnog znanja

Na temelju navedenog, uvodimo sljedeće definicije reprezentacija:

**Definicija 2.7 Reprezentativni podgraf  $RepGPZ$**  usmjerenog grafa područnog znanja je unija najdužih puteva u svim cjelinama grafa GPZ, tj.  $RepGPZ = \bigcup_{i=1}^n PutK_{C_i}MaxVrh_{C_i}$ .

**Definicija 2.8 Reprezentacija područnog znanja  $RepPZ$**  je podskup područnog znanja takav da vrijedi  $RepPZ = \{(K_1, r, K_2) \in PZ \mid K_1, K_2 \in V_{RepGPZ}\}$ .

U modulu stručnjaka je izveden algoritam za reprezentaciju područnog znanja radi određivanja koncepata područnog znanja na temelju kojih će se generirati ulazni test kojim se provjerava predznanje učenika o nekom područnom znanju. Ovako određeni ulazni test garantira da će ispitivanje znanja učenika obuhvatiti sve koncepte i relacije koji su relevantni za neko područno znanje. Ulaznim testom se inicijalizira model učenika opisan u sljedećem poglavlju.

## 2.4.2 Modul učenika

Glavna pretpostavka na kojoj se temelje inteligentni tutorski sustavi je da je svaki učenik jedinstven. Ova paradigma se temelji na stvaranju modela učenika koji pamti učenikove

sklonosti i napredak tijekom učenja i poučavanja. Iako su sve komponente ITS-a važne, jasno je da je model učenika od presudne važnosti. Ako je model učenika „loš“ do te mjere da ni približno ne opisuje značajke nekog učenika, onda će i odluke ostalih komponenti ITS-a koje se zasnivaju na tom modelu biti loše kvalitete. Zato se značajna istraživanja provode upravo u području modeliranja učenika. Glavna karakteristika koja povećava učinkovitost učenja i poučavanja pomoću ITS je upravo prilagođavanje aktualnoj razini znanja učenika.

Karakteristika većine ITS-ova je da zaključuju o tome kako učenik razumije dane sadržaje i to iskoriste da bi poučavanje prilagodili učenikovim potrebama. Ovakvo zaključivanje još se zove i postavljanje dijagnoze. Dijagnostička komponenta ITS-a nam otkriva skriveno stanje spoznaje učenika na temelju promatranog ponašanja. On mora zaključiti što je učenik mislio i čime se bavio tijekom učenja. Trenutačno stanje učenikova znanja predstavljeno je modelom učenika. Učenikov model je skup podataka, a dijagnoza je proces manipulacije tim podacima.

### **2.4.2.1 Teorijska podloga modula učenika**

U ovom dijelu pobliže ćemo objasniti pristupe modeliranju učenika čije elemente koristimo u novom modelu prilagodljivog stjecanja znanja. Objasniti ćemo što je to model prekrivanja, što su stereotipovi i kako ih koristimo, te što se postiže primjenom Bayesovog probabilističkog modela učenika.

Problem u modeliranju učenika postoji i kod prikazivanja informacija u modelu. Uobičajeno je da ITS-ovi koriste jedan oblik jednostavnog *modela prekrivanja* (eng. *overlay model*) (Carr & Goldstein, 1977) kod kojeg se učenikovo znanje prikazuje kao podskup znanja sustava. Međutim, jednostavni model prekrivanja ne može predviđati što učenik može znati na temelju djelomičnih informacija. *Hijerarhijski model prekrivanja* (eng. *hierarchical overlay model*) korišten u sustavu za pomoć pri radu s Unix-om USCHS (Matthews i Biswas, 1986, prema (Chin, 1989)) može predviđati učenikovo poznavanje čvorova roditelja u hijerarhiji promatrajući težinski zbroj (eng. *weighted sum*) procjena poznavanja čvorova djece. Ovakav pristup omogućava predviđanje tema koje su vrlo povezane.

Modeli korisnika se upotrebljavaju za prikazivanje njihovih prioriteta. Sustavi HAM\_ANS za rezervaciju smještaja u hotelu (Hoepfner, Morik i Marburger, 1984, prema (Chin, 1989)) i Real Estate Agent za pronalaženje nekretnina (Morik i Rollinger, 1985, prema (Chin, 1989)) biraju sobe ili stanove prema korisnikovim potrebama i sklonostima o čemu ih oni moraju informirati na početku rada u sustavu. Takvo davanje informacija na početku rada u sustavu nije moguće kod ITS-ova kod kojih učenici (korisnici za učenje) vjerojatno neće spomenuti svoj nivo znanja ili ga uopće nisu svjesni.

Da bi ustanovio što učenik zna, sustav ga može neprestano ispitivati, ali takav način ispitivanja zahtjeva dosta vremena i mnogim učenicima to stalno ispitivanje ne bi odgovaralo. Nasreću, učenici usvajaju područno znanje na predvidiv način, tako da sustav može dodatno zaključiti o tome što učenik vjerojatno zna ili ne zna na temelju parcijalnog



modela učenika. To znači da bi sustav za modeliranje učenika trebao moći predvidjeti njegovo znanje na temelju djelomičnih informacija o tome što učenik zna ili ne zna.

Zaključivanje o znanju učenika temeljeno na ograničenom prethodnom znanju je standardan način razmišljanja. To ne mora značiti da je takav način razmišljanja uvijek valjan, pa sustav mora voditi računa o činjenici da su takva zaključivanja neizvjesna. Obično ovo zahtijeva da sustav može procijeniti vjerojatnost svojih standardnih zaključaka. Također, kako sustav dobiva konkretnije informacije, on mora obnoviti svoj model učenika. Nove informacije mogu biti u kontradikciji s prethodnim predviđanjima, tako da održavanje modela nije jednostavno.

Problem se očituje i kod organiziranja zaključaka. Ljudi u istim fazama učenja imaju slično znanje. Jednom kad je sustav ustanovio učenikov nivo, on može pretpostaviti da taj učenik zna onoliko koliko i drugi učenici koji se nalaze na tom nivou. Dobar sustav za modeliranje učenika nastoji grupirati pretpostavke o nivou učenika na jednostavan i učinkovit način.

Ovakav način razmišljanja kod kojeg se učenike svrstava u kategorije prema nekom kriteriju (ili više njih) se naziva *stereotipiziranje*, odnosno pridjeljivanje stereotipa učeniku. Ova metoda je proizašla iz psihologije jer se smatra da ljudi koriste stereotipove (ili klastere ili karakteristike) da bi složenost svijeta u kojem žive pojednostavnili i klasificirali (Hamilton, 1979, prema (Rich, 1983)).

### **Stereotipovi**

Životne situacije nameću potrebu za stvaranjem modela osobe s kojom smo u kontaktu da bismo se mogli prikladno ponašati. Stvaramo modele osoba sakupljajući specifične dijelove informacija koji nam potom pobuđuju znanje koje imamo o grupi kojoj neka osoba pripada. *Stereotip* je uvriježeno mišljenje o specifičnim socijalnim grupama ili tipovima pojedinaca. On predstavlja skup standardiziranih i pojednostavljenih uvjerenja koja se temelje na pretpostavkama. Stereotip kao spoznajna sastavnica se može definirati kao generalizacija o grupi ljudi kojom se istovjetne osobine pripisuju gotovo svim članovima te grupe, neovisno o stvarnim varijacijama između članova. Riječ **stereotip** (grč. στερεότυπος) se sastoji od dva pojma: **stereo** (grč. στερεός (stereos)) što znači čvrst i **tip** (grč. τύπος (typos)) što znači uvjerenje. Dakle, stereotip je čvrsto uvjerenje. Sam pojam stereotip korijene vuče iz tiskarske tehnologiji gdje on predstavlja metalnu ploču odlivenu s matrice koja služi za višestruko tiskanje složenih tiskovnih oblika. Izraz je tako preuzeo i značenje uvijek istog postupka, rutine, onoga što je banalno i svakidašnje (Wikipedia, 2011).

Stereotipovi predstavljaju kolekcije *karakteristika ili aspekata* (eng. facet). Učenika se može opisati skupom karakteristika koje imaju svoje vrijednosti. Okidači (eng. trigger) su oni događaji čija prisutnost označava prikladnost određenog stereotipa. Npr. ako učenik upotrebljava složene mogućnosti sustava, ta bi spoznaja trebala biti okidač za stereotip „STRUČNJAK“ koji bi se onda trebao aktivirati. Kada je stereotip aktiviran, pretpostavke o

različitim karakteristikama učenika se uključuju u njegov model. Stereotipovi koji se koriste u računalnim sustavima variraju od vrlo općenitih, kao što su muškarac ili žena, do vrlo specifičnih (Rich, 1983).

Da bi računalni sustav mogao učinkovito koristiti stereotipove mora poznavati same stereotipove, kao i *okidače* stereotipova (eng. trigger). Okidač je objekt vezan za određenu situaciju. On sadrži naziv stereotipa kojeg aktivira i *procjenu istinitosti* (eng. rating) koja se pridjeljuje stereotipu. Procjena istinitosti se odnosi na vjerojatnost da je stereotip prikladan za određenu situaciju.

### **Dvostruki stereotipovi**

Dvostruki stereotipni model je korišten u sustavu Unix Consultant – UC (Chin, 1986). Stereotip učenika je određen prema različitim razinama stručnosti: novak (eng. novice), početnik (eng. beginner), osrednji (eng. intermediate), stručnjak (eng. expert).

Stereotip znanja je određen prema težini znanja na temelju tipične lokaciji na krivulji učenja (koja odgovara učenju prosječnog učenika), tj. na temelju činjenice kada će tipični učenik naučiti tu informaciju. Stereotipovi znanja su: jednostavno (eng. simple), srednje (eng. mundane), složeno (eng. complex), ezoterično (eng. esoteric).

Ezoterični su oni koncepti koji ne spadaju u nijednu od prve tri kategorije i obično su to koncepti koji imaju neku posebnu svrhu. Ona uključuje informacije koje nisu tipične da ih učenik nauči u bilo kojoj razini stručnosti. Oni se nauče jedino kada učenici imaju posebne potrebe. Neki takvi pojmovi mogu biti poznati početnicima, ali nepoznati stručnjacima.

Cijeli model učenika može se sažeti u sljedećih pet tvrdnji:

- stručnjak zna sve jednostavne ili srednje činjenice i većinu složenih činjenica
- osrednji zna sve jednostavne, većinu srednjih i nekoliko složenih činjenica
- početnik zna većinu jednostavnih činjenica i nekoliko srednjih činjenica
- novak zna najviše nekoliko jednostavnih činjenica
- svaki učenika može, ali ne mora znati neku ezoteričnu činjenicu, ali iskusniji učenici vjerojatno znaju više ezoteričnih činjenica

Uporaba dvostrukih stereotipova za modeliranje učenika omogućava sprečavanje ponovnog učenja učenika nečega što on već zna određivanjem količine sadržaja za prezentiranje koja treba biti objašnjena učeniku.

### **Bayesov model učenika**

Cilj učenja i poučavanja je usvajanje onog znanja koje je učeniku prezentirano. Pošto je učeniku dano na učenje ono što je stručnjak formalizirao i prikazao u grafu područnog znanja, logično je da od učenika očekujemo upravo poznavanje upravo tako formaliziranog i prikazanog znanja.

Jedna značajna poteškoća koja se javlja kod modeliranja učenika je neizvjesnost (eng. uncertainty). ITS mora izgraditi model učenika na temelju malih količina vrlo neizvjesnih informacija, jer se izvjesnost informacija može dobiti samo na temelju učenikovih aktivnosti u sustavu. Ukoliko do tih aktivnosti ne dođe, dijagnostika se mora provesti na temelju neizvjesnih informacija. Štoviše, zbog toga što ITS zasniva svoje odluke na modelu učenika, neizvjesnost koja se nalazi u modelu učenika doprinosi slabo prilagodljivim učenjem i poučavanjem.

Model učenika se gradi na temelju opažanja koja donosi ITS vezano za učenika. Model učenika se može promatrati kao sažimanje ovih opažanja: sirovi podaci se kombiniraju, neke od njih se i zanemaruje, a rezultat je sažetak u obliku skupa uvjerenja o učeniku. Proces sažimanja se definira kao skup pravila zaključivanja koja prevode opažanja u uvjerenja.

Razvijene su moćne opće teorije o donošenju odluka koje su razvijene posebno za situacije kod kojih se javlja neizvjesnost. Jedna od njih je *Bayesova teorija vjerojatnosti* ((Bayes, 1763), (Cheng & Greiner, 2001), (Mayo, 2001)), koja se bavi zaključivanjem pod neizvjesnošću. Bayesova teorija vjerojatnosti se bavi rješavanjem problema donošenja zaključaka kod neizvjesnosti. Posebna pažnja je usmjerena na predstavljanje neizvjesnih uvjerenja i njihovo mijenjanje kada dođe neki dokaz ili se promjeni neko drugo uvjerenje. Ona se bavi događajima i vjerojatnostima tih događaja.

Ako je  $A$  događaj, onda se vjerojatnost od  $A$  označava realnim brojem  $P(A)$ . Skup svih elementarnih događaja  $\Omega$  se naziva prostor događaja. Svaka vjerojatnosna funkcija  $P$  mora zadovoljavati tri aksioma:

1.  $(\forall A \in \Omega) P(A) \geq 0$  (nenegativnost)
2.  $P(\Omega) = 1$  (normiranost)
3. Ako su  $A_1, \dots, A_k$  međusobno isključivi, onda je  $P(A_1 \cup \dots \cup A_k) = P(A_1) + \dots + P(A_k)$  (aditivnost)

Iz 1. i 2. slijedi:  $\forall A \in \Omega, 1 \geq P(A) \geq 0$ , te ako je  $P(A) = 0$ , onda se  $A$  neće dogoditi, a ako je  $P(A) = 1$ ,  $A$  će se sigurno dogoditi.

Pravilo produkta definira vjerojatnost konjunkcije događaja:

$$P(A \cap B) = P(A|B)P(B) = P(B|A)P(A) \quad (2.1)$$

$P(A \cap B)$  se kraće označava s  $P(A, B)$ . To je vjerojatnost da će se dogoditi i  $A$  i  $B$  i naziva se *spojna vjerojatnost*. Spojna vjerojatnost predstavlja vjerojatnost dva ili više događaja, odnosno prikazuje vjerojatnosti svih mogućih ishoda. Koristeći spojnu vjerojatnost moguće je izračunati svaku „a priori“ i uvjetnu vjerojatnost, budući da su poznate sve moguće vjerojatnosti koje opisuju problem.

Za računanje složenih spojnih vjerojatnosti koristimo sljedeću formulu:

$$\begin{aligned}
 P(X_1, X_2, \dots, X_n) &= P(X_1|X_2, \dots, X_n)P(X_2, \dots, X_n) \\
 &= P(X_1|X_2, \dots, X_n)P(X_1|X_3, \dots, X_n)P(X_3, \dots, X_n) \\
 &= P(X_1|X_2, \dots, X_n)P(X_1|X_3, \dots, X_n) \dots P(X_{n-1}|X_n)P(X_n)
 \end{aligned}
 \tag{2.2}$$

Preformuliranje pravila produkta vodi do poznatog *Bayesovog teorema*:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.3}$$

Bayesov teorem računa vjerojatnost da se ukoliko je ostvaren događaj A, potvrdio skup početnih hipoteza. Bayesov teorem se često koristi za zaključivanje o neizvjesnoj hipotezi A ako je dan dokaz B, i u tom kontekstu se  $P(A|B)$  naziva *uvjetna vjerojatnost* od A (eng. *posterior probability*) tj. vjerojatnost od A nakon što se uzme u obzir utjecaj od B.  $P(A)$  se naziva „*a priori*“ vjerojatnost od A (eng. *prior probability*), a  $P(B|A)$  je vjerojatnost od B pretpostavljajući da je A istinit.

„*A priori*“ vjerojatnost opisuje vjerojatnost nastanka događaja u slučaju nedostatka bilo koje druge informacije. „*A priori*“ vjerojatnost se računa na klasičan matematički način tako da se broj povoljnih ishoda podijeli s ukupnim brojem ishoda. Uvjetna vjerojatnost opisuje vjerojatnost da je nešto istinito u slučaju da su poznate dodatne informacije.

Prije samog izračuna uvjetne i spojne vjerojatnosti, potrebno je utvrditi da li su događaji zavisni, nezavisni ili su možda međusobno isključivi. Dva događaja su *nezavisna*, ako bilo koji ishod jednog događaja ne utječe na vjerojatnost bilo kojeg drugog događaja. Spojna vjerojatnost nezavisnih događaja A i B jest produkt njihovih individualnih vjerojatnosti:  $P(A, B) = P(A) * P(B)$ . Uvjetna vjerojatnost nezavisnih događaja A i B je  $P(A|B) = P(A)$ , odnosno  $P(B|A) = P(B)$ .

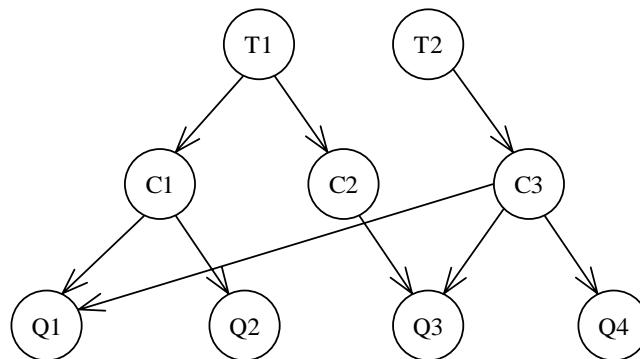
Ponekad je prikladno prikazati varijable i njihove zavisnosti kao usmjereni graf koji se naziva *Bayesova ili probabilistička mreža*. Bayesove mreže je prvi puta upotrijebio J. Pearl 1985. godine (Pearl, 1985). Bayesove mreže predstavljaju jedan od aktualnih pristupa rješavanju neizrazitog modeliranja ((Conati et al., 1997), (VanLehn, Niu, Siler, & Gertner, 1998), (Mayo, 2001), (Conati, A. Gertner, & VanLehn, 2002), (Gamboa & Fred, 2002)). Ova tehnika kombinira strogi formalizam vjerojatnosti s grafičkim prikazom i efikasnim mehanizmima zaključivanja. Bayesova mreža se može koristiti za modeliranje odnosa između akcija određenog učenika, njegovih unutrašnjih stanja i rezultata. Bayesove mreže su razvijene da bi učinile uvjetne nezavisnosti eksplicitnima.

U Bayesovoj mreži, varijabla koja uvjetuje drugu varijablu kod faktorizacije, postaje roditelj te varijable u mreži. Usmjerene petlje u mreži nemaju smisla jer bi one predstavljale faktorizaciju koja ne slijedi iz pravila produkta. Bayesova mreža je, stoga, usmjereni aciklički graf u kojem čvorovi predstavljaju slučajne varijable, a lukovi pokazuju uzročno-posljedične

veze, odnosno uvjetne zavisnosti između slučajnih varijabli. Kada postoje instance svih roditelja slučajne varijable A, kažemo da je ta slučajna varijabla uvjetno nezavisna od ostalih slučajnih varijabli koje nisu potomci od A ako su dani roditelji od A. Pravilo produkta ne određuje redoslijed kojim su varijable faktorizirane, pa ako se može pronaći redoslijed čvorova tako da svaka varijabla u faktorizaciji je povezana samo sa svojim roditeljima i svojim potomcima, tada će potomci „ispasti“ iz jednakosti i svaka varijabla će biti uvjetovana samo svojim roditeljima.

Varijable u Bayesovim mrežama imaju diskretne vrijednosti koje odgovaraju vjerojatnostima („a priori“ ili uvjetnim). Odnosi između čvorova roditelja i djeteta ukazuju da je za računanje distribucije vjerojatnosti djeteta potrebno poznavati određene vjerojatnosti roditelja. Svaki čvor u Bayesovoj mreži ima pripadajuću spojnu vjerojatnost, pa je stoga Bayesova mreža određena strukturom (grafom) i vrijednostima (brojke u tablicama distribucija vjerojatnosti kod čvorova).

Primjer Bayesove mreže na Slika 2.8 prikazuje jednostavan model učenika kod kojeg učenikova usvojenost nastavnih jedinica ( $T1$  i  $T2$ ) implicira usvojenost različitih nastavnih pojmova ( $C1$ ,  $C2$ ,  $C3$ ), što utječe na pitanja u testu ( $Q1$ ,  $Q2$ ,  $Q3$ ,  $Q4$ ) (Mayo, 2001). Pitanja mogu zahtijevati učenikovo usvajanje više od jednog koncepta. S ovom Bayesovom mrežom možemo izvoditi upite o zaključivanju  $P(Q1|T1 = usvojena)$ ; dijagnostičke upite  $P(T2|Q3 = pogrešno)$  ili njihovu kombinaciju  $P(C2|Q4 = točno, T1 = neusvojeno)$ .



Slika 2.8. Grafička struktura Bayesove mreže

Postoje tri klase Bayesovih modela učenika: ekspertni (eng. *expert centric*), učinkoviti (eng. *efficiency-centric*) i podatkovni (eng. *data-centric*) (Mayo, 2001).

Kod *ekspertnih modela* učenika stručnjak specificira direktno ili indirektno potpunu strukturu i uvjetne vjerojatnosti Bayesovog modela učenika na sličan način kako se rade ekspertni sustavi. Ovo je pristup sustava ANDES ((Gertner, Conati, & VanLehn, 1998), (Conati et al., 1997)), HYDRIVE (Mislevy & Gitomer, 1996), DT-Tutor (Murray & VanLehn, 2000). Jedan od nedostataka ovakvog pristupa je to što dobiveni model uključuju mnogo varijabli. Npr., promotrimo vrlo jednostavan Bayesov model učenika s dvije varijable, *Opažanje* (vidljivo) i *Stanje učenika* (nevidljivo), te uvjetnu vjerojatnost  $P(\text{Stanje učenika} | \text{Opažanje})$ . Kada

*Opažanje* postane poznato, uvjetna vjerojatnost *Stanja učenika* se mijenja. Uvjetna vjerojatnost  $P(\text{Stanje učenika} | \text{Opažanje})$  ne može se promijeniti. Kao rezultat, dva učenika s istim *Opažanjem* sustav će smatrati ekvivalentnima, iako opažanja sadržavaju samo nedavne interakcije.

*Učinkoviti modeli* rade na potpuno suprotan način: model je djelomično specificiran ili na neki način ograničen i područno znanje se „smješta“ u model. Ograničenja su odabrana tako da maksimiziraju učinkovitost (Reye, 1998). Općenito, ograničenja koja se uvode da bi se povećala učinkovitost mogu stvoriti netočne pretpostavke o područnom znanju.

*Podatkovni modeli* strukturu i uvjetne vjerojatnosti mreže „uče“ iz podataka sakupljenih stvarnim vrednovanjem znanja od strane živih učitelja. Kako je model induciran iz stvarnih podataka, njegovo predviđanje se može lako vrednovati testirajući mrežu na podacima koji nisu korišteni za njegovo „učenje“.

### ***Bloomova taksonomija znanja***

Najčešće polazište za definiranje obrazovnih ciljeva, ishoda učenja i zadataka zasniva se na Bloomovoj taksonomiji (Bloom, 1956) prema kojoj se razine postignuća dijele u tri područja: kognitivno (područje znanja i razumijevanja), psihomotorno (područje stavova i uvjerenja) i afektivno (područje vještina – umijeća).

Kognitivna domena uključuje učenje i razvoj intelektualnih vještina. To uključuje prisjećanje i prepoznavanje činjenica, proceduralne uzorke, i koncepte koji služe u razvoju intelektualnih sposobnosti i vještina. U okviru kognitivne kategorije Bloom razlikuje šest hijerarhijskih razina učenja:

1. Činjenično znanje – dosjećanje (eng. knowledge)

Odnosi se na temeljna znanja koja učenik mora steći da bi shvatio smisao. Na toj razini znanja treba se samo prisjetiti određene informacije, što ne mora nužno značiti i razumijevanje. Primjerice, učenik treba memorirati, definirati, opisati, označiti, nabrojati, prepoznati.

2. Razumijevanje (eng. comprehension)

Razumijevanje se definira kao sposobnost promišljanja o značenju usvojenih činjenica. Pokazuje se interpretiranjem naučenih činjenica, sažimanjem, objašnjavanjem ili predviđanjem učinaka ili posljedica. Primjerice, učenik treba znati interpretirati slike, karte, tablice i grafikone, verbalne zadatke prevesti u formule, na temelju činjenica predvidjeti posljedice, navesti primjer, interpretirati, parafrazirati.

3. Primjena (eng. application)

Primjena se odnosi na sposobnost uporabe naučenih pravila, zakona, metoda ili teorija u novim, konkretnim situacijama. Primjerice, učenik treba znati riješiti matematički problem, konstruirati grafikon ili krivulju, demonstrirati ispravnu uporabu neke metode ili postupka.

## 4. Analiza (eng. analysis)

Na analitičkoj razini znanja učenik mora biti sposoban naučene sadržaje razdvojiti na sastavne dijelove i razumjeti organizacijsku strukturu. Primjerice, učenik mora uspoređivati, suprotstavljati, prepoznati neizrečene pretpostavke, razlikovati činjenice od zaključaka, razlikovati uzrok od posljedice, odrediti relevantnost podataka, analizirati organizacijsku strukturu.

## 5. Sinteza (eng. synthesis)

Sintetizirati znači iz pojedinačnih dijelova stvoriti novu cjelinu. Primjeri obrazovnih ciljeva sintetičke razine znanja jesu: sposobnost kombinacije, postavljanja hipoteze, planiranja, reorganizacije, pisanja dobro organiziranog rad, održati dobro organizirani govor (predavanje), kreativno napisati priču (pjesmu, glazbu), predložiti plan pokusa.

## 6. Vrednovanje (eng. evaluation)

Vrednovanje znači sposobnost svrhovite prosudbe vrijednosti materijala. Prosudbe se moraju temeljiti na točno definiranim kriterijima. Primjeri obrazovnih ciljeva ove razine znanja jesu: prosuditi primjerenost zaključaka iz prikazanih podataka, prosuditi vrijednost nekog djela uporabom vanjskih standarda odličnosti, prosuditi logičnu postojanost pisanog materijala ili predavanja.

Detaljnija struktura Bloomove taksonomije je prikazana u Tablica 2.2:

Tablica 2.2. Struktura Bloomove taksonomije znanja

Kognitivna kategorija	Podkategorije	Podpodkategorije
dosjećanje	poznavanje posebnosti	poznavanje terminologije poznavanje činjenica
	poznavanje načina i sredstava za upravljanje posebnostima	poznavanje konvencija poznavanje trendova i nizova poznavanje klasifikacija i kategorija poznavanje kriterija poznavanje metodologije
	poznavanje općenitosti i apstrakcija	poznavanje principa i generalizacija poznavanje teorija i struktura
razumijevanje	translacija interpretacija ekstrapolacija	
primjena		
analiza	analiza elemenata analiza odnosa analiza organizacijskih principa	
sinteza	stvaranje jedinstvene komunikacije stvaranje plana ili ponuđenog skupa operacija derivacija skupa apstraktnih odnosa	
vrednovanje	vrednovanje prema unutrašnjim dokazima prosudba prema vanjskim kriterijima	

U revidiranoj Bloomovoj taksonomiji (Anderson, Krathwohl, & Bloom, 2004) promijenjen je redoslijed posljednje dvije kategorije i nazivlje kategorija je promijenjeno u tri slučaja. Revidirana Bloomova taksonomija je prikazana u Tablica 2.3:

Tablica 2.3. Struktura revidirane Bloomove taksonomije znanja

Kognitivna kategorija	Podkategorije
sjećanje	prepoznavanje prisjećanje
razumijevanje	interpretiranje davanje primjera klasificiranje sumiranje zaključivanje uspoređivanje objašnjavanje
primjena	izvršavanje implementiranje
analiza	razlikovanje organiziranje određivanje atributa
vrednovanje	provjeravanje kritiziranje
stvaranje	generiranje planiranje proizvodnja

Pošto ćemo mi zadnje tri kategorije promatrati kao jednu, nadalje u radu ćemo se pozivati samo na originalnu Bloomovu taksonomiju iz 1956. godine.

Dosta učenika prateći na satu nastavnika kako izvršava neku akciju razumije i misli kako akciju može i sam ponoviti, ali pri pokušaju ne uspijeva. Na primjer, dok nastavnik rješava i objašnjava programerski problem, učenik prati i razumije postupak, ali kada sličan ili čak isti zadatak pokuša sam riješiti ne uspijeva. Bloomova taksonomija daje objašnjenja za ovakve slučajeve. Učenikovo praćenje nastavnika na satu spada pod kategoriju razumijevanja, ali primjenu istoga ne uspijeva sam napraviti što spada pod kategoriju primjene.

Učenici napreduju različitim brzinama kroz različita područja učenja zbog motivacije ili sposobnosti. Stoga bi kod poučavanja trebalo početi s kategorijama nižih razina, te postupno prelaziti na više kategorije. U prilagodljivim sustavima e-učenja ovaj postupak je jako bitan. Jako važan postupak je utvrditi postojeću razinu znanja, ali i brzinu usvajanja novih sadržaja kako bi se stalno pratilo učenikov napredak.

Ako se u testovima javljaju pitanja koja spadaju samo u niže i/ili više Bloomove kategorije, učenici će znati odgovoriti na pitanja iz niže kategorije u velikom broju, dok će najčešće samo manji broj odgovoriti na pitanja iz više kategorije. Ako pitanja pokrivaju svih šest kategorija, dani odgovori dati će realniju sliku učenikovog znanja, što je ključno u dinamičkom utvrđivanju stereotipa učenika.



### 2.4.2.2 Formalizacija modula učenika u sustavu *AC-ware Tutor*

Navedena problematika modeliranja učenika i opisani pristupi modeliranju učenika doveli su do primjene dvostrukih stereotipova u našem pristupu modeliranju učenika. Koristimo stereotipove prema razinama stručnosti učenika i stereotipove prema Bloomovoj taksonomiji znanja.

#### ***Stereotipovi prema znanju***

U novom modelu dinamičkog i prilagodljivog računalom oblikovanog nastavnog sadržaja uvažavamo terminologiju uvriježenu u hrvatskoj pedagoškoj praksi te poznatu i priznatu Bloomovu taksonomiju znanja. Naime, kriteriji ocjenjivanja su usklađeni s *Bloomovom taksonomijom znanja* na način da s ocjenom dovoljan (2) opisujemo znanje na razini reprodukcije, dobar (3) razumijevanja, vrlo dobar (4) primjene, odličan (5) analize, sinteze i procjene, dok nedovoljan (1) opisuje nedostatak znanja.

Učenike možemo grupirati prema ocjenama koje su opisane na gornji način. Dakle, na raspolaganju imamo pet ocjena, što nam omogućava da učenike razvrstamo u pet kategorija. Svi učenici iz određene kategorije imaju zajedničko to što su iskazali znanje na istoj razini, tj. ta razina znanja je zajednička karakteristika svih učenika iz iste kategorije. To znači da sve vrlo dobre učenike karakterizira znanje na razini primjene, sve dovoljne učenike znanje na razini reprodukcije, itd.

U našem slučaju koristimo *stereotipove prema znanju*. Ako znamo kojem stereotipu učenik pripada, onda znamo i njegovu razinu znanja. Vrijedi i obratno, ako znamo koju razinu znanja učenik ima, onda znamo i kojem stereotipu prema znanju pripada. Na ovaj način smo „stereotipizirali“ učenike prema Bloomovoj taksonomiji znanja i govorimo o 5 stereotipova prema znanju: novak, početnik, osrednji, napredni i stručnjak. Navedeni broj stereotipova odgovara upravo našoj pedagoškoj praksi koja ocjenjuje učenike u rasponu od 1 do 5. Stereotipovi prema znanju su navedeni u sljedećoj tablici:

Tablica 2.4. Stereotipovi prema znanju

Ocjena	Razina znanja	Stereotip prema znanju
1	Nema znanja	Novak
2	Reprodukcija	Početak
3	Razumijevanje	Osrednji
4	Primjena	Napredni
5	Analiza, sinteza i vrednovanje	Stručnjak

Vidimo da na ovakav način sve učenike koji ne posjeduju nikakvo znanje (osim možda nešto znanja na razini reprodukcije) smatramo *novacima*, one koji iskazuju znanje na razini reprodukcije (i možda malo znanja na razini razumijevanja) smatramo *početnicima*, one koji pokazuju reprodukciju i razumijevanje (i možda malo znanja na razini primjene) smatramo *osrednjima*, one koji reproduciraju, razumiju i primjenjuju znanje (i možda analiziraju, sintetiziraju i procjenjuju) smatramo *naprednima*, dok one koji to znanje mogu reproducirati,

razumjeti, primijeniti, te analizirati, sintetizirati i procijeniti smatramo *stručnjacima* (Tablica 2.5).

Tablica 2.5. Odnos stereotipova učenika i razina znanja

Ocjena	Stereotip prema znanju	Reprodukcija	Razumijevanje	Primjena	Analiza, sinteza i vrednovanje
1	Novak	možda	nema	nema	nema
2	Početak	većina	možda	nema	nema
3	Osrednji	većina	većina	možda	nema
4	Napredni	većina	većina	većina	možda
5	Stručnjak	većina	većina	većina	većina

### **Bayesov model učenika i model prekrivanja**

Za učenika koji se nikada nije učio i poučavao na određenom područnom znanju, smatramo da koncepte iz grafa tog područnog znanja poznaje s vrlo malom vjerojatnošću, odnosno donosimo zaključke o njegovom znanju bez provjere istog. Smatramo, u tom slučaju, da učenik poznaje neki koncept iz tog područnog znanja s vjerojatnošću 0. Isto tako, ako smo provjerili znanje učenika o nekom područnom znanju i sa sigurnošću utvrdili da učenik poznaje sve koncepte iz nekog područnog znanja, onda možemo tvrditi da učenik poznaje neki koncept iz tog područnog znanja s vjerojatnošću 1.

Problem je kako odrediti vjerojatnosti poznavanja između 0 (nepoznavanje) i 1 (poznavanje). Upravo zbog toga definiramo Bayesovu mrežu predikcija poznavanja koncepata nekog područnog znanja. Dakle, koristimo ekspertni Bayesov model učenika u kombinaciji s modelom prekrivanja. Za svaki koncept područnog znanja definiramo vjerojatnost poznavanja tog koncepta za svakog učenika.

Kad se stvori model učenika, sve vjerojatnosti su 0. Nakon svakog pitanja iz testa kojim se ispituje poznavanje jedne ili više relacija, mijenjaju se uvjetne vjerojatnosti o usvojenosti koncepata koji sudjeluju u toj ili tim relacijama. Točni odgovori povećavaju vjerojatnosti, dok netočni odgovori smanjuju vjerojatnosti. Dakle, vjerojatnosti poznavanja određenih koncepata se prvi put promijene nakon ulaznog testa.

Model učitelja koristi uvjetne vjerojatnosti iz modela učenika da bi ustanovio koje koncepte učenik poznaje s visokom vjerojatnošću (npr. više od 0.8 – Bloom, 1976, prema (Gilbert, 2000)) tako da ga ne zamara s učenjem i poučavanjem koncepata koje već poznaje. Na ovaj način Bayesov model učenika služi kao „sito“ koje u proces učenja i poučavanja propušta samo one koncepte koje učenik ne poznaje s visokom vjerojatnošću.

Ako učenik poznaje sve relacije od nadkonceptata prema nekom konceptu i ako poznaje sve relacije od koncepta prema podkonceptima, onda smatramo da je vjerojatnost poznavanja koncepta 1.

### 2.4.3 Modul učitelja

Modeliranje učenika je beskorisna aktivnost ako nema primjenu. Cilj modeliranja učenika je odrediti ukupne učenikove kompetencije unutar područnog znanja. Model učenika se koristi i kod prilagođavanja ponašanja inteligentnih tutorskih sustava da bi optimizirali učenje i poučavanje učenika u odabranom područnom znanju. Prilagođava se računalom oblikovani nastavni sadržaj, povratna informacija i generiranje pomoći, te veličina elementa računalom oblikovanog nastavnog sadržaja.

#### 2.4.3.1 Teorijska podloga modula učitelja

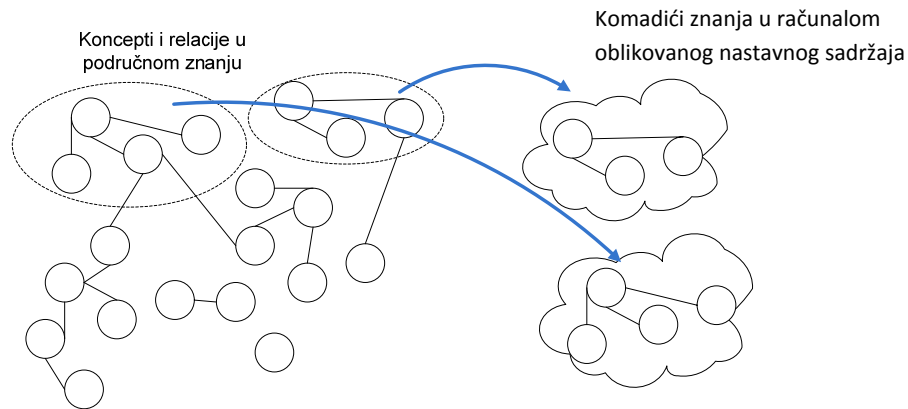
Proces nizanja sadržaja (eng. *course sequencing*) je dobro utemeljena tehnologija u ITS-ovima (Brusilovsky & Vassileva, 2003). Glavna prednost dinamičkog nizanja je sposobnost prilagođavanja sadržaja pojedincu. Mehanizam nizanja se temelji na modelu učenika koji za svaki element područnog znanja sadrži aktualnu razinu učenikovog poznavanja istog. Postoje tri različita pristupa nizanju sadržaja: statički računalom oblikovani nastavni sadržaj, prilagodljivo generiranje računalom oblikovanog nastavnog sadržaja (eng. *adaptive courseware generation*), dinamičko generiranje računalom oblikovanog nastavnog sadržaja (eng. *dynamic courseware generation*).

Model računalom oblikovanog nastavnog sadržaja u većini ITS-ova je statički niz (ili stablo) koje slijede statički testovi. Svi elementi su unaprijed stvoreni od strane učitelja. Ovakav pristup ne podržava prilagođavanje učeniku. Ideja prilagodljivog generiranje računalom oblikovanog nastavnog sadržaja je u generiranju sadržaja prilagođenog potrebama učenika i to prije nego se učenici susretnu s njim. Umjesto inkrementalnog nizanja, cijeli računalom oblikovani nastavni sadržaj treba generirati odjednom, čime se umanjuje prilagodljivost. Model računalom oblikovanog nastavnog sadržaja generiranog na ovaj način je statičan (Brusilovsky & Vassileva, 2003).

Dinamičko generiranje računalom oblikovanog nastavnog sadržaja promatra učenikov napredak i prilagođavaju se generirani sadržaji. Ako učenikovi rezultati ne odgovaraju očekivanima, sadržaji se dinamički regeneriraju. Između regeneriranja, model računalom oblikovanog nastavnog sadržaja je statičan. DCG (Vassileva, 1992) je najbliže dinamičkom generiranju računalom oblikovanog nastavnog sadržaja koji uzima u obzir učenikovo aktualno znanje, ciljeve, vremenski okvir, te se prilagođava njegovom i tempu napredovanja.

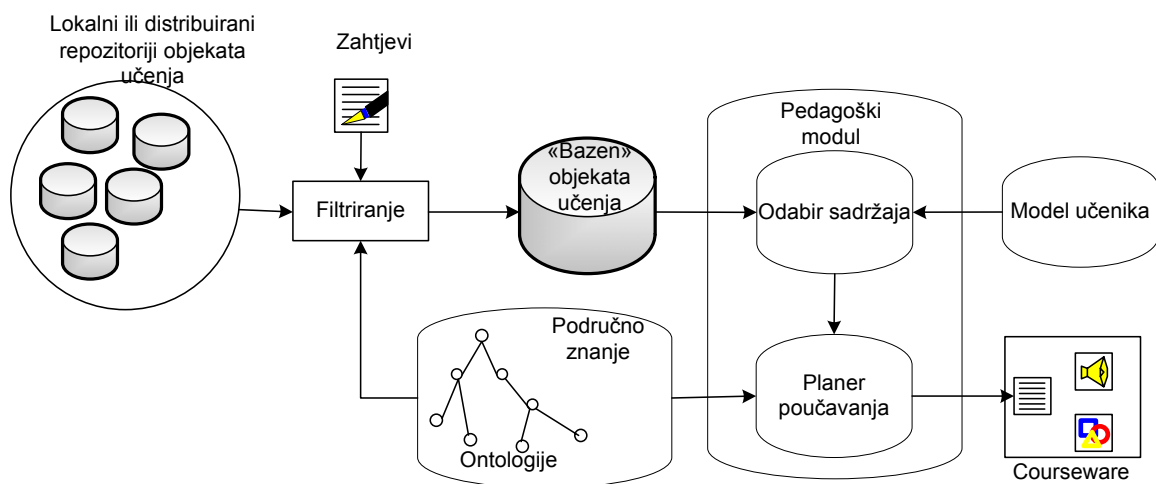
U svim sustavima koji imaju nizanje sadržaja, postoje elementi područnog znanja (eng. *domain knowledge elements – DKE*) koji se u različitim sustavima različito zovu: koncepti, fragmenti znanja, teme, objekti učenja, objekti znanja (Brusilovsky & Vassileva, 2003). Uporaba malenih, ponovno upotrebljivih komadića (eng. *chunks*) se koristi i kod oblikovanja nastave (Reigeluth i Nelson, 1997, prema (Wiley, 2000)). Naime, kada učitelj prvi put pristupa sadržaju, često ga dijeli na sastavne dijelove koje presloži na način koji će podupirati

njegove obrazovne ciljeve. Ovi „komadići“ znanja mogu reprezentirati veći ili manji dio područnog znanja, ovisno o samom područnom znanju i onome tko ih definira (Slika 2.9).



Slika 2.9. Odnos područnog znanja i računalom oblikovanog nastavnog sadržaja

Opća arhitektura sustava koji podržavaju nizanje sadržaja temeljeno na „komadićima“ znanja je prikazana na Slika 2.10.



Slika 2.10. Opća arhitektura sustava koji podržavaju nizanje sadržaja (mod. prema (Karampiperis & Sampson, 2004))

Vidimo da sustavi temeljeni na ovoj arhitekturi moraju filtrirati područno znanje, stvoriti „komadiće“ znanja od kojih će odabrati one koji čine računalom oblikovani nastavni sadržaj, te prilagoditi taj sadržaj učeniku.

### 2.4.3.2 Formalizacija modula učitelja u sustavu *AC-ware Tutor*

U novom modelu dinamičkog i prilagodljivog računalom oblikovanog nastavnog sadržaja uvažavamo terminologiju uvriježenu u hrvatskoj pedagoškoj praksi. Naime, elementi računalom oblikovanog nastavnog sadržaja su *nastavne cjeline* (NC), *nastavne teme* (NT) i *nastavne jedinice* (NJ), kao i *testovi* (T). Nastavne cjeline, teme i jedinice su elementi za učenje, a testovi su elementi za provjeravanje znanja.

Navodimo sljedeće definicije:

**Definicija 2.9** *Računalom oblikovani nastavni sadržaj NS* je niz elemenata računalom oblikovanog nastavnog sadržaja.

**Definicija 2.10** *Element računalom oblikovanog nastavnog sadržaja za učenje* je podgraf  $(V'_{NE}, A'_{NE})$  grafa područnog znanja. Za svaki element za učenje potrebno je samo definirati koncept  $K_x$  koji će biti korijen njegovog stabla. Stoga uvodimo sljedeću oznaku  $NE_{K_x}$  za nastavni element čiji je korijen koncept  $K_x$ , tj. koji počinje u  $K_x$ , odnosno  $K_x$  je početak od  $NE_{K_x}$ .

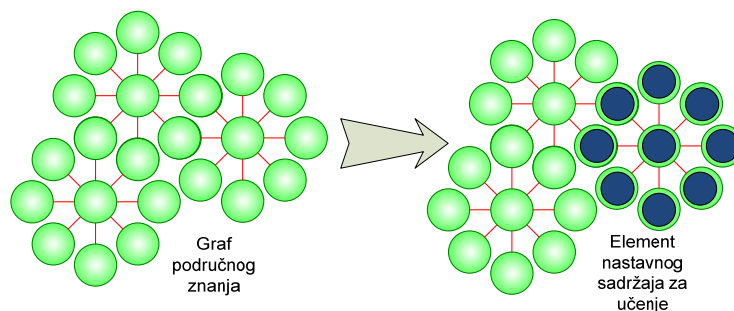
**Definicija 2.11** *Rang elementa računalom oblikovanog nastavnog sadržaja za učenje  $R_{NE_{K_x}}$*  je uređena trojka  $(i, y, t)$  gdje je  $i$  oznaka cjeline  $C_i$  čiji je  $NE_{K_x}$  podgraf,  $y$  duljina puta od korijena cjeline  $C_i$  do početka elementa računalom oblikovanog nastavnog sadržaja, a  $t$  oznaka razine elementa računalom oblikovanog nastavnog sadržaja (1-NC, 2-NT, 3-NJ).

Element računalom oblikovanog nastavnog sadržaja je jedinstveno određen svojim korijenom i rangom. Element računalom oblikovanog nastavnog sadržaja za učenje predstavlja objekt znanja (eng. knowledge object) koji odgovara definiciji Merrill-a (Merrill, 1998): *objekt znanja pruža okvir za organiziranje područnog znanja na takav način da se može koristiti u različitim kontekstima i da se može prikazati na različite načine*. U novije vrijeme se koristi naziv objekt učenja (eng. learning object – LO) za sve digitalne izvore koje se može više puta upotrijebiti u svrhu računalne potpore učenju (LTSC, 2011).

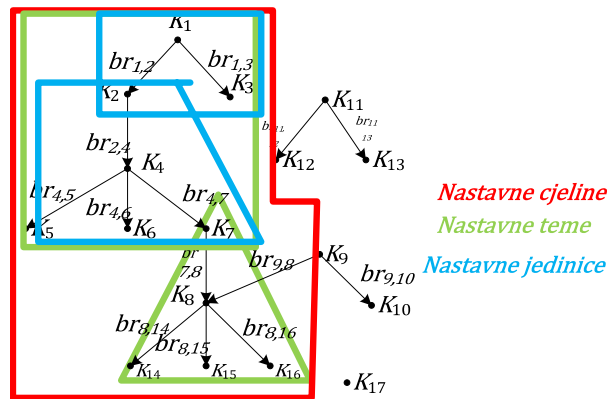
Razina elementa računalom oblikovanog nastavnog sadržaja za učenje određuje koliki podskup područnog znanja čini određeni element računalom oblikovanog nastavnog sadržaja za učenje (Slika 2.11, Slika 2.12).

Minimalni uvjeti za generiranje elemenata računalom oblikovanog nastavnog sadržaja za učenje su sljedeći:

- nastavna cjelina odgovara cjelini područnog znanja → postojanje barem jednog puta duljine 4 i više od početka elementa
- nastavna tema → postojanje barem jednog puta duljine 2 od početka elementa i barem dva podpodkoncepta ili postojanje barem jednog puta duljine 3 od početka elementa
- nastavna jedinica → postojanje barem jednog puta duljine 1 od početka elementa i barem dva podkoncepta ili postojanje barem jednog puta duljine 2 od početka elementa



Slika 2.11. Element računalom oblikovanog nastavnog sadržaja za učenje



Slika 2.12. Primjer elemenata računalom oblikovanog nastavnog sadržaja za učenje različitih razina

**Definicija 2.12** *Element računalom oblikovanog nastavnog sadržaja za provjeravanje znanja* je niz pitanja kojima se ispituje učenikovo poznavanje povezanosti koncepata. Jedinstveno je određen elementima za učenje koji mu prethode u računalom oblikovanom nastavnom sadržaju NS i trenutnim stereotipom učenika.

Pobliže ćemo opisati elemente za provjeravanje znanja jer nam oni služe za određivanje stereotipa učenika.

### **Elementi za provjeravanje znanja**

Osim prilagođavanja prilikom dinamičkog generiranja računalom oblikovanog nastavnog sadržaja, potrebno je prilagoditi i način provjeravanja znanja učenika. Prilagodljiva provjera znanja je u automatskom oponašanju mudrog ispitivača, što znači da ako ispitivač postavi pitanje za koje se ispostavi da je preteško, onda sljedeće pitanje treba biti jednostavnije (Wainer i Mislevy, 1990, prema (Conejo et al., 2004)). *Računalni prilagodljivi testovi* (eng. Computer Adaptive Test – CAT) su testovi administrirani računalom kod kojih se prezentiranje svakog pitanja i odluka o završavanju provjeravanja znanja prilagođava na temelju odgovora na postavljena pitanja. CAT testovi omogućavaju postavljanje nekoliko dobro odabranih težih pitanja učenicima s većim znanjem (da bi im omogućili da pokažu što su naučili), a lakših pitanja učenicima s manje znanja (da bi brzo utvrdili da nisu dovoljno naučili) (Conejo et al., 2004).

Najveći problem koji postoji kod CAT testova je zahtjev za postojanjem baze podataka s velikim brojem pitanja različite težine koja definira učitelj. Mnogo bolje rješenje je *automatizirano dinamičko generiranje testova* na temelju elemenata područnog znanja kod kojeg pitanja definira sam prilagodljivi sustav e-učenja. Takvi testovi se također prilagođavaju učeniku prema znanju pokazanom tijekom prethodnog provjeravanja znanja. Upravo ovakav načina provjeravanja znanja je korišten u modelu *AC-ware Tutor*.

Elementi za provjeravanje znanja sadrže pitanja generirana samo na jednom podskupu područnog znanja. Pošto se znanje provjerava iterativno u procesu učenja i poučavanja, za

generiranje *elemenata za provjeravanje znanja* je potrebno znati samo nad kojim skupom koncepata i relacija je potrebno izvršiti provjeru znanja, te trenutni stereotip učenika. Dakle, ulaz u generiranje elementa za provjeravanje znanja je podskup područnog znanja  $PZ'$ .  $GPZ' = (V', A')$  je graf podskupa područnog znanja nastao na temelju  $PZ'$ . Ako se radi o ulaznom testu, onda je  $PZ' = RepPZ$ ,  $GPZ' = RepGPZ$ .

Elementi za provjeravanje znanja se sastoje od određenog broja pitanja koja se stvaraju prema predlošcima podijeljenima u 4 težinske kategorije koje su u uskoj vezi sa stereotipovima. Svaka težinska kategorija ispituje znanje određene razine (prema Bloomovoj taksonomiji): *prva težinska kategorija* sadrži predloške koji ispituju znanje na razini reprodukcije, *druga* na razini razumijevanja, *treća* na razini primjene, a *četvrta* na razini analize, sinteze i procjene. Predloški pitanja omogućavaju stvaranje zadataka objektivnog tipa (Gronlund, 1985): dopunjavanje, jednostavno dosjećanje, jednostruki izbor, povezivanje, sređivanje, višestruki izbor. Prednosti zadataka objektivnog tipa su: da mjere znanje, objektivni su i ekonomični jer za malo vremena omogućavaju ispitivanje velikog broja učenika, te se brzo ispravljaju. Nedostaci zadataka objektivnog tipa su: ispituju uglavnom poznavanje činjenica, skupi su i dugo se izrađuju. U našem pristupu, zadnji nedostatak klasičnih zadataka objektivnog tipa i ne postoji, već su besplatni pošto naš pristup omogućava automatsko i dinamičko generiranje pitanja za svakog učenika posebno, pa se samim time i ne ponavljaju.

Različitim vrstama pitanja provjeravamo različite razine znanja učenika ((eLAN, 2004), (HEC-PIEAS, 2009), (Notar et al, 2004), (Roberson-Scott, 2005)). Veza razine znanja, stereotipova i vrste pitanja s predlošcima je prikazana u Tablica 2.6.

Tablica 2.6. Razine znanja, stereotipovi i vrste pitanja s predlošcima

Težinska kategorija pitanja	Razina znanja	Stereotip prema znanju	Vrsta pitanja	Predloški pitanja
Nema pitanja	Nema znanja	Novak	-	-
1	Reprodukcija	Početak	Jednostruki izbor Višestruki izbor Jednostavno dosjećanje Povezivanje	Da li su $K_x$ i $K_y$ neposredno povezani? Da li su $K_x$ i $K_y$ neposredno povezani s $r$ ? Da li je $K_x$ podkoncept od $K_y$ ? Da li je $K_x$ nadkoncept od $K_y$ ? Što neposredno povezuje $K_x$ i $K_y$ ?
2	Razumijevanje	Osrednji	Jednostruki izbor Dopunjavanje Višestruki izbor Jednostavno dosjećanje Povezivanje	Da li su $K_x$ i $K_y$ posredno povezani? Što je $K_x$ ? Koji podkoncept je neposredno povezan vezom $r$ s $K_x$ ? Koji nadkoncept je neposredno povezan vezom $r$ s $K_x$ ?
3	Primjena	Napredni	Višestruki izbor Jednostavno dosjećanje Povezivanje	$K_x$ je _____ od $K_y$ S koliko koncepata je povezan $K_x$ ? Koliko podkoncepata ima $K_x$ ? Koliko nadkoncepata ima $K_x$ ? Poveži zadane koncepte
4	Analiza, sinteza i vrednovanje	Stručnjak	Višestruki izbor Jednostavno dosjećanje Povezivanje Sređivanje	$K_x$ i $K_y$ su neposredno povezani s ____ Poredaj: Da li $K_{slot}$ za $K_x$ ima $K_{filler}$ ? Koja je $K_{slot}$ od $K_x$ ? Čija je $K_{slot}$ od $K_{filler}$ ?

Svako pitanje omogućuje ocjenjivanje poznavanja povezanosti između dva koncepta, kao i poznavanje samih konceptata. Poznavanje se boduje na ljestvici od 0 do 4. Bez obzira na pitanje, ako je odgovor „Ne znam“, onda je ostvareni broj bodova na tom pitanju jednak 0. Netočan odgovor donosi 0 bodova, dok točan odgovor donosi onoliko bodova kolika je težina kategorije kojoj pripada.

**Definicija 2.13** Kažemo da je *podskup područnog znanja provjerljiv* ako se  $\forall K_x, K_y \in V'$  može generirati pitanje prema barem jednom predlošku iz svake težinske kategorije.

Kažemo da je *element računalom oblikovanog nastavnog sadržaja za provjeravanje znanja izvediv* samo ako se generira nad provjerljivim podskupom područnog znanja, tj.  $\forall K_x, K_y \in V'$  postoji barem jedno pitanje iz svake težinske kategorije koje ispituje njihovu povezanost.

Svaki test se dinamički generira u trenutku izvršavanja i to jedino ako se utvrdi da je izvediv. U protivnom se iz podskupa područnog znanja moraju izbaciti uređene trojke čija se povezanost ne može provjeriti.

Modul učenika omogućava realizaciju Bayesovog modela učenika. Naime, definiramo dvije težinske funkcije na grafu područnog znanja kojima se vrijednosti određuju nakon svakog provjeravanja znanja:

**Definicija 2.14** Funkcija  $X_A: A' \rightarrow \{-1, 0, 1, 2, 3, 4\}$  definirana sa:  $\forall K_x K_y \in A'$ ,  $X_A(K_x K_y)$  = broj bodova koji je dobiven odgovaranjem na pitanje iz testa koje se odnosi na taj brid (koji odgovara uređenoj trojci područnog znanja),  $\forall K_x K_y \in A_{GPZ} \setminus A'$ ,  $X_A(K_x K_y) = -1$  je *težinska funkcija na skupu bridova grafa područnog znanja*.

$$\forall K_x \in V_{GPZ}, \forall K_x K_{y_i} \in A_{GPZ}, \forall K_{y_i} K_x \in A_{GPZ}, \text{ vrijedi}$$

$$\sum_{i=1}^k X_A(K_x K_{y_i}) = 0, \text{ ako } \forall K_x K_{y_i} \in A_{GPZ}, X_A(K_x K_{y_i}) = -1,$$

$$\sum_{i=1}^k X_A(K_x K_{y_i}) = 0, \text{ ako } \forall K_x K_{y_i} \in A_{GPZ}, X_A(K_x K_{y_i}) = -1$$

Funkcija  $X_A$  omogućava pridjeljivanje težina onim bridovima koji povezuju koncepte čija je povezanost provjeravana određenim pitanjem u testu. Svaki brid iz  $A'$  će imati pridijeljenu težinu između 0 i 4 (što odgovara mogućim bodovima), dok će svi ostali bridovi u grafu područnog znanja imati pridijeljenu vrijednost -1, što će označavati da poznavanje povezanosti ta dva koncepta nismo provjeravali. Dakle, graf područnog znanja uz pomoć težinske funkcije  $X_A$  postaje bridno-težinski graf kojem se vrijednosti težinske funkcije mijenjaju nakon svakog testiranja znanja.

Na temelju vrijednosti bridno-težinske funkcije  $X_A$  određuju se vrijednosti vršno-težinske funkcije  $X_V$ :



**Definicija 2.15** Funkcija  $X_V: V_{GPZ} \rightarrow [0,1]$  definirana sa:

$$\forall K_x \in V_{GPZ},$$

$$\forall K_x K_{y_i} \in A_{GPZ}, X_A(K_x K_{y_i}) \neq -1,$$

$$\forall K_{y_i} K_x \in A_{GPZ}, X_A(K_{y_i} K_x) \neq -1,$$

$$X_V(K_x) = \frac{\left( \frac{\sum_{i=1}^k X_A(K_x K_{y_i})}{4 * pK_x} + \frac{\sum_{i=1}^l X_A(K_{y_i} K_x)}{4 * nK_x} \right)}{2} =$$

$$= \left( \frac{\sum_{i=1}^k X_A(K_x K_{y_i})}{pK_x} + \frac{\sum_{i=1}^l X_A(K_{y_i} K_x)}{nK_x} \right) * \frac{1}{8}, pK_x \neq 0, nK_x \neq 0$$

$$X_V(K_x) = \frac{\sum_{i=1}^k X_A(K_x K_{y_i})}{4 * pK_x}, pK_x \neq 0, nK_x = 0$$

$$X_V(K_x) = \frac{\sum_{i=1}^l X_A(K_{y_i} K_x)}{4 * nK_x}, pK_x = 0, nK_x \neq 0$$

$$X_V(K_x) = 0, pK_x = 0, nK_x = 0$$

je **težinska funkcija na skupu vrhova grafa područnog znanja**

Vrijednost  $X_V(K_x)$  predstavlja normiranu sumu vrijednosti funkcije  $X_A$  bridova prema nadkonceptima koncepta  $K_x$  i bridova prema podkonceptima koncepta  $K_x$ . Smatramo da učenik potpuno poznaje koncept  $K_x$  akko je  $X_V(K_x) = 1$ , što je istina samo ako je poznavanje svih relacija prema nadkonceptima i podkonceptima na razini 4. Tada je vjerojatnost poznavanja koncepta najviša, odnosno jednaka 1.

Rezultati provjeravanja znanja, odnosno vrijednosti težinskih funkcija omogućavaju definiranje sljedećih podskupova konceptata područnog znanja:

**Definicija 2.16** Skup  $Z_{Razina_0} = \{K_x \in E_{KCP} | X_V(K_x) < 0,2\}$  je **skup konceptata područnog znanja koje učenik ne poznaje**.

Skup  $Z_{Razina_1} = \{K_x \in E_{KCP} | 0,2 \leq X_V(K_x) < 0,4\}$  je **skup konceptata područnog znanja koje učenik poznaje na razini 1** - znanje na razini reprodukcije činjenica.

Skup  $Z_{Razina_2} = \{K_x \in E_{KCP} | 0,4 \leq X_V(K_x) < 0,6\}$  je **skup konceptata područnog znanja koje učenik poznaje na razini 2** - znanje na razini razumijevanja.

Skup  $Z_{Razina_3} = \{K_x \in E_{KCP} | 0,6 \leq X_V(K_x) < 0,8\}$  je **skup konceptata područnog znanja koje učenik poznaje na razini 3** - znanje na razini primjene.

Skup  $Z_{Razina_4} = \{K_x \in E_{KCP} | 0,8 \leq X_V(K_x) \leq 1\}$  je **skup konceptata područnog znanja koje učenik poznaje na razini 4** - znanje na razini analize, sinteze i procjene.

Upravo nam je ova definicija omogućila određivanje stereotipova prema znanju u novom modelu prilagodljivog stjecanja znanja. Dakle, na najnižoj razini formalizacije definiramo:

**Definicija 2.17** Definiramo stereotipove prema znanju:

Za nekog učenika kažemo da je **novak** (eng. novice) ako nakon riješenog ulaznog testa najviše elemenata ima u skupu  $Z_{Razina_0}$ . On ima **razinu znanja 0**.

Za nekog učenika kažemo da je **početnik** (eng. beginner) ako nakon riješenog ulaznog testa najviše elemenata ima u skupu  $Z_{Razina_1}$ . On ima **razinu znanja 1**.

Za nekog učenika kažemo da je **osrednji** (eng. intermediate) ako nakon riješenog ulaznog testa najviše elemenata ima u skupu  $Z_{Razina_2}$ . On ima **razinu znanja 2**.

Za nekog učenika kažemo da je **napredni** (eng. advanced) ako nakon riješenog ulaznog testa najviše elemenata ima u skupu  $Z_{Razina_3}$ . On ima **razinu znanja 3**.

Za nekog učenika kažemo da je **stručnjak** (eng. expert) ako nakon riješenog ulaznog testa najviše elemenata ima u skupu  $Z_{Razina_4}$ . On ima **razinu znanja 4**.

U sljedećoj tablici je sažeto prikazan odnos težinskih kategorija pitanja, broja bodova i stereotipova učenika:

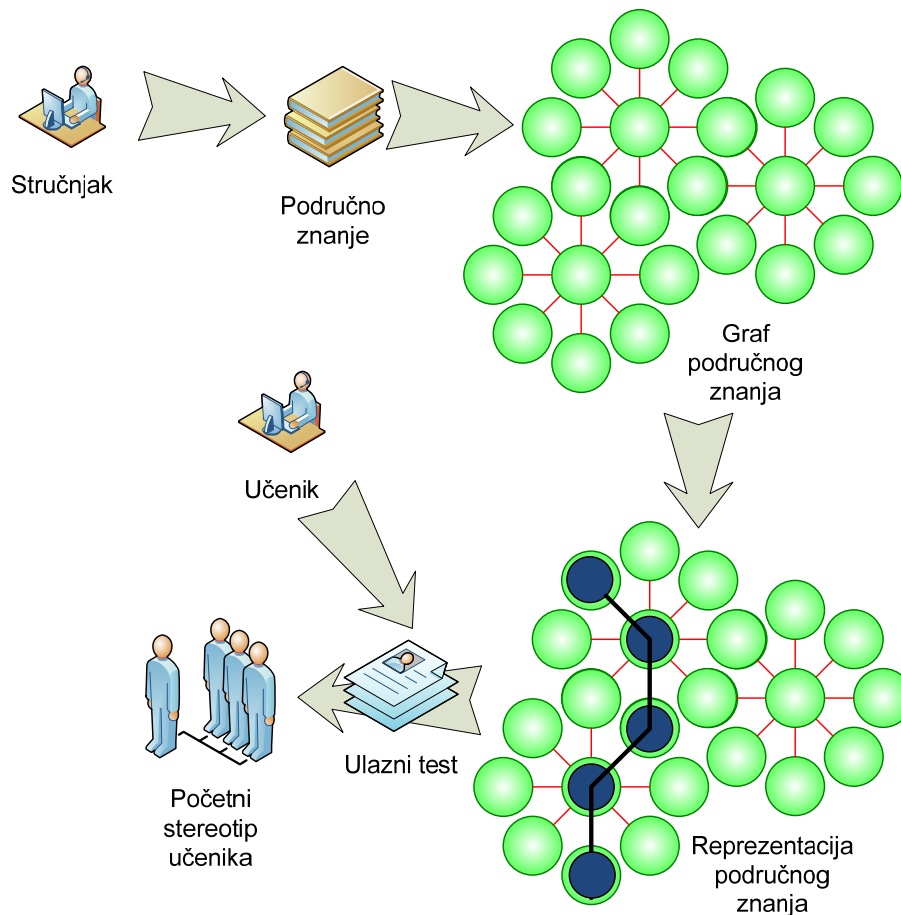
**Tablica 2.7. Težinske kategorije pitanja i stereotipovi**

Težinska kategorija pitanja	Razina znanja	Stereotip prema znanju	Razina znanja	Bodovi
Nema pitanja	Nema znanja	Novak	<i>razina=0</i>	-
1	Reprodukcija	Početnik	<i>razina=1</i>	0, 1
2	Razumijevanje	Osrednji	<i>razina=2</i>	0, 2
3	Primjena	Napredni	<i>razina=3</i>	0, 3
4	Analiza, sinteza i vrednovanje	Stručnjak	<i>razina=4</i>	0, 4

Elemente za provjeravanje znanja koristimo da bi ustanovili kojem stereotipu učenik pripada. Problem je kako utvrditi *početni stereotip* prema kojem će se učenika u nastavku učiti i poučavati i na kojem će se temeljiti prvi test u računalom oblikovanom nastavnom sadržaju. Ovaj problem se rješava generiranjem ulaznog testa nad *reprezentacijom područnog znanja* (Definicija 2.8).

Provjeravanje znanja u ulaznom testu koji se generira nad reprezentacijom područnog znanja, započinje s pitanjima iz *treće* kategorije. Nad konceptima koji su obuhvaćeni s pitanjima na koja učenik nije točno odgovorio, generiraju se pitanja iz *druge* težinske kategorije, a nad konceptima koji su obuhvaćeni s pitanjima na koja je učenik točno odgovorio, generiraju se pitanja iz *četvrte* težinske kategorije. Nad konceptima koji su obuhvaćeni s pitanjima iz druge težinske kategorije na koja učenik nije točno odgovorio, generiraju se pitanja iz *prve* težinske kategorije.

Određivanje stereotipa nakon ulaznog testa (i nakon svih ostalih) se radi na način da se pronađe težinska kategorija pitanja koja sadrži najveći broj točno odgovorenih pitanja. Ako je najbrojnija prva kategorija (učenik ima najviše točno odgovorenih pitanja koja ispituju znanje na razini reprodukcije), onda učenik pripada stereotipu *početnik*, itd. Ako učenik ima najviše netočno odgovorenih pitanja ili pitanja s odgovorom „Ne znam“, onda on pripada stereotipu *novak*.

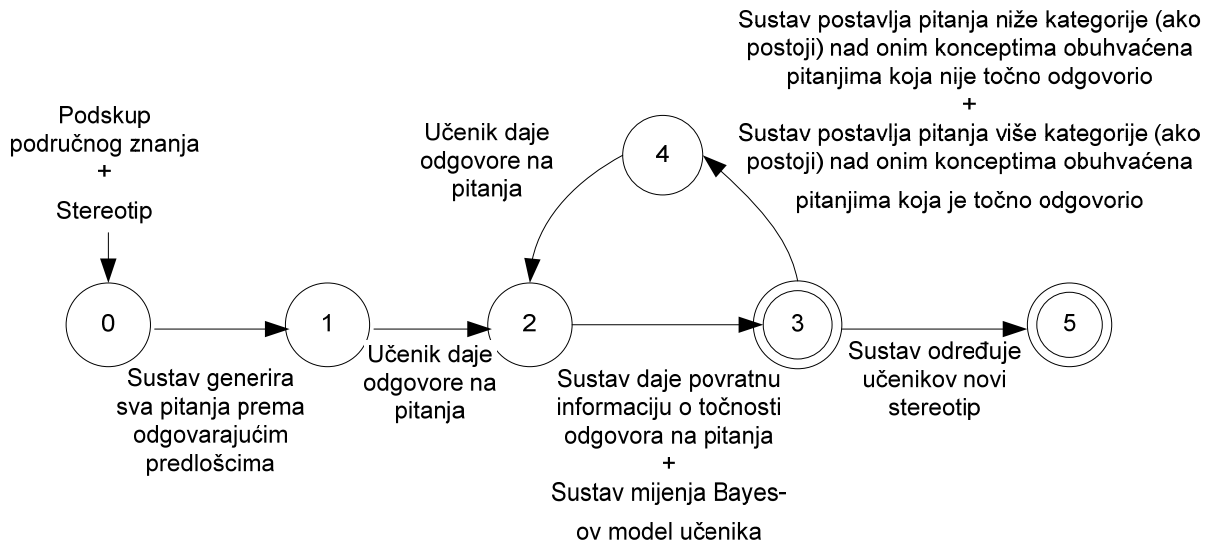


Slika 2.13. Određivanje početnog stereotipa učenika

Prije generiranja testa potrebno je utvrditi kojem stereotipu učenik trenutno pripada. Zatim se  $\forall K_x, K_y \in V'$ , koje učenik poznaje na razini jednakoj ili manjoj od one kojoj on pripada, generira po jedno pitanje prema predlošcima iz težinske kategorije koja je za jedan veća od one kojoj učenik pripada, tj. *razina* + 1 (osim ako je učenik *stručnjak*, tj. *razina* = 4, pa se tada kreće od četvrte težinske kategorije). Ovakvim pristupom odmah možemo utvrditi da li je došlo da napretka u znanju učenika (ako riješi pitanja iz veće težinske kategorije nego što je njegovo prethodno iskazano znanje). Također ovakav pristup povećava pozitivnu korelaciju između motivacije i rezultata (Pintrich, 1999, prema (Ullrich, 2008)).

Nad konceptima koji su obuhvaćeni s pitanjima na koja učenik nije točno odgovorio, generiraju se pitanja iz manje težinske kategorije (osim ako su nad njima već bila generirana pitanja iz prve težinske kategorije, pa ne postoji manja težinska kategorija, tada se za te koncepte više ne generiraju pitanja), a nad konceptima koji su obuhvaćeni s pitanjima na koja je učenik točno odgovorio, generiraju se pitanja iz veće težinske kategorije (osim ako su nad njima već bila generirana pitanja iz četvrte težinske kategorije, pa ne postoji veća težinska kategorija, tada se za te koncepte više ne generiraju pitanja).

Realizaciju provjeravanja znanja u modulu učitelja novog modela za prilagodljivo stjecanje znanja opisujemo strojem s konačnim brojem stanja (Slika 2.14):



Slika 2.14. Realizacija provjeravanja znanja

Vidimo da se stereotip učenika može promijeniti samo nakon provedenog testiranja znanja. Nastavak učenja i poučavanja se prilagođava dobivenom stereotipu u načinu odabira, nizanja i prezentiranja računalom oblikovanog nastavnog sadržaja.

### ***Dinamički računalom oblikovani nastavni sadržaj – odabir, nizanje, prezentiranje***

Nakon što smo definirali stereotipove i način na koji ćemo odrediti kojem stereotipu učenik pripada, potrebno je definirati način na koji će se određenom stereotipu odabrati, nizati i prezentirati elementi računalom oblikovanog nastavnog sadržaja.

*Odabir elemenata* računalom oblikovanog nastavnog sadržaja se vrši prema razini elemenata. Razina elementa nam određuje količinu znanja koju će učenik učiti u jednom ciklusu učenja i poučavanja. S obzirom da smo definirali pet stereotipova učenika (novak, početnik, osrednji, napredni i stručnjak), a imamo samo tri razine elemenata računalom oblikovanog nastavnog sadržaja (jedinica, tema, cjelina), neki od stereotipova će se učiti i poučavati na istoj razini (Tablica 2.8):

Tablica 2.8. Stereotipovi i najveća razina elemenata računalom oblikovanog nastavnog sadržaja

Stereotip prema znanju	Razina elementa računalom oblikovanog nastavnog sadržaja
Novak	Nastavna jedinica
Početnik	Nastavna jedinica
Osrednji	Nastavna tema
Napredni	Nastavna tema
Stručnjak	Nastavna cjelina

Nakon što se za određeni stereotip odaberu svi elementi za učenje određene razine, *nizanje elemenata* predstavlja dodavanje sortiranih elemenata u računalom oblikovani nastavni

sadržaj i dodavanje elemenata za provjeravanje znanja prema pravilima koja ovise o stereotipu učenika.

Elementi računalom oblikovanog nastavnog sadržaja se *sortiraju* prema njihovom rangu: prvo prema oznaci cjeline, a onda rastući prema udaljenosti početaka elemenata od korijena cjeline, zatim prema razini elementa, te na kraju abecedno prema nazivu korijena.

*Elementi za provjeravanje znanja se dodaju* u ovisnosti o stereotipu kojem pripada. Za sve stereotipove je zajedničko da zadnji element računalom oblikovanog nastavnog sadržaja mora biti test. Uvažavajući razinu elementa koja je pridijeljena kojem stereotipu, predlažemo sljedeći raspored testiranja znanja učenika (Tablica 2.9).

Nakon što smo definirali način na koji ćemo odrediti kojem stereotipu učenik pripada, način na koji će se određenom stereotipu odabrati i nizati elementi računalom oblikovanog nastavnog sadržaja, potrebno je definirati i način prezentiranja elemenata računalom oblikovanog nastavnog sadržaja koje je prilagođeno stereotipovima. *Prezentiranje elemenata* računalom oblikovanog nastavnog sadržaja se vrši s aspekta Bloomove taksonomije. Svakom stereotipu se prezentira odgovarajuća razina znanja (Tablica 2.10).

Tablica 2.9. Pravila za dodavanje elemenata za provjeravanje znanja

Stereotip prema znanju	Pravila za dodavanje elemenata za provjeravanje znanja
Novak	<ul style="list-style-type: none"> <li>- kod promjene oznake cjelina,</li> <li>- kod promjena udaljenosti od korijena (osim ako bi testu prethodila samo jedna nastavna jedinica ili ako bi iza testa ostala samo jedna nastavna jedinica),</li> <li>- nakon tri nastavne jedinice u nizu koje nisu test (osim ako bi testu prethodila samo jedna nastavna jedinica ili ako bi iza testa ostala samo jedna nastavna jedinica)</li> <li>- zadnji element računalom oblikovanog nastavnog sadržaja mora biti test</li> </ul>
Početak	<ul style="list-style-type: none"> <li>- kod promjene oznake cjelina (jedino ako se ispred zadnje nastavne jedinice u cjelini nalaze još barem dvije nastavne jedinice u nizu),</li> <li>- kod promjena udaljenosti od korijena (osim ako bi testu prethodile samo dvije nastavne jedinice ili ako bi iza testa ostale samo dvije nastavne jedinice)</li> <li>- zadnji element računalom oblikovanog nastavnog sadržaja mora biti test</li> </ul>
Osrednji	<ul style="list-style-type: none"> <li>- kod promjene oznake cjelina (jedino ako se ispred zadnjeg elementa u cjelini nalaze još barem dva elementa u nizu koja nisu test),</li> <li>- kod promjena udaljenosti od korijena (osim ako bi testu prethodila samo dva elementa koja nisu test ili ako bi iza testa ostale samo dva elementa koja nisu test),</li> <li>- nakon tri elementa u nizu koje nisu test (osim ako bi testu prethodila samo jedan element koji nije test ili ako bi iza testa ostao samo jedan element koji nije test)</li> <li>- zadnji element računalom oblikovanog nastavnog sadržaja mora biti test</li> </ul>
Napredni	<ul style="list-style-type: none"> <li>- kod promjene oznake cjelina (jedino ako se ispred zadnjeg elementa u cjelini nalaze još barem dva elementa u nizu koja nisu test),</li> <li>- kod promjena udaljenosti od korijena (osim ako bi testu prethodila samo dva elementa koja nisu test ili ako bi iza testa ostale samo dva elementa koja nisu test)</li> <li>- zadnji element računalom oblikovanog nastavnog sadržaja mora biti test</li> </ul>
Stručnjak	<ul style="list-style-type: none"> <li>- zadnji element računalom oblikovanog nastavnog sadržaja mora biti test</li> </ul>

Tablica 2.10. Prezentiranje znanja prema stereotipovima

Stereotip prema znanju	Razina znanja
Novak	Reprodukcija
Početak	Razumijevanje
Osrednji	Razumijevanje
Napredni	Primjena
Stručnjak	Analiza, sinteza i vrednovanje

U Tablica 2.4 smo definirali koju razinu znanja posjeduje koji stereotip. Vidimo da se stereotipovima *osrednji*, *napredni* i *stručnjak* znanje prezentira na istoj razini koja je njihova, dok se stereotipovima *novak* i *početnik* znanje prezentira na za jedan višoj razini od one koju oni posjeduju jer se zahtijeva njihov što brži pomak prema barem *osrednjem* stereotipu.

Razina elementa računalom oblikovanog nastavnog sadržaja nam određuje količinu znanja koju će učenik učiti u jednoj iteraciji učenja i poučavanja, dok Bloomova taksonomija definira način prezentiranja znanja učeniku koji se podudara s učenikovim stereotipom. Dakle, prezentiranje elemenata računalom oblikovanog nastavnog sadržaja promatramo s dva aspekta:

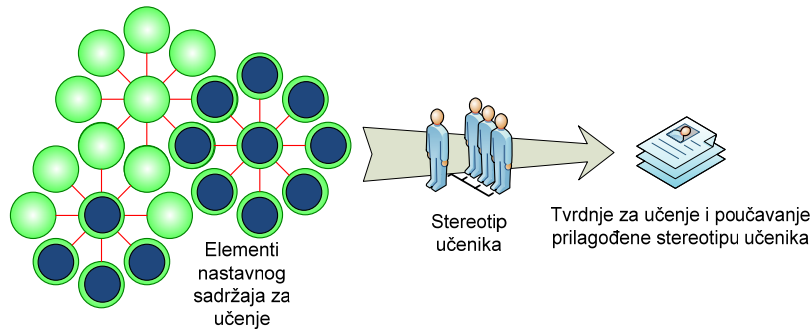
- prema razini elementa računalom oblikovanog nastavnog sadržaja (nastavna cjelina, tema i jedinica)
- prema Bloomovoj taksonomiji (dosjećanje, razumijevanje, primjena te zajedno analiza, sinteza i vrednovanje)

Ova dva aspekta zajedno nam omogućavaju prezentiranje elemenata računalom oblikovanog nastavnog sadržaja na sljedeće načine (Tablica 2.11):

Tablica 2.11. Stereotipovi i količina i razina znanja koja se prezentira

Stereotip prema znanju	Razina elementa nastavnog sadržaja	Razina znanja
Novak	Nastavna jedinica	Reprodukcija
Početak	Nastavna jedinica	Razumijevanje
Osrednji	Nastavna tema	Razumijevanje
Napredni	Nastavna tema	Primjena
Stručnjak	Nastavna cjelina	Analiza, sinteza i vrednovanje

Način na koji će se znanje prezentirati učeniku u procesu učenja i poučavanja je definirano predlošcima za generiranje tvrdnji koje učenik mora usvojiti. Pošto smo definirali četiri težinske kategorije pitanja koje odgovaraju Bloomovoj taksonomiji znanja, definirat ćemo i odgovarajuće četiri težinske kategorije tvrdnji za učenje i poučavanje. Svaka težinska kategorija prezentira određenu razinu znanja: prva težinska kategorija sadrži predloške koji prezentiraju znanje na razini reprodukcije, druga na razini razumijevanja, treća na razini primjene, a četvrta na razini analize, sinteze i procjene (Slika 2.15).



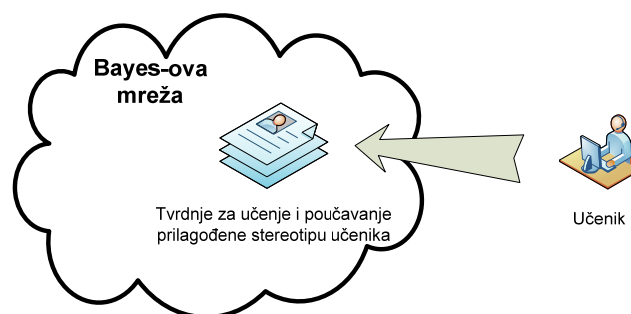
Slika 2.15. Generiranje tvrdnji za učenje i poučavanje

Veza stereotipova, razine znanja koja im se prezentira i predložaka tvrdnji za prezentiranje znanja su prikazani u Tablica 2.12.

Tablica 2.12. Razine znanja, stereotipovi i vrste tvrdnji s predlošcima

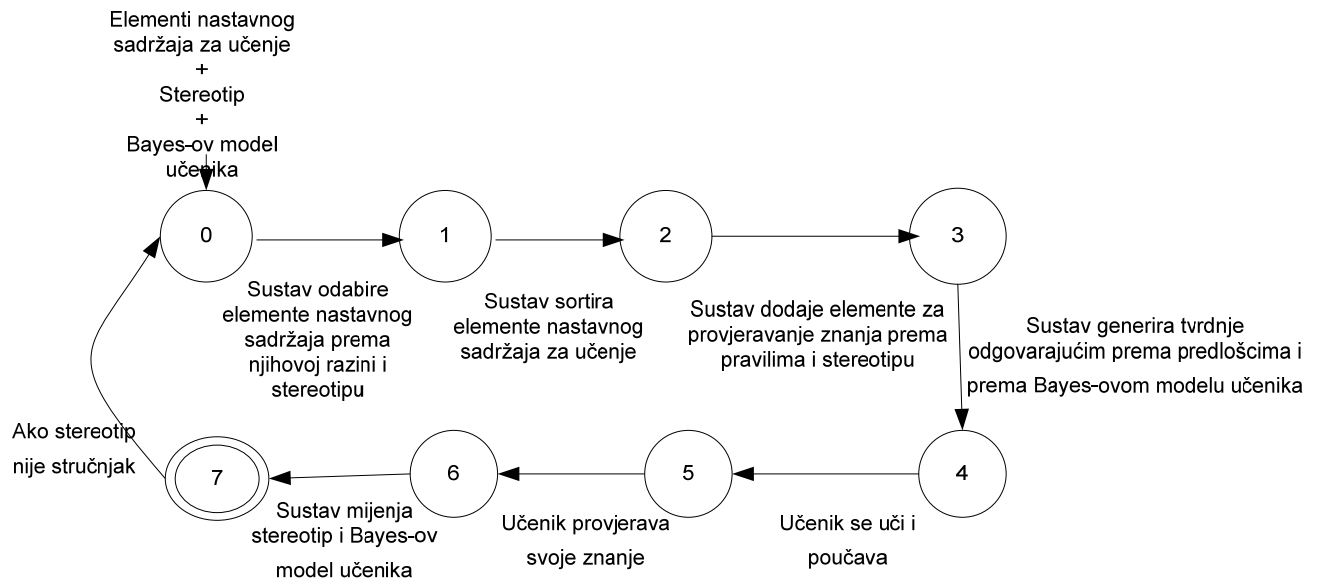
Težinska kategorija tvrdnji	Razina znanja koja se prezentira	Predlošci tvrdnji
1	Reprodukcija	$K_y r K_x$ . $K_x$ je podkoncept od $K_y$ . $K_x r K_y$ . $K_x$ je nadkoncept od $K_y$ .
2	Razumijevanje	$K_x$ i $K_y$ su posredno povezani jer postoji put $Setnja_{K_x K_y}$ $K_x r: K_{y_1}, \dots, K_{y_n}$ . Ovo su podkoncepti od $K_x$ . $K_{y_1}, \dots, K_{y_n} r K_x$ . Ovo su nadkoncepti od $K_x$ . $K_y r K_x$ . $K_x$ ima nadkoncept $K_y$ . $K_x r K_y$ . $K_x$ ima podkoncept $K_y$ .
3	Primjena	$K_x$ je neposredno povezan s $nK_x + pK_x$ koncepata. Ti koncepti su: $K_1, \dots, K_n$ . $K_x$ ima $pK_x$ podkoncepta. $K_x r_1 K_{y_{1_1}}, \dots, K_{y_{n_1}}, \dots, K_x r_m K_{y_{1_m}}, \dots, K_{y_{n_m}}$ . $K_x$ ima $nK_x$ nadkoncepta. $K_{y_{1_1}}, \dots, K_{y_{n_1}} r_1 K_x, \dots, K_{y_{1_m}}, \dots, K_{y_{n_m}} r_m K_x$ .
4	Analiza, sinteza i vrednovanje	Koncepti su neposredno povezani sljedećim redoslijedom: $K_1, r_1, \dots, r_n, K_{n+1}$ $K_{slot}$ od $K_x$ je $K_{filter}$ .

Jedan od glavnih problema modeliranja učenika u sustavima koji koriste stereotipove je određivanje stereotipa koji najbolje opisuje učenika, te u čemu se individualni učenik razlikuje od svog stereotipa. U našem pristupu razlikovanje učenika od pridijeljenog mu stereotipa opisujemo pomoću Bayesovog modela učenika koji omogućava da sustav učeniku, u svakom ciklusu učenja i poučavanja, prezentira samo one tvrdnje koje sadrže ono znanje koje učenik vjerojatno ne posjeduje (Slika 2.16).



Slika 2.16. Prezentiranje tvrdnji za učenje i poučavanje učeniku

Realizaciju učenja i poučavanja u modulu učitelja novog modela za prilagodljivo stjecanje znanja opisujemo strojem s konačnim brojem stanja (Slika 2.17):

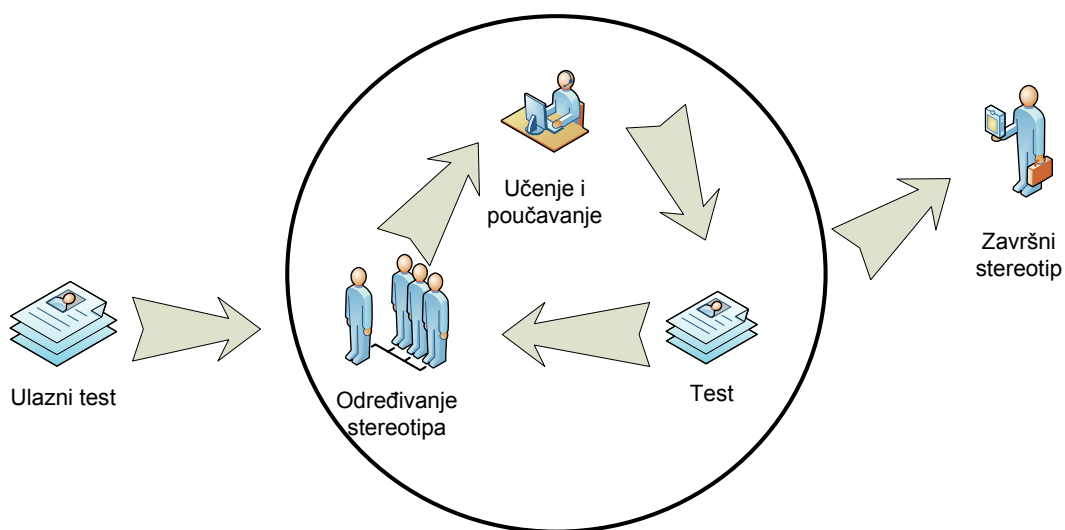


Slika 2.17. Realizacija učenja i poučavanja

Cijeli proces učenja i poučavanja u sustavu koji podržava automatsko i dinamičko generiranje, te prilagodljivi odabir, nizanje i prezentiranje računalom oblikovanog nastavnog sadržaja temeljeno na rezultatima testa, a prilagođeno stereotipu koji se pridruži učeniku upravo prema rezultatima tog istog testa, se sastoji od ciklusa:

**učenje i poučavanje prilagođeno stereotipu → testiranje → određivanje stereotipa**

Ulaz u proces učenja i poučavanja, odnosno u njegov prvi ciklus, je početni stereotip određen na temelju ulaznog testa. Svaki ciklus završava određivanjem novog stereotipa učenika (ili eventualno zadržavanjem već pridijeljenog stereotipa iz prošlog ciklusa) (Slika 2.18).

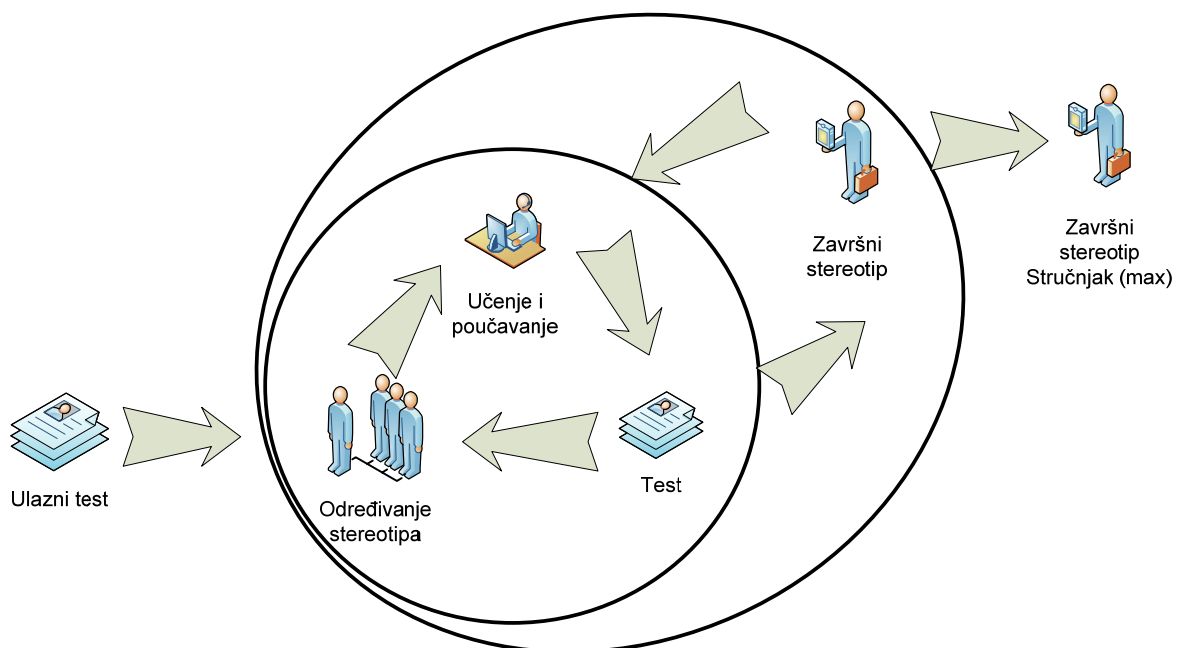


Slika 2.18. Ciklusi u procesu učenja i poučavanja



U svakom ciklusu učenik se uči i poučava samo na jednom dijelu područnog znanja, koje mu je u tom ciklusu prezentirano na način koji je prilagođen onom stereotipu koji je rezultat prethodnog ciklusa. Tada se pristupa testiranju tog znanja koje dovodi do mijenjanja Bayesovog modela učenika (mijenjaju se vrijednosti težinskih funkcija koje ga opisuju), te poželjno mijenjanje stereotipa na više. Proces učenja i poučavanja završava kada učenik pokaže poznavanje svih koncepata barem na razini 1. Izlaz iz cijelog procesa učenja i poučavanja predstavlja određivanje učenikova *završnog stereotipa*.

Primijetimo da učenik ne može završiti proces učenja i poučavanja kao *novak*. Ukoliko učenik želi završiti proces učenja i poučavanja kao bolji stereotip, onda ponovno ulazi u proces učenja i poučavanja s tim da ne treba pisati ulazni test, već mu je početni stereotip onaj koji je bio izlaz u prethodnom procesu učenja i poučavanja (Slika 2.19).



Slika 2.19. „Popravljanje“ završnog stereotipa (do max. stručnjak)

Prednost našeg pristupa nad ostalima je u tome što se stereotip učenika može promijeniti tijekom samog procesa učenja i poučavanja. Naime, sustav UnixConsultant – UC (Chin, 1989), kao jedan od sustava s dvostrukim stereotipnim modelom učenika, odredi stereotip učenika na samom početku i više ga ne mijenja, jer ne postoji mehanizam za ponovnu procjenu podataka i eventualnu promjenu stereotipa. U samom početku, sustav UC smatra da je učenika vjerojatno početnik. Nakon toga svaki podatak o tome što učenik zna može eliminirati neke stereotipove kojima učenik može pripadati. Stereotipovi se eliminiraju ako podaci pokažu da učenik ne može biti član tog stereotipa. Kada se eliminiraju svi osim jednog stereotipa, onda učenik pripada upravo tom stereotipu. Jednom eliminirani, stereotipovi se više ne uzimaju u obzir u UC-u.

Uobičajeno je da poslije procesa učenja i poučavanja učenik pristupa postupku utvrđivanja i provjeravanja kvalitete znanja. Ovaj se postupak naziva testiranje znanja. Sustavi koji dinamički generiraju nastavni sadržaj, provjeravaju znanje učenika prije i nakon prezentiranja novih nastavnih sadržaja. Sve provjere se vrše u svrhu što bolje prilagodbe nastavnog sadržaja aktualnom učenikovom znanju (stereotipu). Provjeravanje znanja, prije obrade novih nastavnih sadržaja, se provodi zbog određivanja stupnja učenikovog predznanja, odnosno određivanja stereotipa učenika, kako bi se nastavni sadržaj mogao prilagoditi učeniku. Provjeravanje znanja nakon učenja se radi kako bi se utvrdio napredak učenika, i po potrebi nova prilagodba nastavnog sadržaja, ovisno o učenikovom napretku.



U ovom poglavlju opisana je zamisao modela inteligentnog tutorskog sustava za prilagodljivo stjecanje znanja temeljeno na stereotipovima, Bayesovim mrežama i Bloomovoj taksonomiji znanja - Adaptive Courseware Tutor Model (AC-ware Tutor). Nakon prikaza tradicionalne strukture inteligentnih tutorskih sustava (ITS), navodi se motivacija za nadogradnjom takve strukture novim elementima kojima bi se povećala prilagodljivost u inteligentnim tutorskim sustavima. Da bi se bolje objasnila ideja koja je realizirana u novom modelu, pristupilo se razjašnjavanju prilagođavanja u sustavima e-učenja općenito. Navedeno je usmjerilo istraživanje u skladu s odrednicama modela na kojima leži cjelokupna zamisao koja je ovdje iznesena. Poboljšanja koja se uvode u tradicionalnu strukturu ITS-a s ciljem povećanja prilagodljivosti prema znanju učenika, opisana su prema komponentama ITS-a u koje se uvode. Za svaki modul ITS-a je dana teorijska podloga na kojoj se zasnivaju poboljšanja koja se uvode, kao i sama formalizacija u modelu AC-ware Tutor. Tako se u modul stručnjaka uvode poboljšanja u formi određivanja reprezentacije područnog znanja koja je nužna za realizaciju ulaznog testa kojim se inicijalizira model učenika. U modul učenika se uvode dvostruki stereotipovi prema razinama stručnosti učenika i prema Bloomovoj taksonomiji znanja, te Bayesove mreže za predikciju znanja koje se kombiniraju s modelom prekrivanja. Najviše se promjena uvodi u modul učitelja koji je zadužen za automatsko generiranje elemenata računalom oblikovanog nastavnog sadržaja, njihov dinamički odabir i sortiranje, te prilagodljivo prezentiranje korištenjem predložaka za tvrdnje i pitanja. Prilagođavanje u našem pristupu je realizirano nad „atomima“ znanja, nad samim konceptima, čime se našim pristupom realizira maksimalna prilagodljivost prema znanju.

U sljedećem poglavlju obradit će se problematika prilagođavanja općenito, s naglaskom na prilagođavanje u sustavima e-učenja, posebice inteligentnim tutorskim sustavima. Također, obrađeni su sustavi koji implementiraju ideju stereotipnog modela učenika, kao i oni koji modeliraju učenikovo znanje temeljem Bayesovih mreža.

## 3 Stanje istraženosti

„To teach successfully we must tell all we know, but only what is adaptable to the student.“

Frederic-Cesar La Harpe

Problematika kojom se bavi ova doktorska disertacija je *promjena poučavanja učenika prilagođavanjem učeniku*. Upravo je to odlika ljudskih tutora koji su učinkoviti jer održavaju ravnotežu dozvoljavajući učeniku da radi koliko može i da ima osjećaj kontrole, nudeći mu pritom dovoljno pomoći kako ne bi postao frustriran ili zbunjen ((Bloom, 1984), (Cohen, Kulik & Kulik, 1982), (Lepper, Aspinwall, Mumme, & Chabay, 1990)).

Tutori učenicima omogućuju *učenje kroz rad* (eng. learn by doing), ostavljajući učenicima da rade većinu posla pri rješavanju problema. Problem se javlja kada učenici prilikom traženja rješenja troše mnogo vremena zbog loših strategija učenja ili pogrešaka. Ponekad se iz tog traženja može nešto i naučiti, ali većinom učenici postaju zbunjeni i frustrirani, a moguće je i da krivo odrede što je izvor pogreške pa tako stječu pogrešno znanje. Pomoć tutora omogućuje *vođeno učenje kroz rad* (eng. guided learning by doing) gdje im tutor pomaže da ne bi krenuli krivim putem (Merrill, Reiser, Ranney, & Trafton, 1992). To znači da tutor složeni zadatak dijeli na više dijelova, obavlja dio zadatka, započinje rješavanje zadatka i pušta učenika da napravi ostatak, podsjeća učenika o nekim aspektima zadatka, itd. (Chi, Siler, Jeong, Yamauchi, & Hausmann, 2001).

### **3.1 Razvoj sustava za generiranje prilagodljivog računalom oblikovanog nastavnog sadržaja**

Najvažnija karakteristika tutorske nastave je u sposobnosti ljudskog tutora da prilagodi proces učenja i poučavanja učeniku uzimajući u obzir njegove određene značajke. Prvo ćemo objasniti što je to prilagođavanje i koje su najčešće korištene izvedenice te riječi, da bi utvrdili terminologiju koja se primjenjuje u ovom istraživanju. Objašnjenje terminologije je bitno i zbog utvrđivanja stanja istraženosti, jer je potrebno utvrditi koji su drugi pristupi realiziranju prilagodljivosti i usporediti novi pristup koji mi predlažemo. Zatim ćemo objasniti što je to prilagodljiva nastava, kao i primjenu prilagodljive nastave u okvirima računalnih sustava. U ovom dijelu doktorske disertacije analiziramo različite pristupe za generiranje prilagodljivog računalom oblikovanog nastavnog sadržaja, kao i sustave koji implementiraju određeni pristup generiranju prilagodljivog računalom oblikovanog nastavnog sadržaja.

## **Prilagođavanje**

Riječ „prilagoditi se“ (eng. *adapt*) interpretiramo kao „učiniti prikladnim mijenjajući trenutno stanje“. U stranim rječnicima definira se na sljedeće načine (namjerno ne prevodimo da se ne bi izgubio smisao):

- „*make fit for, or change to suit a new purpose*“ ili „*conform oneself to new or different conditions*“ – WordNet ([wordnetweb.princeton.edu/perl/webwn](http://wordnetweb.princeton.edu/perl/webwn))
- „*to make fit (as for a specific or new use or situation) often by modification*“ ili „*to become adapted*“ – Merriam Webster ([www.merriam-webster.com](http://www.merriam-webster.com))
- „*make (something) suitable for a new use or purpose; modify*“ ili „*become adjusted to new conditions*“ – Oxford ([oxforddictionaries.com](http://oxforddictionaries.com))

Prvo značenje ovih definicija je mijenjanje nečega da bi se ispunili određeni zahtjevi ili svrha. Drugo značenje uključuje čin navikavanja na promjenjivo okruženje.

U engleskom jeziku korijen *adapt* se kombinira s različitim sufiksima i na taj način dobivaju se riječi s različitim značenjem i različitim primjenama. Navodimo samo one morfološke varijante riječi koje su važne za područje ove doktorske disertacije: *adaptive* (sposoban za provođenje prilagođavanja – prvo značenje), *adaptable* (sposoban da ga se prilagodi – drugo značenje), *adaptation* (proces provođenja prilagođavanja drugoga i prilagođavanja samog sebe – prvo i drugo značenje), *adaptivity* (odnosi se na kvalitet i kapacitet adaptivnosti), *adaptability* (odnosi se na kvalitet i kapacitet adaptabilnosti), *adapter* ili *adaptive engine* (komponenta koja prepoznaje, započinje i izvršava prilagođavanje – stroj za prilagođavanje), *adapter* (nešto što zamjenjuje ili dopunjuje objekt prilagođavanja koji se ne smije preoblikovati tijekom procesa prilagođavanja).

## **Prilagodljiva nastava**

Pošto je nemoguće uvijek ostvariti tutorsku nastavu u procesu učenja i poučavanja, treba pronaći načine kako ostvariti približno jednaku učinkovitost uporabom različitih metoda i tehnika prilagođavanja u drugim oblicima nastave. Nastavni pristupi i tehnike koje su usmjerene prema potrebama pojedinog učenika nazivaju se *prilagodljiva nastava* (eng. *adaptive instruction*) (Corno & Snow, 1986). Dakle, prilagodljiva nastava se odnosi na obrazovne intervencije namijenjene za učinkovito prilagođavanje pojedinačnim razlikama učenika koje mu pomažu da razvije znanja i vještine potrebne za izvršavanje određene zadaće. Prilagodljiva nastava se općenito karakterizira kao pristup obrazovanju koji uključuje alternativne postupke i strategije te fleksibilnost za učenje (Wang & Lindvall, 1984, prema (Park & Lee, 1996)).

Bilo koji oblik nastave jedan-na-jedan može se smatrati individualiziranom nastavom (nastava usmjerena na učenika). Ipak, ako individualizirana nastava nije dovoljno fleksibilna prema učenikovim specifičnim potrebama, ne može se smatrati prilagodljivom. Slično, nastava u okruženju grupe može biti prilagodljiva ako je osjetljiva na jedinstvene potrebe

svakog učenika, kao i na zajedničke potrebe grupe. Idealna individualizirana nastava treba biti prilagodljiva, jer je nastava najučinkovitija kad je prilagođena jedinstvenim potrebama svakog učenika (Park & Lee, 1996).

Prilagođavanje nastave pojedinačnim učenikovim potrebama ima dugu povijest. Još se u četvrtom stoljeću prije Krista, prilagođavanje smatralo glavnim čimbenikom uspjeha nastave (Corno & Snow, 1986), a prilagodljiva tutorska nastava je bila uobičajena metoda obrazovanja do sredine 19. stoljeća (Reiser, 1987). Tijekom 20. stoljeća stalno se naglašavala važnost prilagođavanja nastave specijalizacijom kurikuluma za različite razrede i različite učenike (Park & Lee, 1996). Počela su se provoditi brojna istraživanja koja su imala cilj utvrditi koje učenikove značajke treba uzeti u obzir prilikom prilagođavanja nastave pojedincu te kako nastavne metode i procedure prilagoditi tim značajkama.

Značajke učenika prema kojima se treba prilagoditi nastava se nazivaju *sklonosti ili sposobnosti* (eng. aptitude), a pristup prilagođavanja je *interakcija sklonosti i tretmana* (eng. aptitude-treatment interaction - ATI). Sklonost ili sposobnost je bilo koja individualna značajka koja povećava ili umanjuje vjerojatnost učenikova uspjeha, a tretman je varijacija u načinu poučavanja ((Cronbach & Snow, 1977), (Snow & Swanson, 1992)).

Sklonosti ili sposobnosti za učenje mogu biti kognitivne ili konativne i afektivne (Snow & Swanson, 1992). Kognitivne sklonosti ili sposobnosti uključuju: (a) intelektualne sposobnosti (analitičko zaključivanje, vizualno-prostorne, verbalne, matematičke sposobnosti, kapacitet pamćenja, mentalna brzina); (b) kognitivne stilove i stilove učenja, te (c) predznanje. Konativne i afektivne sklonosti ili sposobnosti uključuju: (a) motivaciju, te (b) samoprocjenu.

Model primjene ATI-a za generiranje računalom oblikovanog nastavnog sadržaja se sastoji od osam koraka: (1) identificiraj ciljeve; (2) specificiraj karakteristike zadaće; (3) identificiraj inicijalni skup značajki učenika; (4) odaberi najvažnije učenikove značajke; (5) analiziraj učenike u ciljanoj populaciji; (6) odaberi konačne razlike (u učenikovim karakteristikama); (7) odredi kako prilagoditi nastavu; (8) oblikuj alternativne načine poučavanja (Carrier & Jonassen, 1988). Ovaj model je zapravo modificirani pristup oblikovanju nastave ((Gagne & Briggs, 1979), (Dick & Carey, 1985)).

Implementiranje ATI modela zahtjeva odgovore na sljedeća pitanja (Shute, Lajoie, & Gluck, 2000): koje značajke treba mjeriti prije učenja i poučavanja, kojim varijablama treba manipulirati, koji indikatori učenja se koriste za mjerenje napretka u učenju i poučavanju, te koje mjere učinkovitosti koristiti? U odgovorima na ova pitanja može pomoći *taksonomija vještina učenja* (eng. learning skills) koju su razvili (Kyllonen & Shute, 1988). Ova taksonomija definira četiri-dimenzionalni prostor koji uključuje područno znanje, okruženje za učenje i poučavanje, željeni rezultat procesa učenja i poučavanja i attribute učenika.

Kako se povećavala primjena računala, počeli su se razvijati sustavi programske podrške na koje su se primjenjivala spomenuta načela prilagodljivosti.

## **Prilagodljivi sustavi**

*Prilagodljivi sustavi* (eng. adaptive systems - AS) su sustavi koji mogu mijenjati svoju strukturu, funkcionalnosti ili sučelje da bi se prilagodili različitim potrebama pojedinca ili grupe, kao i promjenama njihovih potreba tijekom vremena (Benyon & Murray, 1993). Prilagodljivi sustavi mogu biti *adaptabilni* (eng. adaptable - adaptability) i *adaptivni* (end. adaptive - adaptivity) (Oppermann & Rasher, 1997). Sustavi koji dozvoljavaju samim učenicima mijenjanje određenih parametara i prilagođavaju svoje ponašanje u skladu s time se nazivaju *prilagodljivim adaptabilnim sustavima*. Sustavi koji se automatski prilagođavaju učeniku na temelju svojih pretpostavki o učeniku se nazivaju *prilagodljivi adaptivni sustavi*. U nastavku ćemo se orijentirati isključivo na adaptivne sustave koji omogućavaju automatsko prilagođavanje.

Prilagodljivi sustavi i prilagodljiva nastava definiraju novu klasu sustava - *prilagodljive sustave e-učenja* (eng. adaptive e-learning systems) koji prilagođavaju proces učenja, poučavanja i testiranja znanja različitim značajkama učenika, tj. prilagođavaju odabir i prezentaciju sadržaja za učenje statusu, potrebama, stilu, predznanju i preferencama učenika (Brusilovsky & Nijhawan, 2002), (Santos, Anido, Llamas, & Rodríguez, 2002), (Shute & Towle, 2003), (Paramythis & Loidl-Reisinger, 2003). Ponekad se koristi i naziv *prilagodljivi obrazovni sustavi* (eng. adaptive educational systems - AES) (Brusilovsky, 1998).

U sustavima e-učenja prilagodljivost se realizira prema *ciljevima učenika* (Kaplan i dr., 1993, Grunst, 1993, Vassileva 1996), *predznanju* (Milosavljevic, 1997, Hockemeyer i dr., 1998, Kayama i Okamoto, 1998), *stilovima učenja* (Gilbert i Han, 1999, Specht i Oppermann, 1998), *stručnosti* (Pérez i dr., 1995, Vassileva, 1996), te prema brojnim drugim karakteristikama učenika (sve prema (Dagger, Wade, & Conlan, 2005)).

Dva najpoznatija predstavnika prilagodljivih sustava e-učenja su *inteligentni tutorski sustavi* (eng. intelligent tutoring systems - ITS) i *sustavi prilagodljive hipermedije* (eng. adaptive hypermedia systems – AHS). Značajke učenika prema kojima se prilagođavaju ITS-ovi su orijentirane prema znanju (ciljevi, predznanje, stručnost, i sl.), dok se AHS-ovi uglavnom prilagođavaju prema stilovima učenja.

Postoje dva glavna pristupa u ostvarivanju prilagodljivosti (Shute, Lajoie & Gluck, 2000): makro- i mikro-prilagođavanje. *Makro-prilagođavanje* se događa prije procesa učenja i poučavanja. Najprije treba sakupiti podatke o kognitivnim sposobnostima učenika, zatim upotrijebiti te informacije za donošenje odluka o vrsti okruženja za učenje i poučavanje koje će najbolje odgovarati tim odlukama. *Mikro-prilagođavanje* se događa tijekom samog procesa učenja i poučavanja. Uključuje promjene u onome ŠTO je prezentirano, a ne u onome KAKO je prezentirano. Ove odluke se donose na temelju trenutnog znanja učenika kada ga se uspoređuje sa znanjem kojeg bi trebali imati kada proces učenja i poučavanja završi. Ako učenik pokazuje manjak znanja, prezentira mu se novi sadržaj ili mu se ponovno prezentira stari. Ako učenik pokaže znanje, dozvoljava mu se preskakanje određenih dijelova.

Bez obzira o kojem pristupu prilagođavanju se radi, treba voditi računa o sljedećem (Bork, Bagley i Zhang, 2008, prema (Riad, El-Minir & El-Ghareeb, 2009)): svaki učenik treba učiti svojim ritmom; prilagođavanje se treba događati često; svaki učenik treba uspješno završiti proces učenja i poučavanja; kada je nešto uspješno naučio, učenik treba nastaviti dalje; nikoga ne treba poučavati ono što već zna.

Kad promatramo prilagodljivost u sustavima e-učenja, posebice inteligentnim tutorskim sustavima, moramo razlučiti koje komponente sustava se moraju prilagoditi na temelju kojih informacija, kada, na koji način i zašto (Mödrtscher, 2008):

1. Informacija za prilagođavanje (eng. adaptation information) određuje što se koristi kao izvor prilagođavanja. Najčešće je to znanje o učenicima. Ipak, prilagodba se može temeljiti na bilo kojem stanju iz okruženja sustava. Stoga, prilagodljivi sustavi posjeduju najmanje jednu komponentu koja promatra njihovo okruženje.
2. Pravila prilagođavanja (eng. adaptation rules) su neophodna za donošenje odluke kada započeti proces prilagođavanja. Ova pravila se temelje na informaciji za prilagođavanje te se smatraju okidačima prilagođavanja.
3. Procedure prilagođavanja (eng. adaptation procedures) označavaju koje komponente sustava (tzv. adaptori) vrše prilagođavanje i na koji način.
4. Predmeti prilagođavanja (eng. adaptation targets) se odnose na ciljeve prilagođavanja, ali ne i na njihove izvore. Stoga komponente koje modeliraju ciljeve prilagođavanja također moraju promatrati stanja iz okruženja, te se koriste za vrednovanje učinka prilagođavanja.

Često se pojam *personalizacija* koristi kao sinonim za prilagođavanje. Međutim, personalizirani sustavi su tek posebna vrsta prilagodljivih sustava, kod kojih je predmet prilagođavanja učenik (García-Barrios, Mödrtscher, & Gütl, 2005). Isti autori definiraju pet dimenzija personalizacije:

1. Eksplicitna – implicitna: Eksplicitna personalizacija opisuje prilagođavanje prema konkretnom modelu učenika. Implicitna personalizacija se odnosi na prilagođavanje u određenom kontekstu bez eksplicitne uporabe informacija o učeniku.
2. Vidljiva – skrivena: Personalizacija je vidljiva ako učenik prepoznaje rezultate personalizacije. Skrivena personalizacija nije vidljiva učeniku.
3. Predvidiva – determinirana: Predvidljiva personalizacija obuhvaća unaprijed pripremljene korake prilagođavanja. Determinirana personalizacija se odvija unutar jednog koraka prilagođavanja.
4. Kontrolirana – nekontrolirana personalizacija: Kontrolirana personalizacija omogućava učeniku preuzimanje kontrole nad procesom prilagođavanja u bilo kojem trenutku. Nekontrolirana personalizacija ne dozvoljava učeniku da utječe na proces prilagođavanja.
5. Individualna – stereotipna: Individualna personalizacija uključuje personalizaciju prema jednom učeniku. Stereotipna personalizacija se odnosi na grupu učenika.

Metoda prilagođavanja je određena idejom o prilagođavanju definiranoj na konceptualnoj razini, dok je tehnika prilagođavanja realizacija metode. Prilikom određivanja načina prilagođavanja, potrebno je voditi računa o: sustavu prilagođavanja (GDJE), ciljevima prilagođavanja (ZAŠTO); značajkama učenika prema kojima se prilagođava (PREMA ČEMU); tehnologijama prilagođavanja (ŠTO – sadržaj ili navigacija); metodama i tehnikama prilagođavanja (KAKO) (Brusilovsky, 1996).

Tehnologije prilagođavanja koje se koriste u ITS-ovima su nizanje sadržaja, inteligentna analiza učenikovih rješenja, podrška interaktivnom rješavanju problema, podrška rješavanju problema temeljeno na primjerima, podrška suradnji, itd., a u AHS-ovima prilagodljiva prezentacija i podrška prilagodljivoj navigaciji (Brusilovsky, 1998).

Ponašanja koja treba mjeriti u prilagodljivim sustavima e-učenja, a naročito u inteligentnim tutorskim sustavima, su ona koja se mogu predvidjeti na temelju rezultata određene nastavne tehnike. Znanje omogućava valjanije i pouzdanije osnove za utvrđivanje rezultata prilagođavanja od ostalih sklonosti ili sposobnosti (Glaser & Nitko, 1970).

Pošto je učenikovo znanje glavni razlog postojanja sustava e-učenja općenito, usmjerili smo naše istraživanje prema prilagođavanju računalom oblikovanog nastavnog sadržaja u inteligentnim tutorskim sustavima, odnosno proučavamo nizanje sadržaja i sadržaj prilagođavanja (ŠTO), a ne način prilagodljivog prezentiranja (KAKO), što je karakteristično za sustave prilagodljive hipermedije.

### ***Generiranje računalom oblikovanog nastavnog sadržaja***

Unatoč činjenici da se s elementarnim oblicima prilagođavanja nastavnog sadržaja kod računalnih tutora započelo prije dvadesetak godina, referentni model za generiranje računalom oblikovanog nastavnog sadržaja još nije definiran (Ullrich, 2008). Postoje mnogi nazivi za generiranje računalom oblikovanog nastavnog sadržaja: nizanje sadržaja (eng. course(ware) sequencing), nizanje kurikuluma (eng. curriculum sequencing), generiranje putanje (eng. trail generation), planiranje kolegija (eng. course planning), modeliranje nastave (eng. instructional modeling), prilagodljivo nizanje objekata učenja (eng. adaptive sequencing of learning objects), planiranje nastave (eng. instructional planning). Također se pojmovi kao što su dinamičko, prilagodljivo, personalizirano ili inteligentno generiranje računalom oblikovanog nastavnog sadržaja koriste u značenju kojeg su definirali Chapelle i Mizuno (Chapelle & Mizuno, 1989): generiranje računalom oblikovanog nastavnog sadržaja osjetljivog na učenikove potrebe.

Bez obzira na korištenu terminologiju, generiranje računalom oblikovanog nastavnog sadržaja obuhvaća *planiranje sadržaja* i *planiranje isporuke* tog sadržaja (Dijkstra, Krammer, & Van Merriënboer, 1992). Planiranje sadržaja uključuje generiranje, nizanje i odabir elemenata sadržaja na temelju aktualnog znanja učenika, te nadgledanje izvršavanja plana sadržaja da bi se utvrdilo kada treba pre-planirati ili generirati novi plan. Planiranje isporuke



---

sadržaja ili, kako se još naziva, strategije učenja i poučavanja (eng. teaching strategies) se odnosi na odabir aktivnosti i interakcija koje će pomoći učeniku ostvariti ciljeve (Vassileva & Wasson, 1996).

Generiranje računalom oblikovanog nastavnog sadržaja je proces odabira elemenata područnog znanja i njihovog nizanja na način koji je prikladan za ciljanu grupu učenika ili pojedinca (Knolmayer, 2003, prema (Karampiperis & Sampson, 2004b), (Mohan et al., 2003)), te se smatra najzanimljivijim područjem istraživanja sustav e-učenja ((Brusilovsky & Vassileva, 2003), (Karampiperis & Sampson, 2004)).

Generiranje računalom oblikovanog nastavnog sadržaja je dobro utemeljena metoda prilagođavanja koja se koristi kod inteligentnih tutorskih sustava. Ideja generiranja računalom oblikovanog nastavnog sadržaja je u stvaranju individualiziranih nastavnih sadržaja za svakog učenika, te dinamičkim odabirom optimalnog načina poučavanja u svakom koraku procesa učenja i poučavanja. Optimalan način poučavanje je onaj koji će u tom trenutku dovesti učenika bliže ostvarenju cilja učenja. Najčešće je cilj naučiti traženo znanje u najkraćem roku. ITS koji generira računalom oblikovani nastavni sadržaj, predstavlja područno znanje kao mrežu koncepata, gdje svaki koncept predstavlja mali dio područnog znanja. Mehanizam za generiranje računalom oblikovanog nastavnog sadržaja temelji se na modelu učenika koji predstavlja težinsko prekrivanje s modelom područnog znanja (za svaki koncept područnog znanja, model učenika mora sadržavati aktualnu razinu učenikovog poznavanja tog koncepta) (Brusilovsky & Vassileva, 2003).

Postoje različiti pristupi generiranju računalom oblikovanog nastavnog sadržaja u ITS-ovima. Većina ITS-ova može samo generirati računalom oblikovani nastavni sadržaj koji koristi samo jedan način poučavanja, neki ITS-ovi mogu samo mijenjati redoslijed problema ili pitanja, neki mogu organizirati redoslijed lekcija. Najnapredniji sustavi mogu generirati računalom oblikovani nastavni sadržaj koji koristi više načina poučavanja (Brusilovsky & Vassileva, 2003).

Postoje tri različita pristupa generiranju računalom oblikovanog nastavnog sadržaja: *statički računalom oblikovani nastavni sadržaj*, *prilagodljivo* generiranje računalom oblikovanog nastavnog sadržaja (eng. adaptive *courseware* generation), *dinamičko* generiranje računalom oblikovanog nastavnog sadržaja (eng. dynamic *courseware* generation) (Brusilovsky & Vassileva, 2003).

Tradicionalni ITS se prilagođava procesu učenja i poučavanja učeniku primjenjujući prikladne pedagoške strategije tijekom prezentiranja područnog znanja učeniku (Oppermann & Rasher, 1997). Model računalom oblikovanog nastavnog sadržaja u većini ITS-ova je statički niz (ili stablo) koji slijede statički testovi. Svi elementi su unaprijed stvoreni od strane učitelja. Umjesto inkrementalnog nizanja, cijeli računalom oblikovani nastavni sadržaj treba generirati odjednom, čime se umanjuje prilagodljivost. Ideja prilagodljivog generiranja računalom oblikovanog nastavnog sadržaja je u stvaranju sadržaja prilagođenog potrebama

učenika i to prije nego se učenici susretnu s njim. Dinamičko generiranje računalom oblikovanog nastavnog sadržaja promatra učenikov napredak. Ako učenikove performanse ne odgovaraju očekivanima, sadržaji se dinamički re-generiraju. Između re-generiranja, model računalom oblikovanog nastavnog sadržaja je statičan. Ovaj pristup uzima u obzir učenikovo trenutno znanje, ciljeve, vremenski okvir, te se prilagođava njegovim poteškoćama i tempu napredovanja (Brusilovsky & Vassileva, 2003).

U svim sustavima koji generiraju računalom oblikovani nastavni sadržaj, postoje *elementi područnog znanja* (eng. domain knowledge elements – DKE) koji se u različitim sustavima različito zovu: koncepti, fragmenti znanja, teme, objekti učenja, objekti znanja (Brusilovsky & Vassileva, 2003). Uporaba malenih, ponovno upotrebljivih granula (eng. chunks) se koristi i kod *oblikovanja nastave* (eng. instructional design) (Reigeluth i Nelson, prema (Wiley, 2000)). Naime, kada učitelj prvi put pristupa sadržaju, često ga dijeli na sastavne dijelove koje presloži na način koji će podupirati njegove obrazovne ciljeve. Ove granule znanja mogu reprezentirati veći ili manji dio područnog znanja, ovisno o samom područnom znanju i onome tko ih definira.

Tehnike za realizaciju prilagođavanja računalom oblikovanog nastavnog sadržaja s obzirom na elemente područnog znanja su (Bhaskar, Das, Chithralekha, & Sivasathya, 2010):

1. Graf putanje učenja (eng. learning path graph) je usmjereni aciklički graf koji određuje sve moguće putanje učenja koje odgovaraju zadanom cilju učenja (Karampiperis & Sampson, 2005).
2. Graf putanje koncepata (eng. concept path graph) je usmjereni aciklički graf koji reprezentira skup pravila nizanja kojima se određuje redoslijed koncepata ((Pukkhem, Evens, & Vatanawood, 2006), (Carchiolo, Longheu, Malgeri, & Mangioni, 2003)).
3. Mapa koncepata (eng. concept map) se koristi za grafičko prezentiranje područnog znanja na temelju klastera računalom oblikovanog nastavnog sadržaja i matrice korelacija koncepata koji omogućuju sistematično učenje (Chang, Chang, Heh, & Liu, 2008).
4. Ontologija je prezentirana skupom apstraktnih koncepata i semantičkim vezama između njih, umjesto stvarnog sadržaja za učenje (Gascuena, Fernandez-Caballero, & Gonzalez, 2006).
5. Graf aktivnosti za učenje (eng. learning activity graph) je usmjereni graf koji se koristi za organiziranje sadržaja učenja unutar zadaća (Zhu & Cao, 2008).
6. Bayesova mreža (eng. Bayesian network) je usmjereni graf čiji su čvorovi neizvjesne varijable interesa, a bridovi su uzročne veze između varijabli. Svaki čvor ima pridruženu tablicu uvjetnih vjerojatnosti koje ovise o učenikovim značajkama ((Márquez, Ortega, Gonzalez-Abril, & Velasco, 2008), (Anh, Ha, & Dam, 2008)).

Od navedenih tehnika, najsloženija je Bayesova mreža koja predviđa vjerojatnost svladavanja novog znanja na temelju učenikovih značajki utvrđenih prije učenikovog uključivanja u nastavu, a kasnije mijenjanih na temelju pokazanih performansi učenika (Rothen &

Tennyson, 1978; Tennyson & Christensen, 1988, prema (Park & Lee, 1996)). Na temelju dobivenih vjerojatnosti, odabire se i prilagođava način poučavanja. Istraživanja su pokazala učinkovitost primjene Bayesovog modela vjerojatnosti za prilagođavanje CBI sustava (Tennyson & Park, 1984; Tennyson, Park & Christensen, 1985, Tennyson & Christensen, 1988, prema (Park & Lee, 1996)). Također, pokazalo se da je najteži dio u primjeni Bayesovog modela određivanje „a priori“ vjerojatnosti na temelju rezultata ulaznog testa. Stoga će se posebno objasniti primjena Bayesove mreže u modeliranju učenika.

Shute i Zapata-Rivera predlažu *ciklični četvero-fazni proces prilagođavanja računalom oblikovanog nastavnog sadržaja u sustavima e-učenja*: (a) prikupljanje informacija o učeniku (eng. capture) , (b) analiza (eng. analyze) – stvaranje i održavanje modela učenika, (c) odabir elemenata računalom oblikovanog nastavnog sadržaja temeljem modela učenika (eng. select), te (d) prezentiranje elemenata računalom oblikovanog nastavnog sadržaja temeljem modela učenika (eng. present) (Shute & Zapata-Rivera, 2008).

### ***Prilagodljivi sustavi za generiranje računalom oblikovanog nastavnog sadržaja***

Prije nego što detaljno prikažemo naš pristup automatskom, dinamičkom i prilagodljivom generiranju računalom oblikovanog nastavnog sadržaja, neophodno je obraditi ključne radove iz ovog područja. Stoga ćemo u nastavku poglavlja opisati sustave koji generiraju prilagodljivi računalom oblikovani nastavni sadržaj.

Najveći problem je bilo pronaći relevantne sustave koji odgovaraju upravo navedenoj teoriji. Naime, tražili smo sustave koji automatski i dinamički generiraju prilagodljivi računalom oblikovani nastavni sadržaj, s naglaskom na prilagođavanje sadržaja, a ne sučelja. Grupa autora je dala pregled prilagodljivih sustava e-učenja u razdoblju od 1996. do 2006. godine, s naglaskom na značajku učenika prema kojoj se prilagođavaju, ali nisu naveli koja tehnologija prilagođavanja se koristila (Essalmi, Ayed, Jemni, Kinshuk, & Graf, 2010). Među spomenutim sustavima najviše ima onih koji se prilagođavaju prema znanju učenika.

Od svih sustava koji se prilagođavaju znanju učenika, nas zanimaju samo oni koji automatski generiraju računalom oblikovani nastavni sadržaj: stvaraju, odabiru, nižu i prezentiraju elemente računalom oblikovanog nastavnog sadržaja (za učenje i provjeravanje znanja). Pošto je naš znanstveni doprinos upravo u ovom segmentu, nismo mogli pronaći sustave koji posjeduju sve navedene elemente, pa smo odabrali one koji su najsličniji našem pristupu.

Dakle, ključne riječi koje su nam služile kao temelj za pretraživanje izvora na internetu su: **adaptive e-learning systems, intelligent tutoring systems, courseware generation, courseware sequencing, automatic courseware, dynamic courseware, adaptive courseware, automatic generation of courseware**. Koristili smo i one ključne riječi u kojima smo **courseware** zamijenili s **course** ili **curriculum**. Iz analize sustava smo izbacili prilagodljive sustave hipermedije, pošto oni uglavnom koriste samo prilagodljivu navigaciju.

Rezultati pretraživanja u bazi ScienceDirect ([www.sciencedirect.com](http://www.sciencedirect.com)) su navedeni u sljedećoj tablici:

Tablica 3.1. Rezultat pretraživanja u bazi ScienceDirect

Ključne riječi	Broj referenci
adaptive e-learning systems + <i>courseware</i> (course, curriculum) generation	1
intelligent tutoring systems + <i>courseware</i> (course, curriculum) generation	20
adaptive e-learning systems + <i>courseware</i> (course, curriculum) sequencing	4
intelligent tutoring systems + <i>courseware</i> (course, curriculum) sequencing	23
adaptive e-learning systems + automatic + <i>courseware</i> (course, curriculum)	1
intelligent tutoring systems + automatic + <i>courseware</i> (course, curriculum)	3
adaptive e-learning systems + dynamic + <i>courseware</i> (course, curriculum)	1
intelligent tutoring systems + dynamic + <i>courseware</i> (course, curriculum)	16
adaptive e-learning systems + adaptive <i>courseware</i> (course, curriculum)	3
intelligent tutoring systems + adaptive <i>courseware</i> (course, curriculum)	22
adaptive e-learning systems + automatic generation	2
intelligent tutoring systems + automatic generation	46

Analiza radova (od 142 izvora, mnogi se ponavljaju u različitim pretragama) je pokazala da većina ovih izvora samo spominje pojam inteligentni tutorski sustavi i generiranje, odnosno nizanje računalom oblikovanog nastavnog sadržaja, a u načelu govore o sustavima prilagodljive hipermedije. Navodimo najčešće citirane sustave u sljedećoj tablici:

Tablica 3.2. Pregled pronađenih sustava i njihova citiranost

Rbr	Sustav	Godina	Science Direct	Google Scholar
1	GTE (Van Marcke, 1990)	1990	2	513
2	ECSAI (Gavignet, 1991)	1991	0	5
3	DCG (Vassileva, 1992)	1995	7	209
4	ELM-ART (Brusilovsky, Schwarz, & Weber, 1996)	1996	17	1240
5	AST (Specht, Kravcik, Pesin, & Klemke, 2001)	1997	3	447
6	ACE (Specht & Oppermann, 1998)	1998	3	149
7	KBS HYPERBOOK SYSTEM (Henze & Nejd, 1999)	1999	2	184
8	ATLAS (Macías & Castells, 2001)	2001	3	67
9	INSPIRE (Grigoriadou, Papanikolaou, Kornilakis, & Magoulas, 2001)	2001	6	363
10	WINDS (Specht et al., 2001)	2001	2	142
12	NETCOACH (Weber, Kuhl, & Weibelzahl, 2002)	2002	5	339
11	ApelS (Bruen, Conlan, & Wade, 2002)	2002	0	66
13	ACCT (Dagger, Wade, & Conlan, 2004)	2004	0	340
14	ASM (Karampiperis & Sampson, 2005)	2005	0	74
15	ICLASS (O Keeffe, Brady, Conlan, & Wade, 2006)	2006	0	60
16	PAIGOS (Ullrich, 2008)	2008	1	35
17	ADOPTA (Bontchev & Vassileva, 2009)	2009	0	227

Odabrali smo, dakle, sustave koji se spominju u najviše izvora i koje najveći broj autora smatra najvažnijima u ovom području. Analizirat ćemo kako navedeni sustavi generiraju elemente računalom oblikovanog nastavnog sadržaja, kako ih nižu i kako ih prezentiraju učeniku. Objasnit ćemo kakvo je *područno znanje*, *model učenika*, *generiranje* samog računalom oblikovanog nastavnog sadržaja, kako je realizirana *prilagodljivost*, *značajka* i *element* prilagođavanja, način *provjeravanja znanja* učenika (da se provjerava predznanje i na temelju čega, da li su testovi statični ili se dinamički generiraju i prilagodljivi su). Njihov detaljan prikaz je dan u Prilogu 8.1, a u nastavku je dana njihova usporedba.

Svaki je sustav opisan na način da je ukratko navedena svrha sustava i njegovi autori. Zatim je navedeno da li sustav ima i kakvo je područno znanje, model učenika, računalom oblikovani nastavni sadržaj, provjeravanje znanja i prilagodljivost. Također je eksplicitno navedena značajka učenika prema kojoj se sustav prilagođava, te koji je element prilagođavanja. Ovakva analiza do sada nije bila napravljena na način da su prilagodljivi sustavi jednoobrazno opisani. Naime, na temelji izvornih radove trebalo je izvući opise navedenih karakteristika koje nigdje nisu eksplicitno navedene.

### **3.1.1 Analiza sustava za generiranje prilagodljivog računalom oblikovanog nastavnog sadržaja**

Sada možemo utvrditi glavne razlike i prednosti našeg pristupa naspram spomenutih. Usporedbu ćemo realizirati uspoređujući glavne elemente u svakom od sustava s našim pristupom.

#### ***Područno znanje***

Područno znanje se u početku uglavnom reprezentiralo nekim oblikom grafa ili semantičke mreže (ECSAI - hijerarhijska mreža jedinica, DCG - I/II graf koncepata područnog znanja, ELM-ART - konceptualna mreža jedinica, ACE - graf koncepata područnog znanja, KBS - graf zavisnosti elemenata znanja, INSPIRE - mreža koncepata, NetCoach - mreža koncepata, APeLS - model opisa). Od 2004. godine više se koristi ontološki pristup formalizaciji područnog znanja (ACCT, ASM, iClass, ADOPTA).

U našem pristupu također pratimo novije trendove i formaliziramo područno znanje ontološkim pristupom.

#### ***Model učenika***

Uglavnom se kod modeliranja učenika koristi neka verzija modela prekrivanja (jednostavni ECSAI, ELM-ART, INSPIRE, ASM, težinski GTE, DCG, ACE, višeslojni NetCoach). Kod nekih sustava prisutan je Bayesov probabilistički model učenika (AST, KBS) i epizodni model učenika (ACE, WINDS), dok se stereotipovi se koriste samo kod ASM metodologije.

U našem pristupu kombiniramo model prekrivanja, Bayesov probabilistički model učenika i stereotipove, čime osiguravamo modeliranje učenika s više aspekata. Složeniji pristupi modeliranju učenika omogućavaju i složenije prilagođavanje učeniku.

#### ***Računalom oblikovani nastavni sadržaj***

*Prva faza* u generiranju računalom oblikovanog nastavnog sadržaja je generiranje elemenata računalom oblikovanog nastavnog sadržaja, što može raditi učitelj ili se elementi mogu

---

generirati automatski. U većini sustava elemente računalom oblikovanog nastavnog sadržaja generira učitelj (GTE, ECSAI, AST, KBS, ATLAS, INSPIRE, WINDS, NetCoach, APeLS, ACCT, ADOPTA), kod nekih ima naznaka automatizma, ali s još velikim utjecajem učitelja (ELM-ART, ACE), dok je automatizacija prisutna u sustavima DCG, ASM, iClass, PAIGOS.

U našem pristupu nema nikakvog utjecaja učitelja na generiranje elemenata računalom oblikovanog nastavnog sadržaja. Elementi računalom oblikovanog nastavnog sadržaja se potpuno automatizirano generiraju na temelju ontologije područnog znanja. Generiraju se elementi različite složenosti, što osigurava kasniju realizaciju prilagodljivosti prema znanju.

*Druga faza* u generiranju računalom oblikovanog nastavnog sadržaja je odabir i nizanje elemenata računalom oblikovanog nastavnog sadržaja. Taj proces se može izvršiti samo jedan put prije nego što učenik započne s procesom učenja i poučavanja (statički) ili se može ponavljati više puta tijekom procesa učenja i poučavanja (dinamički). Upravo dinamika osigurava prilagodljivost računalom oblikovanog nastavnog sadržaja, pa većina spomenutih sustava ima upravo dinamički odabir i nizanje elemenata računalom oblikovanog nastavnog sadržaja (ECSAI, DCG, ELM-ART, AST, ACE, KBS, ATLAS, INSPIRE, NetCoach, APeLS, ACCT, ASM, ADOPTA). Statički odabir i nizanje elemenata računalom oblikovanog nastavnog sadržaja je prisutno u sustavima GTE, WINDS, iClass, dok PAIGOS ima statički odabir i nizanje s naznakama dinamike.

*Treća faza* u generiranju računalom oblikovanog nastavnog sadržaja se odnosi na prezentiranje elemenata računalom oblikovanog nastavnog sadržaja. Prezentiranje je faza u kojoj se također realizira prilagođavanje računalom oblikovanog nastavnog sadržaja. Najveći broj sustava realizira prilagođavanje prezentiranja elemenata računalom oblikovanog nastavnog sadržaja korištenjem tehnika prilagodljive hipermedije, kao što su prilagodljiva podrška navigaciji i prilagodljivo označavanje veza (ECSAI, ELM-ART, AST, ACE, KBS, INSPIRE, WINDS, NetCoach, ADOPTA).

Za veliki broj sustava se navodi da se prilagođavaju prema znanju učenika. Međutim, to prilagođavanje ne zadire u mijenjanje samog sadržaja elementa računalom oblikovanog nastavnog sadržaja, već se odnosi na odabir i nizanje elementa (ECSAI, DCG, ELM-ART, AST, ACE, KBS, INSPIRE, NetCoach, APeLS, ACCT, ASM, PAIGOS).

Prilagođavanje prezentiranja samog sadržaja elementa računalom oblikovanog nastavnog sadržaja prema znanju učenika nije ni u jednom sustavu napravljeno na način da se prezentiranje sadržaj elementa računalom oblikovanog nastavnog sadržaja mijenja prema razini znanja koju učenik ima. U našem pristupu razine znanja su definirane prema Bloomovoj taksonomiji znanja, čime smo uveli inovaciju u prilagođavanje računalom oblikovanog nastavnog sadržaja i odmakli se od prilagodljivosti koja je prisutna kod prilagodljive hipermedije i prilagođavanja prema stilu učenja. Prilagođavanje je realizirano uporabom predložaka za učenje.

---

### ***Značajka učenika prema kojoj se prilagođava***

Prilagođavanje prema stilu učenja (AST, ATLAS, INSPIRE, APeLS, ACCT, ASM, ADOPTA).

Prilagođavanje prema znanju (ECSAI, DCG, ELM-ART, AST, ACE, KBS, INSPIRE, NetCoach, APeLS, ACCT, ASM, PAIGOS).

Prilagođavanje prema ciljevima učenja (KBS, NetCoach, ACCT, iClass).

### ***Element prilagođavanja***

Prilagođavanje računalom oblikovanog nastavnog sadržaja se može realizirati odabirom i nizanjem elemenata računalom oblikovanog nastavnog sadržaja ili prilagođavanjem prezentiranja elemenata računalom oblikovanog nastavnog sadržaja. Kod realizacije prilagođavanja promatramo najmanju granulu računalom oblikovanog nastavnog sadržaja ili područnog znanja koja se može manipulirati. Što je manji element prilagođavanja, to je samo prilagođavanje „finije“ i „osjetljivije“, te u konačnici uspješnije.

Kod spomenutih sustava uglavnom je element prilagođavanja jedinica ili stranica ili lekcija ili zadatak koja uključuje više koncepata područnog znanja (GTE, ECSAI, ELM-ART, AST, ACE, KBS, ATLAS, INSPIRE, WINDS, NetCoach, APeLS, ACCT, iClass, ADOPTA). Samo je kod dva sustava element prilagođavanja koncept područnog znanja (DCG i ASM).

U našem pristupu najmanji element raspoloživ za prilagođavanje je koncept znanja, dakle, „atomska čestica“ znanja i u tom smislu nedjeljiva. Pošto smo se u novom modelu dotakli najmanjeg mogućeg elementa za prilagođavanje, smatramo da našim pristupom realiziramo maksimalnu prilagodljivost.

### ***Provjeravanje znanja***

Provjeravanje znanja učenika je „vrući krumpir“ u sustavima e-učenja. Naime, u većini sustava e-učenja učitelj je taj koji mora osmisliti pitanja za provjeravanje znanja učenika. Nedostaci su u tome što je to mukotrpan posao jer treba ručno unijeti pitanje u sustav, odrediti način ocjenjivanja, navesti točne i netočne odgovore, napisati povratnu informaciju, a onda sve to ponoviti što je moguće više puta jer nije dobro da se pitanja ponavljaju (učenici nauče odgovore napamet).

Dakle, kad promatramo provjeravanje znanja, zanima nas da li sustav sam generira pitanja (automatsko) ili pitanja unosi učitelj. Kod navedenih sustava ili uopće ne postoji provjeravanje znanja ili pitanja stvara učitelj (DCG, ELM-ART, AST, ACE, INSPIRE, WINDS, NetCoach, PAIGOS, ADOPTA).

U našem pristupu sustav sam generira pitanja u formi zadataka objektivnog tipa (dopunjavanje, jednostavno dosjećanje, jednostruki izbor, povezivanje, sređivanje, višestruki izbor) bez ikakvog uplitanja učitelja, sam pridružuje pitanja u test, te sam vrednuje znanje

učenika. Pitanja se generiraju na temelju koncepata i relacija iz ontologije područnog znanja, odnosno nad podskupom ontologije nad kojim se učenik učio i poučavao, uz pomoć predložaka za pitanja. Težina pitanja u testu također je usklađena s razinom znanja učenika (Bloomova taksonomija).

### ***Ulazni test***

U sustavima koji žele prilagoditi proces učenja i poučavanja učeniku, potrebno je što prije započeti s prilagođavanjem. Međutim, to nije moguće dok god se ne inicijalizira model učenika. Inicijalizacija se realizira nekim oblikom ulaznog testa ili upitnika (ECSAI, DCG, AST, ACE, ATLAS, APeLS, iClass).

Kao i kod provjeravanja znanja, i kod ulaznog testa ili upitnika problem je u generiranju samih pitanja, ali i u određivanju podskupa područnog znanja koji bi trebao predstavljati nužno predznanje učenika. Nijedan od navedenih sustava ne razmatra navedenu problematiku u svom pristupu.

Pošto smo već kazali da se u našem pristupu generiranje pitanja i testova realizira automatski, potrebno je još samo objasniti kako odrediti podskup područnog znanja nad kojim će se generirati ulazni test. Upravo u tom segmentu smo napravili najveći iskorak definirajući *reprezentaciju područnog znanja* temeljenu na matematičkom modelu kojim smo osigurali ispravnost navedenog pristupa. Sličan matematički pristup se spomije kod ASM metodologije, ali za različite svrhe.

Sažeti prikaz navedene usporedbe prikazan je u Tablica 3.3.



Tablica 3.3. Usporedna tablica

Rbr	Sustav	Generiranje elemenata (učitelj/ aut)	Odabir i nizanje (stat/ din)	Prilagođavanje	Generiranje pitanja (učitelj/ aut)	Ulazni test
1	GTE	učitelj	statičko	primjenjivost metoda poučavanja	nema	nema
2	ECSAI	učitelj	dinamičko	znanje i ponašanje učenika prilagodljiva prezentacija prilagodljiva podrška navigaciji	nema	upitnik
3	DCG	automatsko	dinamičko	znanje i kognitivni stil učenika	učitelj	test znanja
4	ELM-ART	učitelj elementi automatskog	dinamičko	znanje prilagodljivo označavanje (semafor)	učitelj	nema
5	AST	učitelj	dinamičko	znanje i stil učenja prilagodljiva podrška navigaciji (semafor), prilagodljivo nizanje i prilagodljivo testiranje	učitelj	upitnik i test znanja
6	ACE	učitelj elementi automatskog	dinamičko	znanje, interesi i preference učenika prilagodljivo označavanje inkrementalno povezivanje hiperveza	učitelj	upitnik i test znanja
7	KBS HYPERBOOK SYSTEM	učitelj	dinamičko	znanje i cilj učenja prilagodljiva podrška navigaciji, prilagodljivo označavanje	nema	nema
8	ATLAS	učitelj	dinamičko	profil i akcije učenika	nema	upitnik
9	INSPIRE	učitelj	dinamičko	znanje i stil učenja prilagodljiva podrška navigaciji prilagodljiva prezentacija	učitelj	nema
10	WINDS	učitelj	statičko	prilagodljiva podrška navigaciji prilagodljiva prezentacija	učitelj	nema
12	NETCOACH	učitelj	dinamičko	znanje i cilj učenja prilagodljivo označavanje	učitelj	nema
11	ApeLS	učitelj	dinamičko	znanje i stil učenja opisi i kandidatske grupe	nema	upitnik
13	ACCT	učitelj	dinamičko	znanje, cilj i stil učenja opisi i kandidatske grupe	nema	nema
14	ASM	automatsko	dinamičko	znanje i stil učenja	nema	nema
15	ICLASS	automatsko	statičko	cilj učenja, preference učenika i učitelja	nema	upitnik
16	PAIGOS	automatsko	statičko elementi dinamičkog	znanje, motivacija, razina sposobnosti	učitelj	nema
17	ADOPTA	učitelj	dinamičko	znanje i stil učenja prilagodljiva podrška navigaciji	učitelj	nema

Temelje za generiranje računalom oblikovanog nastavnog sadržaja postavila je Vassileva sa svojim sustavom DCG (Vassileva, 1992), dok je temelje za prilagodljivu navigaciju postavio Brusilovsky sustavom ELM-ART (Brusilovsky, Schwarz, & Weber, 1996). Slijedilo je razdoblje u realiziranju generiranja računalom oblikovanog nastavnog sadržaja za koje je karakteristična nerazvijenost pedagoškog znanja koje se koristilo u tim sustavima. Zatim su se pojavili sustavi PAIGOS i ADOPTA koji posjeduju značajnu količinu pedagoškog znanja, te mogu ostvariti različite ciljeve učenja.

Iz prethodne tablice vidimo da bi bilo idealno kada bi postojao sustav koji bi imao *automatsko* generiranje elemenata računalom oblikovanog nastavnog sadržaja, *dinamički* odabir i nizanje elemenata računalom oblikovanog nastavnog sadržaja, *automatsko* generiranje pitanja i ulazni test nad *reprezentativnim* podskupom područnog znanja, u kojem bi se prilagođavanje realiziralo prema *znanju* učenika, kao njegovoj najvažnijoj značajki. Upravo takav model, AC-ware Tutor, se opisuje ovom doktorskom disertacijom i on je uspostavljen, prototipno implementiran i testiran.

### **3.2 Modeliranje učenika u sustavima za generiranje prilagodljivog računalom oblikovanog nastavnog sadržaja**

Modeliranje učenika je beskorisna aktivnost ako nema primjenu. Model učenika se koristi kod prilagođavanja procesa učenja i poučavanja učenika u odabranom područnom znanju. Prilagođava se računalom oblikovani nastavni sadržaj, povratna informacija i generiranje pomoći, te veličina jedinice poučavanja. Da bi bio potpuno prilagodljiv, ITS mora pružiti optimalno korištenje informacija koje su sadržane u modelu učenika, da bi vodio svoje akcije prema specifičnim potrebama svakog učenika.

Ukratko ćemo opisati pristupe modeliranju učenika koje kombiniramo u realizaciji modela učenika u našem pristupu. Već smo kazali da ćemo kombinirati model prekrivanja, Bayesov probabilistički model učenika i stereotipove, jer složeniji pristupi modeliranju učenika omogućavaju i složenije prilagođavanje učeniku.

Ideja *modela prekrivanja* je u predstavljanju znanja učenika kao prekrivanja s područnim znanjem (Carr & Goldstein, 1977). Za svaki koncept područnog znanja, model prekrivanja sadržava određenu vrijednost koja predstavlja procjenu učenikova poznavanja određenog koncepta. Ta vrijednost može biti binarna vrijednost (zna – ne zna), kvalitativna mjera (dobro – srednje – loše) ili kvantitativna mjera, kao vjerojatnost da učenik zna koncept. Model prekrivanja se, stoga, može realizirati skupom parova koncept-vrijednost.

Modeli učenika koji se temelje na Bayesovim mrežama (zvat ćemo ih *Bayesovim modelima učenika*) ne koriste se u velikom broju inteligentnih tutorskih sustava, jer je predviđanje vjerojatnost svladavanja novog znanja na temelju učenikovih značajki, najsloženija tehnika modeliranja učenika.

*Stereotipni model učenika* omogućava razlikovanje nekoliko tipičnih stereotipova učenika ((Rich, 1983), (Beaumont, 1994)). Za svaku dimenziju modela učenika, sustav može imati skup mogućih stereotipova. Ovaj model učenika možemo također reprezentirati kao skup parova stereotip-vrijednost, gdje vrijednost može biti binarna (istina – laž) ili vjerojatnost da učenik pripada stereotipu. Stereotipni model učenika je jednostavniji od modela prekrivanja, ali ga je lakše primijeniti.

Moguće je kombinirati model prekrivanja i stereotipni model (sustavi Anatom-Tutor, EPIAIM, KN-AHS, Hypadapter) pridruživanjem skupa parova koncept-vrijednost svakom stereotipu ((Boyle & Encarnacion, 1994), (de Rosis, De Carolis, & Pizzutilo, 1993)) ili određivanjem težine koncepta (Hohl, Böcker, & Gunzenhäuser, 1996). U ovakvim kombinacijama pristupa, stereotipni model se koristi za inicijalizaciju modela prekrivanja.

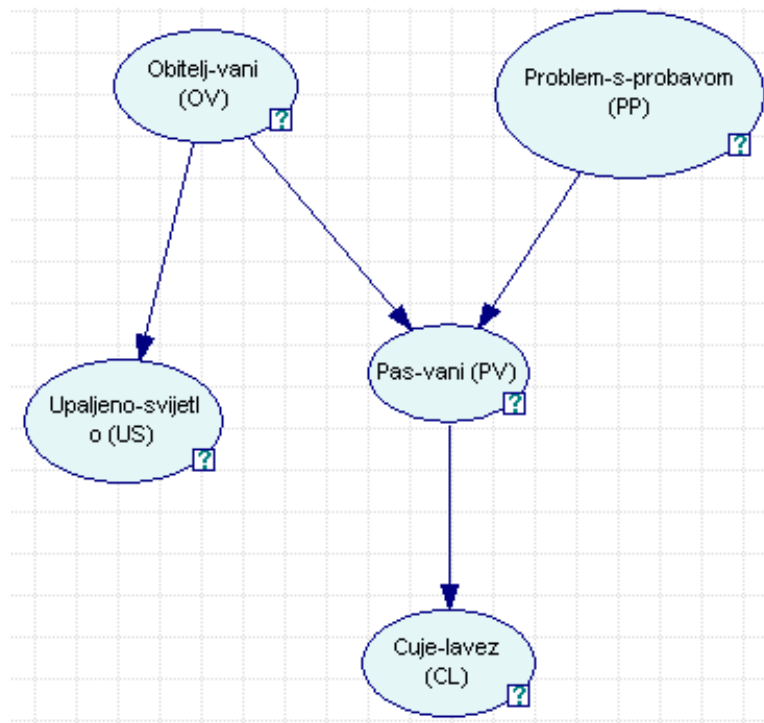
Za svaku tehniku modeliranja učenika, osim tehnike prekrivanja (jer se smatra uobičajenim pristupom za modeliranje učenika u ITS-ovima), navest ćemo po jedan do dva sustava koji ju koriste, a sve u svrhu boljeg razumijevanja kompleksnosti rješenja kojeg implementiramo u našem pristupu. Pošto neki od opisanih sustava nisu ITS-ovi, oni kao temeljnog sudionika imaju korisnika (eng. user), a ne učenika (eng. student), pa govore o modelu korisnika (eng. user model), umjesto o modelu učenika (eng. student model).

### 3.2.1 Primjeri sustava s Bayesovim modelom učenika

Charniak u svom radu iz 1991. godine opisuje jednostavan primjer koji pokazuje prednosti grafičkog modeliranja problema pomoću Bayesove mreže (Charniak, 1991). Autor opisuje problemsku domenu u kojoj on živi u kući zajedno sa ženom i imaju psa kao kućnog ljubimca. Iza kuće je dvorište, a ispred kuće je trijem na kojem se nalazi rasvjeta. Nakon posla autor dolazi kući i zanima ga kolika je vjerojatnost da se netko nalazi u kući. Budući da živi dugo u tom susjedstvu poznaje određene varijable koje mu mogu pomoći u dobivanju odgovora:

- svjetlo na trijemu
  - o žena ga često upali kad ode negdje
  - o žena ga upali i kad očekuje goste
- pas
  - o kad nema nikog kod kuće nalazi se u dvorištu
  - o psa vode u dvorište i kad ima probleme s probavom
  - o ako je pas iza kuće može se čuti kako laje, ali njegov lavež se može zamijeniti sa drugim psima koji se čuju u susjedstvu

U ovom problemu je moguće uočiti pet slučajnih varijabli: žena, pas, problem s probavom, svjetlo i lavež. Koristeći grafički prikaz pomoću Bayesove mreže, ovaj problem je moguće prikazati na sljedeći način (Slika 3.1):



Slika 3.1. Bayesova mreža

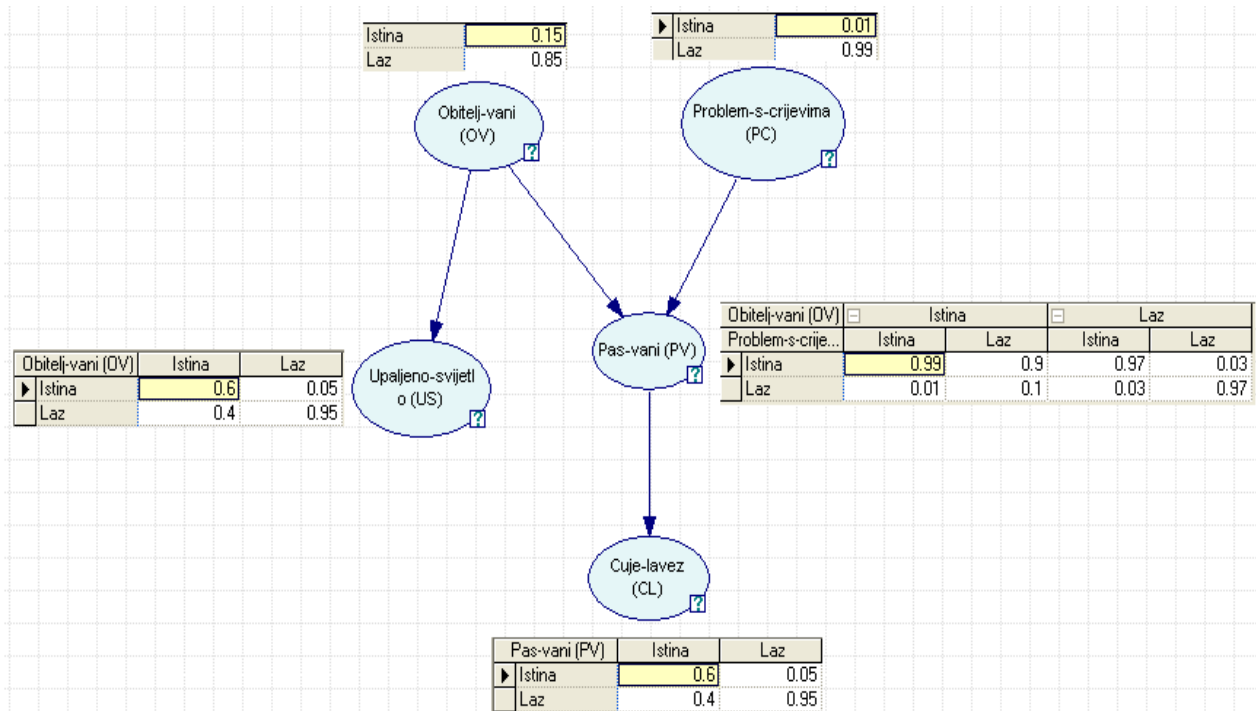
Iz mreže možemo zaključiti da ako žena nije u kući, tada je ili upaljeno svjetlo ili je pas u dvorištu. Pas je u dvorištu i ako ima problema s probavom. Ako je pas u dvorištu, onda se može čuti njegov lavež. Naravno, ovaj model je probabilistički, što znači da se može dogoditi i da žena nije kod kuće, a svjetlo nije upaljeno. Isto tako, pas može biti u dvorištu, a da ne laje.

Kako bi se ovaj model upotpunio, potrebno je svakom čvoru pridružiti vjerojatnosti:

- „a priori“ vjerojatnosti za čvorove bez roditelja
- uvjetne vjerojatnosti za sve čvorove s roditeljima i to za sve moguće kombinacije ishoda roditelja

Na Slika 3.2 je prikazana Bayesova mreža za prethodni primjer koja je upotpunjena s vjerojatnostima za slučajne varijable. Čvorovi *Obitelj vani* i *Problem s probavom* su roditelji u grafu, pa je njima pridružena „a priori“ vjerojatnost. Čvor *Pas vani* ima ukupno 4 vjerojatnosti jer ima 2 roditelja koji su logičke varijable (imaju dva stanja).

Između ostalog, iz spomenute slike jasno je da, ako članovi obitelji napuste kuću (OV), oni će ostaviti upaljeno svjetlo (US) vani 60 posto vremena, ali svjetlo će biti upaljeno (US) čak i kad ne napuste kuću ( $\overline{OV}$ ) 5 posto vremena (recimo, zato što nekoga očekuju).



Slika 3.2. Bayesova mreža upotpunjena vjerojatnostima

Bayesova mreža omogućava izračunavanje uvjetne vjerojatnosti čvorova u mreži ako su poznate vrijednosti nekih čvorova. Na primjeru od prije, ako imamo dokaz da je svjetlo upaljeno ( $US$ ), ali ne čujemo svoga psa ( $\overline{CL}$ ), možemo izračunati uvjetnu vjerojatnost da obitelj nije doma ( $\overline{OV}$ ) (0,5).

Od velike je važnosti, da se iz postavljenih vjerojatnosti mogu izračunati „a priori“ vjerojatnosti za sve varijable. Ako su roditelj i dijete dvije međusobno nezavisne slučajne varijable, onda je moguće lako definirati vjerojatnost djeteta na temelju vjerojatnosti njegovih roditelja. Problem se svodi na računanje „a priori“ vjerojatnosti za dijete na temelju podataka o roditeljima koji utječu na dijete. U dobro definiranoj Bayesovoj mreži moguće je za svaki čvor odrediti njegovu „a priori“ vjerojatnost, npr. može se odrediti „a priori“ vjerojatnost da je upaljeno svjetlo ( $US$ ) (Primjer 1):

**Primjer 1.**

$$P(\text{Upaljeno svjetlo}) = P(\text{Upaljeno svjetlo}|\text{Obitelj vani}) * P(\text{Obitelj vani}) \\ + P(\text{Upaljeno svjetlo}|\text{Obitelj nije vani}) * P(\text{Obitelj nije vani})$$

$$P(US) = P(US|OV) * P(OV) + P(US|\overline{OV}) * P(\overline{OV})$$

$$P(US) = 0,6 * 0,15 + 0,05 * 0,85 = 0,09 + 0,0425 = 0,1325$$

Ako je u problemu nešto poznato, očekivanja svih varijabli se mijenjaju, te se „a priori“ vjerojatnosti više ne očekuju jer postoje dokazi. Postavljanjem dokaza na mrežu može se

zaključiti kako to utječe na očekivanja ostalih varijabli. Može se provoditi zaključivanje kao što je:

- „kada vidim upaljeno svjetlo (US), koja je vjerojatnost da nema nikog kod kuće ( $\overline{OV}$ )“ (Primjer 2)
- „kada znam da je obitelj vani (OV) i znam da pas nema problema sa probavom ( $\overline{PP}$ ), koja je vjerojatnost da ću čuti lavež psa (CL)“ (Primjer 3)
- „kad čujem da pas laje (CL), koja je vjerojatnost da je obitelj vani (OV)“
- „kad vidim svjetlo (US) i čujem psa (CL), koja je vjerojatnost da je pas vani (PV)“

**Primjer 2.**

$$P(\text{Obitelj vani} | \text{Upaljeno svjetlo}) = \frac{P(\text{Upaljeno svjetlo} | \text{Obitelj vani}) * P(\text{Obitelj vani})}{P(\text{Upaljeno svjetlo})}$$

$$P(OV|US) = \frac{P(US|OV) * P(OV)}{P(US)} = \frac{0,6 * 0,15}{0,1325} = 0.6792$$

**Primjer 3.**

$$P(\text{Čuje lavež} | \text{Obitelj vani, Nema problema s probavom}) = \\ P(\text{Čuje lavež} | \text{Pas vani}) * P(\text{Pas vani} | \text{Obitelj vani, Nema problema s probavom}) \\ + P(\text{Čuje lavež} | \text{Pas nije vani}) * P(\text{Pas nije vani} | \text{Obitelj vani, Nema problema s probavom})$$

$$P(CL|OV, \overline{PP}) = P(CL|PV) * P(PV|OV, \overline{PP}) + P(CL|\overline{PV}) * P(\overline{PV}|OV, \overline{PP})$$

$$P(CL|OV, \overline{PP}) = 0,6 * 0,9 + 0,05 * 0,1 = 0,54 + 0,005 = 0,545$$

Moguće je postaviti i pitanje poput „kada vidim upaljeno svjetlo, kolika je vjerojatnost da će pas lajati“, samo što u ovom slučaju treba voditi računa o smislu određene kombinacije događaja - ono što je matematički moguće ne znači da vrijedi i u realnom svijetu. U Bayesovoj mreži ne postoje sve vjerojatnost za sve kombinacije događaja, pa je potrebno modelirati one najznačajnije događaje.

Naravno, zaključci u Bayesovoj mreži ne moraju uvijek biti apsolutni: žena može biti vani bez da upali svjetlo, pas može imati probleme s probavom i ostati u kući, ali zbog toga je ovaj model probabilistički.

Spojna vjerojatnost u Bayesovoj mreži je jedinstveno definirana, kao produkt individualnih distribucija za svaku pojedinačnu varijablu. Za mrežu iz primjera sa Slika 3.2, spojna vjerojatnost događaja “obitelj vani” (OV), “problem s probavom” (PP), “upaljeno svjetlo” (US), “pas vani” (PV) i “čuje lavež” (CL) je prikazana u sljedećem primjeru:

**Primjer 4.**

$$P(\text{Obitelj vani}, \text{Problemi s probavom}, \text{Upaljeno svjetlo}, \text{Pas vani}, \text{Čuje lavež}) = \\ P(\text{Obitelj vani}) * P(\text{Problemi s probavom}) * P(\text{Upaljeno svjetlo} | \text{Obitelj vani}) \\ * P(\text{Pas vani} | \text{Obitelj vani} \wedge \text{Problemi s probavom}) * P(\text{Čuje lavež} | \text{Pas vani})$$

$$P(OV, PP, US, PV, CL) = P(OV) * P(PP) * P(US | OV) * P(PV | OV \wedge PP) * P(CL | PV)$$

Kod Bayesovih modela učenika, mreža će se sastojati od stotine ili tisuće čvorova, i njihove uvjetne vjerojatnosti se mogu mijenjati mnogo puta, ovisno o novim informacijama. Na taj način se utječe na mijenjanje dokaza.

**3.2.1.1 ANDES**

Sustav Andes ((Conati et al., 1997), (VanLehn et al., 1998)) je inteligentni tutorski sustav za učenje i poučavanje osnova fizike. Andes pruža učenicima podršku pri rješavanju problema iz fizike, gdje zahtijeva prikaz cijelog postupka rješavanja, a ne samo konačan rezultat. Učenicima daje povratnu informaciju nakon svakog njihovog koraka. Kada učenik zatraži pomoć usred rješavanja problema sustav ga navodi na otkrivanje pogrešaka ili ga na vodi na sljedeći korak.

Bayesova mreža ekspertnog modela učenika u sustavu Andes se sastoji od dva dijela: statičkog i dinamičkog. Statički dio oblikuje učenikovu sposobnost primjene pravila u specifičnom kontekstu i općenito. Dinamički dio oblikuje činjenice, cilj, strategije i primjene pravila koja su važna za modeliranje koraka u procesu učenja i poučavanja.

Statički dio mreže se sastoji od čvorova *Pravila* (eng. Rule nodes) i čvorova *Pravila konteksta* (eng. Context-Rule nodes). Ove dvije vrste čvorova se koriste da bi se modeliralo što to znači za učenika poznavanje jedinice znanja. Učenik je usvojio neko pravilo ako ga može točno primijeniti kadgod je ono potrebno za rješavanje problema. Učeničeva usvojenost nekog pravila predstavlja se vjerojatnošću čvora *Pravila* koja se povećava za isti iznos kad god učenik ispravno upotrijebi to pravilo.

Čvorovi *Pravila* imaju binarne vrijednosti T i F, koje označavaju vjerojatnost da je pravilo usvojeno ili ne. Čvor *Pravilo* je istinit ako je učenik usvojio pravilo, tj. ako učenik uvijek koristi to pravilo kadgod je to prikladno. Za svaki čvor *Pravilo* postoji onoliko čvorova *Pravila konteksta* koliko ima konteksta definiranih za to pravilo. Čvorovi *Pravila konteksta* imaju binarne vrijednosti T i F, koje označavaju vjerojatnost da učenik zna primijeniti pravilo kod svakog problema u odgovarajućem kontekstu. Svaki čvor *Pravila konteksta* ima samo jednog roditelja i to čvor *Pravila* koje predstavlja odgovarajuće opće pravilo. Tablica uvjetne vjerojatnosti *Pravila konteksta*, ako je dan čvor *Pravila* je definirana sa:

1.  $P(\text{Pravilo konteksta} = T \mid \text{Pravilo} = T)$  je uvijek jednaka 1 jer po definiciji  $P(\text{Pravilo} = T)$  znači da učenik može primijeniti pravilo u svakom kontekstu.
2.  $P(\text{Pravilo konteksta} = T \mid \text{Pravilo} = F)$  predstavlja vjerojatnost da učenik može primijeniti opće pravilo u odgovarajućem kontekstu iako ga ne može primijeniti u svim kontekstima. Ova vjerojatnost implicitno definira razinu težine primjene pravila u kontekstu – što je lakši kontekst, viša je uvjetna vjerojatnost.

Dinamički dio mreže sadrži čvorove *Pravila konteksta* i četiri dodatna tipa čvorova: čvorove *Činjenica* (eng. Fact nodes), *Ciljeva* (eng. Goal nodes), *Primjene pravila* (eng. Rule-Application nodes) i *Strategije* (eng. Strategy nodes). Svi čvorovi se unose u mrežu iz grafa rješenja za određeni problem na početku rješavanja problema.

Čvorovi *Činjenica* i *Ciljeva* izgledaju isto s gledišta Bayesove mreže. Oni oba predstavljaju informaciju koja proizlazi iz rješavanja problema primjenjujući pravila iz područnog znanja. Razlika je u njihovom značenju za modul davanja pomoći: vjerojatnost čvorova *Ciljeva* se koristi za generiranje sugestija fokusiranih na kvalitativnu analizu problema i planiranje rješenja, dok se vjerojatnost čvorova *Činjenica* koristi da bi se pružile specifičnije sugestije vezane za aktualne korake u rješavanju problema. Čvorovi *Činjenica* i *Ciljeva* imaju binarne vrijednosti T i F koje označavaju da li učenik može izvesti odgovarajuće akcije ako posjeduje određeno znanje. Čvorovi *Činjenica* i *Ciljeva* (zajedno se nazivaju čvorovi *Propozicije* (eng. nodes)) imaju onoliko roditelja koliko je načina da ih se ostvari. Čvor *Propozicija* je istinit ako je propozicija u učenikovoju radnoj memoriji. Propozicije su ili činjenice specifične za problem ili ciljevi.

Čvorovi *Strategije* predstavljaju točke u kojima učenik može odabrati različite planove za rješavanje problema. Oni su jedini nebinarni čvorovi u mreži: imaju onoliko vrijednosti koliko je alternativnih strategija. Djeca čvorova *Strategije* su čvorovi *Primjene pravila* koji predstavljaju implementacije različitih strategija. Čvor *Primjena pravila* je istinit ako se primjena pravila dogodila. Vrijednosti čvorova *Strategije* su ekskluzivne strategije koje predstavljaju činjenicu da pri svakom koraku rješavanja problema, učenik prati točno jednu od predstavljenih strategija.

Čvorovi *Primjene pravila* spajaju čvorove *Pravila konteksta*, *Strategije*, *Činjenica* i *Ciljeva* s novim čvorovima *Propozicija*. Kao i čvorovi *Propozicija*, čvorovi *Primjene pravila* imaju binarne vrijednosti koje indiciraju da li su izvodivi ili ne. Čvor *Primjene pravila* je izvediv ako učenik primjenjuje odgovarajuće *Pravilo konteksta* na činjenice i ciljeve koji predstavljaju njegove preduvjete da bi se izvele činjenice i ciljevi koji predstavljaju njegov zaključak. Roditelji svakog čvora *Primjene pravila* uključuju točno jedno *Pravilo konteksta*, nekoliko čvorova *Propozicije* i moguće jedan čvor *Strategije*.

Čvorovi *Propozicija* i *Primjena pravila* imaju čvorove *Strategija* koji se koriste kada postoje dva načina rješavanja problema. Svaki čvor ima tablicu uvjetne vjerojatnosti koja mu je pridružena. Za čvorove bez roditelja, tablica predstavlja „a priori“ vjerojatnost tog čvora.



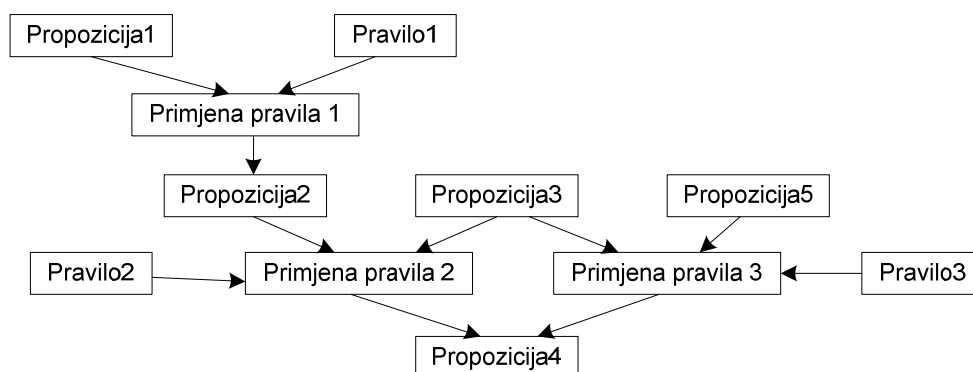
Npr., čvorovi *Pravila* nemaju roditelja pa se njihova tablica uvjetne vjerojatnosti sastoji od jedne vjerojatnosti koja označava vjerojatnost da je učenik usvojio to pravilo prije početka rješavanja problema. Ako propozicija nema roditelja, tada je ona činjenica ili cilj koji se spominje u tekstu problema.

Svi čvorovi *Primjene pravila* imaju tablicu uvjetne vjerojatnosti. One označavaju da se primjena pravila javlja samo ako su svi roditelji *Propozicije* u radnoj memoriji i *Pravilo* je usvojeno. Tablica 3.4. prikazuje tablicu uvjetne vjerojatnosti za pravilo koje ima dva roditelja propozicije za *Primjenu pravila 2* na Slika 3.3.

Ako je pravilo usvojeno (T u prvom stupcu) i obje propozicije su prisutne u radnoj memoriji (T u drugom i trećem stupcu), tada je vjerojatnost primjene pravila 0.999. Ako pravilo nije usvojeno (F u prvom stupcu) ili je neka od propozicija odsutna, tada je vjerojatnost primjene pravila nula. Postoji mala vjerojatnost 0.001 da iako su svi roditelji istiniti i pravilo je usvojeno, da se pravilo ipak neće primijeniti. Ova vjerojatnost se naziva *slučajna pogreška* (eng. unlucky slip), odnosno „slip“ parametar, jer predstavlja mogućnost da će učenik napraviti nenamjerni propust.

Tablica 3.4. Tablica uvjetne vjerojatnost čvora Primjena Pravila

Pravilo	Propozicija 2	Propozicija 3	P(pojaviti)	P(ne pojaviti)
T	T	T	0.999	0.001
T	T	F	0	1.0
T	F	T	0	1.0
T	F	F	0	1.0
F	T	T	0	1.0
F	T	F	0	1.0
F	F	T	0	1.0
F	F	F	0	1.0



Slika 3.3. Bayesova mreža sustava Andes

Tablica uvjetne vjerojatnosti pridružena propoziciji odražava pretpostavku da je propozicija došla u radnu memoriju ili kao produkt primjene pravila ili je učenik slučajno pogodilo. Tablica 3.5 pokazuje tablicu za propoziciju koja ima dva izvođenja i stoga dvije primjene pravila za roditelje (npr., *Propozicija 4* na Slika 3.3). Ako su se jedna od ili obje primjene

pravila dogodile, propozicija je definitivno u radnoj memoriji ( $P=1.0$ ). Ako se nijedna primjena pravila nije dogodila, postoji 0.2 šanse *slučajnog pogotka* (eng. lucky guess), odnosno „*guess*“ parametar. Različite propozicije imaju različite vrijednosti za svoje slučajne pogotke parametre, jer neke je propozicije lakše pogoditi. Trenutno, oko 74% čvorova ima vrijednost guess parametra 0.2, 25% 0.1 i 1% 0.01.

Tablica 3.5. Tablica uvjetne vjerojatnost čvora Propozicije

Primjena pravila 2	Primjena pravila 3	P(u radnoj memoriji)	P(nije u radnoj memoriji)
T	T	1.0	0.0
T	F	1.0	0.0
F	T	1.0	0.0
F	F	0.2	0.8

Glavna svrha mreže je da ustanovi koja je pravila učenik vjerojatno usvojio. Stoga, Andes za svakog učenika u njegovom modelu pamti za svako pravilo, vjerojatnost da je učenik usvojio to pravilo. Kada se generira problem i kada se stvori njegova Bayesova mreža, Andes inicijalizira „a priori“ vjerojatnosti čvorova *Pravila*.

Kako učenik rješava problem, mreža mijenja uvjetne vjerojatnosti pravila. Svaka učenikova akcija se uspoređuje s propozicijama u mreži. Točna akcija uvijek odgovara najmanje jednoj propoziciji, a često i većem broju njih. Netočne akcije obično ne odgovaraju nijednoj propoziciji. Propozicije koje su izjednačene označavaju se s True.

Kada je to mijenjanje gotovo, svi čvorovi imaju uvjetne vjerojatnosti koje odgovaraju učenikovim akcijama koje je do tada napravio. Kada se problem zatvori, uvjetne vjerojatnosti se pohranjuju u učenikov model. Ove nove vrijednosti predstavljaju vjerojatnost usvojenosti znanja, ako su dani svi dokazi koji su sakupljeni rješavajući dani problem.

### 3.2.1.2 CAPIT

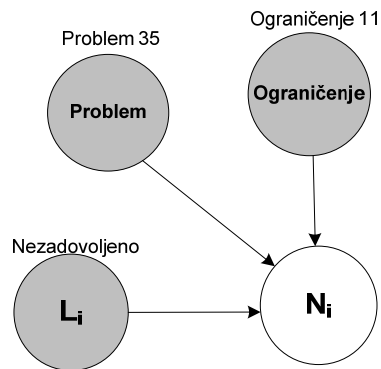
Capitalisation And Punctuation Intelligent Tutor (CAPIT) je normativni inteligentni tutorski sustav temeljen na ograničenjima (eng. constraint-based) koji uči i poučava djecu od 10 do 11 godina o znakovima interpunkcije u rečenicu, kao i određivanje koje se riječi u rečenici pišu velikim, a koje malim slovom ((Mayo, 2001), (Mayo & Mitrovic, 2001)). CAPIT za modeliranje učenika koristi modeliranje temeljeno na ograničenjima (eng. Constraint-Based Modelling - CBM) (Ohlsson, 1994).

Proces učenja i poučavanja u CAPIT-u uključuje prezentiranje problema učeniku navođenjem ciljeva i uputstava za rješavanje problema. Nakon što učenik završi s rješavanjem danog mu problema, predaje svoje rješenje dijagnostičkoj komponenti sustava. CAPIT tada utvrđuje koja su ograničenja (eng. constraints) važna za rješenje, te koja od njih su zadovoljena. Na temelju provedene dijagnostike, pedagoški modul provodi analizu i ukoliko je učenik

uspješno zadovoljio sva ograničenja, zadaje mu novi problem, a ukoliko postoje nezadovoljena ograničenja, onda učeniku prikaže poruku o grešci.

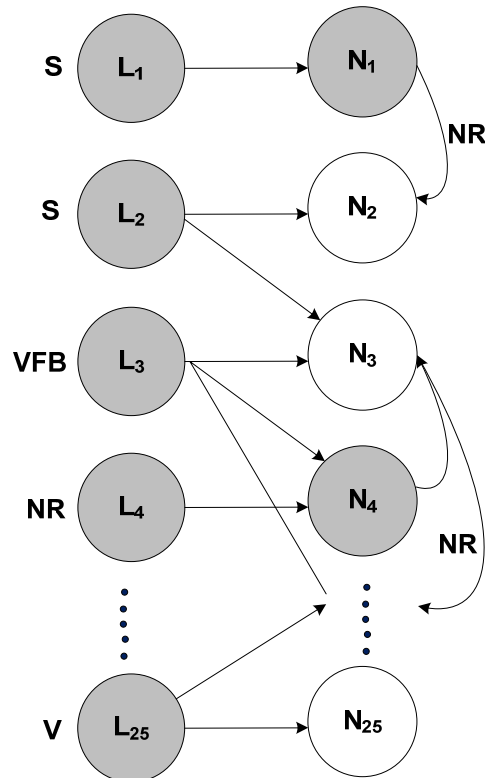
Modeliranje učenika u CAPIT-u se temelji na Bayesovoj mreži čiji je zadatak predvidjeti učenikovo zadovoljavanje ograničenja u problemu. Promatramo  $L_i$  rezultat posljednjeg pokušaja zadovoljenja ograničenja, te  $N_i$  koji odgovara rezultatu koji se predviđa kod sljedećeg pokušaja.  $L_i$  može poprimiti vrijednosti S (zadovoljen, eng. satisfied), V (nezadovoljen, eng. violeted), VFB (nezadovoljen s povratnom informacijom, eng. violated with feedback) ili NR (nevažan, eng. not relevant).  $N_i$  može poprimiti vrijednosti {S, V, NR}.

Na Slika 3.4 je prikazana „mala“ Bayesova mreža za predviđanje rezultata sljedećeg pokušaja zadovoljavanja  $i$ -tog ograničenja. U ovom primjeru, mreža predviđa se rezultat sljedećeg pokušaja zadovoljenja ograničenja 11 koje je važno za trenutni problem 35, a to ograničenje je prije bilo nezadovoljeno.



Slika 3.4. „Mala“ Bayesova mreža

Na Slika 3.5 je prikazana „velika „ Bayesova mreža. Učenik je zadovoljio ograničenja 1 i 2, nije zadovoljio ograničenja 3 i 25 (dobio je povratnu informaciju za 3), te još nije pokušao zadovoljiti ograničenje 4. Ova mreža može predvidjeti učenikove rezultate pri rješavanju problema čija su važna ograničenja 2, 3 i 25.



Slika 3.5. „Velika“ Bayesova mreža

### 3.2.2 Primjeri sustava sa stereotipnim modelom učenika

Stereotipovi predstavljaju kolekcije karakteristika ili aspekata (eng. facet). Učenik se može opisati skupom karakteristika koje imaju svoje vrijednosti. Da bi računalni sustav mogao učinkovito koristiti stereotipove mora poznavati same *stereotipove*, kao i *okidače* stereotipova. Okidač je objekat vezan za određenu situaciju. On sadrži *naziv stereotipa kojeg aktivira i procjenu istinitosti* (broj između 0 i 1000) koja se pridjeljuje stereotipu. Procjena istinitosti se odnosi na vjerojatnost da je stereotip prikladan za određenu situaciju.

*Stereotipni model učenika* se naziva *sinopsis* (eng. synopsis) i sastoji se od skupa stereotipova kojima određena osoba pripada. Vrlo je malo informacija u modelu korisnika za koje se može sa velikom sigurnošću kazati da su istinite, jer se ovakav model zasniva na vjerojatnosti (sustav MYCIN (Shortliffe et al., 1975)). Zbog toga model korisnika za svaku svoju informaciju mora imati i procjenu o njenoj istinitosti. Stoga se *stereotipovi* obično sastoje od *skupa trojki* (*atribut, vrijednost, procjena istinitosti* – eng. rating). Kod modela učenika situacija je još složenija. Naime, model mora pamtit i zašto o učeniku sustav vjeruje ono što vjeruje. Dakle, *model učenika* se sastoji od *skupa četvorki* (*atribut, vrijednost, procjena istinitosti, opravdanost*).

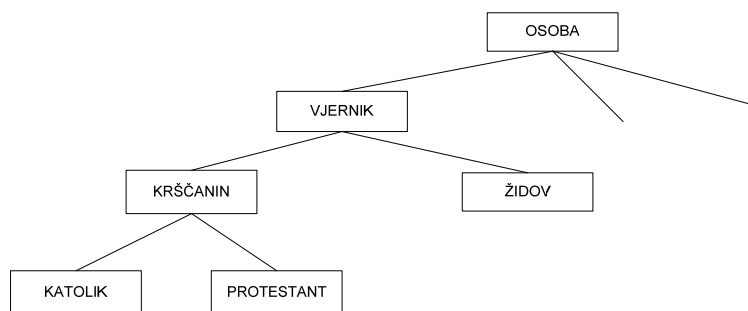
U slučaju stereotipova, procjena istinitosti predstavlja uvjerenost u činjenicu da će svaka osoba koja pripada određenom stereotipu imati određene karakteristike. U slučaju modela učenika, procjena istinitosti predstavlja uvjerenost sustava u svoja stajališta da je određena činjenica istinita za određenog učenika.

### 3.2.2.1 GRUNDY

Grundy (Rich, 1979) je sustav koji glumi knjižničara i predlaže knjige korisniku na temelju njegovih osobina. On koristi stereotipove korisnika da bi grupirao standardne karakteristike o korisnikovim osobinama. Na samom početku, Grundy pita korisnika da se opiše, pa na temelju toga određuje koji su stereotipovi prikladni za određenog korisnika. Pojedinačni izrazi *okidaju* (eng. trigger) stereotipove koji se mogu primijeniti na korisnika. Takav pristup samoopisivanjem funkcionira kod modeliranja osobnih karakteristika, ali se ne može primijeniti kod modeliranja područnog znanja. Korisnikovo samoopisivanje kao „početnika“ često se ne poklapa onim kako sustav vidi „početnika“. Također, okidači stereotipova se ne mogu koristiti za zaključivanje kao što je npr. zaključivanje da korisnik nešto ne zna ako pita o tome.

Stereotipovi su u GRUNDY-u organizirani u uređeni aciklički graf (eng. directed acyclic graph – DAG) s djelomično usmjeravajućom relacijom „generalizacija od“ (eng. partial ordering relation). Ovakav graf dozvoljava da jedan stereotip ima više od jedne generalizacije (Slika 3.6).

Najopćenitiji čvor ove strukture je stereotip „OSOBA“ koji unaprijed definira početne vrijednosti svih atributa s vrlo malim procjenama istinitosti. Iako je obično istina da atributi podstereotipova nasljeđuju vrijednosti od nadstereotipova, moguća je situacija u kojoj će doći do *nadvladavanja* (eng. override) vrijednosti atributa nadstereotipa s vrijednošću atributa podstereotipa.



Slika 3.6. Stereotipovi u sustavu Grundy (Rich, 1979)

Bilo koja aktivnost može aktivirati stereotipove. Stereotip se aktivira pojavljivanjem nekog od njegovih okidača. Jedan stereotip može imati više okidača. Procjene istinitosti stereotipa razlikuju se za različite situacije, odnosno okidače, koje ga aktiviraju, pa je korisno znati koja je situacija, odnosno okidač, aktivirala taj stereotip. Vrijednosti atributa kod GRUNDY-evih stereotipova su u rasponu od -5 do +5, a procjene istinitosti od 0 do 1000 (Slika 3.7).

Karakteristika	Vrijednost	Procjena istinitosti
Aktiviran od	Atletičar	
Nadstereotip	OSOBA	
Motivacija		
Uzbuđenje	3	600
Interes		
Sport	4	800
Uzбудljivost	5	700
Tolerancija nasilja	4	600
Romatičar	-5	500
Obrazovanje	-2	500
Tolerancija patnje	4	600
Snaga		
Fizička snaga	4	900
Ustrajnost	3	600

Slika 3.7. Stereotip SPORTAŠ u sustavu Grundy (Rich, 1979)

Svaki aktivni stereotip sadrži popis okidača koji su ga aktivirali. Ako se pojavi neki okidač, aktiviranje stereotipa će ovisiti o tome da li se taj okidač već prije pojavio. Ako jeste, ništa se neće dogoditi, a ako nije, moguće je sljedeće:

- stereotip nikad nije bio aktiviran – stereotip se aktivira
- stereotip je već bio aktiviran i još je aktivan - procjena istinitosti stereotipa se povećava, kao i svim njegovim okidačima, da bi odražavali potvrdu ispravnosti do koje je upravo došlo
- stereotip je već bio aktiviran, ali više nije aktivan – situacija se provjerava da bi se utvrdilo da li su dokazi dobiveni iz novih informacija za ili protiv stereotipa (možda je potrebno smanjiti procjene istinitosti stereotipa i njegovih okidača) - Slika 3.8

NE-TV-OKIDAČ		(Okidač se aktivira kada korisnik kaže da ne gleda TV)
KARAKTERISTIKA	VRIJEDNOST	
Stereotip	NE-TV-OSOBA	(osoba je vjerojatno obrazovana i ozbiljna)
Procjena istinitosti	800	(možda ova osoba nije stvarno osoba koja ne gleda TV, možda jednostavno sebi ne može priuštiti TV)

Slika 3.8. Okidači u sustavu Grundy (Rich, 1979)

Mijenjanje vrijednosti stereotipa:

$$\text{nova vrijednost stereotipa} = \frac{\text{stara vrijednost stereotipa} * \text{vrijednost stereotipa}}{\text{const} + 1} \quad (3.1)$$

gdje je *nova vrijednost stereotipa* dobivena kao težinski prosjek (eng. *weighted average*) *stare vrijednosti stereotipa* i vrijednosti koja proizlazi iz nove situacije.

Mijenjanje procjene istinitosti stereotipa:

$$\frac{\textit{nova procjena istinitosti stereotipa}}{\textit{istinitosti stereotipa}} = \frac{\textit{stara procjena istinitosti stereotipa}}{\textit{istinitosti stereotipa}} + \frac{\textit{procjena istinitosti stereotipa}}{\textit{stara procjena istinitosti stereotipa}} \quad (3.2)$$

gdje je *nova procjena istinitosti stereotipa* dobivena iz stare procjene istinitosti stereotipa i procjene istinitosti stereotipa koja proizlazi iz nove situacije.

Mijenjanje procjene vrijednosti okidača:

$$\frac{\textit{nova procjena istinitosti okidača}}{\textit{istinitosti okidača}} = \frac{\textit{stara procjena istinitosti okidača}}{\textit{istinitosti okidača}} + \frac{\textit{procjena istinitosti okidača}}{\textit{stara procjena istinitosti okidača}} \quad (3.3)$$

gdje je *nova procjena istinitosti okidača* dobivena stare procjene istinitosti okidača i procjene istinitosti okidača koja proizlazi iz nove situacije.

### 3.2.2.2 UNIX Consultant

UNIX Consultant (UC) ((Wilensky, Arens, & Chin, 1984), (Wilensky et al., 1988)) je računalni sustav koji uporabom prirodnog jezika savjetuje učenika o korištenju UNIX operacijskog sustava.

Da bi se bolje prilagodio velikom rasponu učenika, UC sadrži komponentu za eksplicitno modeliranje učenika koja se zove KNOME (KNOWledge Model of Expertise). KNOME sadrži informaciju o tome što učenik zna i zbog toga može:

- izbjeći kazivanje učeniku onoga što taj učenik već zna
- prilagoditi objašnjenja učenikovom nivou razumijevanja
- iskoristiti učenikovo predznanje kod interpretacije onoga što učenik kaže

U nedostatku već razvijenog modela, autor predlaže dvostruki stereotipni model – stereotip učenika i znanja. U KNOME-u je korišten pristup sličan ljudskom kategoriziranju, učenici su podijeljeni u 4 kategorije ili stereotipa koje odgovaraju različitim nivoima stručnosti (prema 4 faze razvoja znanja tijekom učenja (Kay & Black, 1985, prema (Chin, 1986)). Kategorije učenika su: novak (eng. novice), početnik (eng. beginner), osrednji (eng. intermediate) i stručnjak (eng. expert)

Svaka kategorija sadrži informacije o stanju znanja učenika koji pripada toj kategoriji. Konfliktne informacije o stanju znanja pojedinog učenika se eksplicitno opisuje i prelazi se preko nasljeđivanja od kategorije kojoj učenik pripada.

Osim kategoriziranja učenika, KNOME kategorizira informacije u težinske nivoe. Informacije se grupiraju prema njihovoj tipičnoj lokaciji na krivulji učenja (koja odgovara učenju prosječnog učenika), tj. na temelju činjenice kada će prosječni učenik naučiti tu informaciju. Težinske kategorije znanja su: jednostavno (eng. simple), srednje (eng. mundane), složeno (eng. complex) i ezoterično (eng. esoteric).

Ezoterični su oni koncepti koji ne spadaju u nijednu od prve tri kategorije i obično su to koncepti koji imaju neku posebnu svrhu. Ona uključuje informacije koje nisu tipične da ih učenik nauči u bilo kojem nivou stručnosti. Oni se nauče jedino kada učenici imaju posebne potrebe. Neki takvi pojmovi mogu biti poznati početnicima, ali nepoznati stručnjacima.

Ako je poznat nivo stručnosti učenika, onda se u model učenika pohranjuje samo ona informacija koja se razlikuje od referentne točke koju daje nivo stručnosti pohranjen u modelu učenika. Kada se prikupljaju informacije o tome što učenik zna, prvo se prikupljaju specifične informacije, a zatim se pristupa nasljeđivanju informacija iz stereotipa učenika.

Utvrđivanje nivoa učenika se radi uspoređivanjem svega što se zna o nekom učeniku sa stereotipovima učenika. Najbliži izbor, dobiven procesom eliminacije (npr. ako ne zna neku naredbu koju bi stručnjak morao znati, onda taj učenik ne može biti stručnjak), postaje učenikova kategorija. Pri eliminaciji se koriste sljedeća pravila:

- ako učenik ne zna činjenicu koju učenici nekog stereotipa trebaju znati, onda učenik ne može biti u toj kategoriji
- ako učenik ne zna činjenicu koja ima težinski nivo kao i one činjenice koje članovi tog stereotipa poznaju u najvećem broju, onda je nevjerovatno da je učenik član tog stereotipa
- ako učenik poznaje jednostavne činjenice, onda je nevjerovatno da je učenik novak
- ako učenik poznaje složene činjenice, onda učenik nije novak ili početnik, nevjerovatno je da je učenik srednji, te je vjerovatno učenik stručnjak

Na primjer, ako učenik ne zna neke činjenice srednje težine, KNOME zaključuje da:

- učenik ne može biti *stručnjak* jer stručnjaci znaju sve činjenice srednje težine,
- *donekle manje vjerovatno* da je učenik *osrednji* jer osrednji učenici znaju većinu činjenica srednje težine,
- *donekle više vjerovatno* da je učenik početnik jer početnici znaju nekoliko činjenica srednje težine,
- *više vjerovatno* da je učenik novak jer novaci ne znaju činjenice srednje težine

U samom početku, sustav smatra da je učenik vjerovatno *početnik*. Nakon toga svaki podatak o tome što učenik zna može eliminirati neke stereotipove kojima učenik može pripadati. Stereotipovi se eliminiraju ako podaci pokažu da učenik ne može biti član tog stereotipa ili ako je *vjerovatnost stereotipa* (razlika zbroja ocjena vjerovatno i zbroja ocjena nevjerovatno) padne ispod granice (proizvoljan broj, obično između 1 i 3). Kada se eliminiraju svi osim jednog stereotipa, onda učenik pripada upravo tom stereotipu. Ponekad se kategorija može odabrati i prije nego se ostale eliminiraju, ako vjerovatnost jedne od kategorija prelazi granicu (proizvoljan broj, obično između 2 i 4). U sustavu UC, jednom eliminirani, stereotipovi se više ne uzimaju u obzir (ne postoji mehanizam za ponovnu procjenu podataka i eventualnu promjenu kategorije).



KNOME je prvi sustav za modeliranje učenika s dvostrukim stereotipovima. Uporaba stereotipova omogućava KNOME-u predviđanje o tome što učenik vjerojatno zna ili ne zna na temelju parcijalnog modela učenika. Nakon što KNOME sakupi mali broj činjenica (oko 3) o tome što učenik zna ili ne zna, može zaključiti o učenikovom nivou stručnosti. Zatim na temelju učenikova nivoa stručnosti, KNOME može predvidjeti vjerojatnost o tome da li će učenik znati ili neće druge činjenice.

Pojedini učenik obično se u potpunosti ne uklapa u jedan stereotip. Stoga model učenika mora prikazati i kako se učenici razlikuju od ostalih stereotipova. KNOME pohranjuje specifične informacije o tome što učenik zna ili ne zna.

U svakom modelu učenika, zaključci koje model donosi o učeniku sadrže dozu *neizvjesnosti*. U KNOME-u se pokušalo izbjeći uporabu brojčanog prikazivanja neizvjesnosti jer oni vode k precjenjivanju točnosti rangiranja neizvjesnosti. Stoga, umjesto toga UC koristi fiksni broj jednostavnih nivoa (Zadeh, 1965, prema (Chin, 1986)): vjerojatno (eng. likely), nevjerojatno (eng. unlikely), vrlo vjerojatno (eng. very likely), vrlo nevjerojatno (eng. very unlikely), donekle vjerojatno (eng. somewhat likely), donekle nevjerojatno (eng. somewhat unlikely), istina (eng. tru), laž (eng. false), neodređeno (eng. uncertain).

KNOME koristi i *iskaze* (eng. predicates) kao što su (Zadeh, 1965, prema (Chin, 1986)): nekoliko (eng. afew), većina (eng. most), sve (eng. all), nijedan (eng. none).

Da bi zaključio da li učenik zna neku činjenicu, KNOME zaključuje u nekoliko koraka. Prvo, KNOME provjerava individualni model učenika. Ako individualni model učenika ima pohranjeno da ovaj učenik poznaje ili ne poznaje tu činjenicu, onda je odgovor *istina* ili *laž*. Drugo, ako nema specifične informacije u individualnom modelu učenika, KNOME provjerava kategoriju stereotipa učenika. Ako učenici koji pripadaju tom stereotipu znaju ili ne znaju tu činjenicu, onda je odgovor *istina* ili *laž*. Treće, ako ove provjere ne daju rezultat, KNOME pribjegava zaključivanju temeljenom na težinskom nivou činjenice. Ako učenikov stereotip zna sve činjenice tog težinskog nivoa, onda je odgovor *istina*; ako stereotip zna većinu činjenica te težine, onda je odgovor *vjerojatno*; ako stereotip zna nekoliko činjenica te težine, onda je odgovor *nevjerojatno*. Konačno, ako ovaj proces zaključivanja zakaže zbog nedostatka informacija, onda je odgovor *neodređeno*.

Kada procjena vjerojatnosti stereotipa postane istina, on postaje učenikova kategorija. Učenikova kategorija se može deducirati eliminacijom. Stereotipovi se eliminiraju kada njihova vjerojatnost postane laž. Kada se eliminiraju svi osim jednog, onda on postane učenikov stereotip. Prije nego što utvrdi učenikov stereotip, KNOME pretpostavlja da učenik pripada onom stereotipu s najvišom procjenom vjerojatnosti. U slučaju da postoje dva stereotipa s istom procjenom vjerojatnosti, KNOME odabire onaj s nižim nivoom stručnosti.

▪ ■ ▪ ■ ▪ ■ ▪ ■ ▪ ■ ▪ ■ ▪ ■ ▪ ■ ▪ ■ ▪ ■ ▪ ■ ▪ ■ ▪ ■ ▪ ■ ▪ ■ ▪

U ovom poglavlju, u svrhu utvrđivanja terminologije koja se primjenjuje u ovom istraživanju, opisan je pojam prilagođavanja i koje su najčešće korištene izvedenice te riječi. Objasnili smo pojam prilagodljive nastave i prilagodljivih sustava, kao i primjenu prilagodljive nastave u okvirima računalnih sustava. Težište ovog dijela doktorske disertacije je u objašnjavanju pristupa generiranju računalom oblikovanog nastavnog sadržaja, kao i opisivanja sustava koji implementiraju određeni pristup prilagodljivom generiranju računalom oblikovanog nastavnog sadržaja. Stoga smo istražili koji se drugi pristupi koriste u prilagodljivim sustavima za generiranje računalom oblikovanog nastavnog sadržaja i usporedili s njima naš pristup koji je opisan u prethodnom poglavlju. Usporedba je napravljena na temelju sljedećih parametara: područno znanje, model učenika, računalom oblikovani nastavni sadržaj, značajka učenika prema kojoj se prilagođava, element prilagođavanja, provjeravanje znanja, ulazni test. Pored navedenog, istražili smo kako se stereotipovi i Bayesove mreže koriste za modeliranje učenika kod najvažnijih predstavnika sustava koji ih implementiraju. Osim opisa realizacije navedenih pristupa modeliranju učenika, dan je i kratak osvrt na teorijske postavke.

U sljedećem poglavlju, na temelju spoznaja o realizaciji prilagođavanja u sustavima e-učenja, posebice u inteligentnim tutorskim sustavima, pristupamo detaljnoj razradi i oblikovanju modela sustava za automatsko i dinamičko generiranje prilagodljivog računalom oblikovanog nastavnog sadržaja.

## 4 Model sustava AC-ware Tutor

„Razuman se čovjek prilagođava svijetu oko sebe.  
Nerazuman uporno nastoji svijet prilagoditi sebi. Stoga sav  
napredak čovječanstva ovisi o nerazumnim ljudima.“  
poslovica

Model sustava za automatsko i dinamičko generiranje računalom oblikovanog nastavnog sadržaja, čija je zamisao opisana u drugom poglavlju, detaljno je prikazan u ovom poglavlju. Još jednom napominjemo važnost ovakvog pristupa, jer se prilagođavanje u sustavima e-učenja, u većini slučajeva, realizira kroz prizmu prilagođavanja sučelja prema različitim stilovima učenja, dok mi promatramo prilagođavanje sadržaja znanju učenika.

### 4.1 Područno znanje i reprezentacija područnog znanja

Osnovne vrste znanja kojim se opisuje određeno područno znanje su proceduralno i deklarativno znanje. Za formalizaciju proceduralnog znanja koriste se produkcijska pravila, dok se za deklarativno znanje koristi formalni jezik simbola ili neka druga shema prikazivanja znanja. Ontologijom se opisuje konceptualni model nekog područja, odnosno njegovo deklarativno znanje. Dakle, svako područje ljudskog djelovanja može se prikazati skupom pravilno povezanih koncepata koji odgovara područnom znanju.

Ontologija se u literaturi definira kao formalna i eksplicitna specifikacija dijeljene konceptualizacije (Gruber, 1993). Konceptualizacija je pojednostavljeno i apstraktno viđenje vanjskog svijeta, a osnovni strukturni elementi konceptualnog modela su koncepti i relacije koji čine formalni prikaz znanja (Genesereth & Nilsson, 1987).

#### 4.1.1 Struktura područnog znanja

Ontološki pristup u opisivanju područnog znanja omogućava jednostavnu formalizaciju deklarativnog znanja korištenjem različitih alata koji podržavaju rad s konceptima i relacijama. Na temelju skupa koncepata  $E_{KCP} = \{K_1, \dots, K_n\}$ ,  $n \geq 0$ , skupa relacija  $E_{REL} = \{r_1, \dots, r_m\} \cup \{isa, inst, part, slot, filler\}$ ,  $m \geq 0$  i  $\emptyset_E$  praznog elementa područnog znanja u drugom poglavlju smo definirali što je to područno znanje (Definicija 2.1),

Relacije slot i filler omogućavaju atributno-vrijednosni prikaz znanja definiranjem okvira (eng. frame) koncepta ((Touretzky, 1992), (Eugene Charniak & McDermott, 1985)). Koncept može imati proizvoljni broj atributa iskazanih konceptima koji su s njim povezani relacijom *slot*. Vrijednosti atributa su iskazane konceptima koji su s atributima i konceptima povezane relacijom *filler*. Na taj način jednoznačno određujemo parove *koncept-atribut*, *atribut-vrijednost*, *koncept-vrijednost*.

Opravdanost korištenja ovakvog pristupa prema ostalim, najbolje se vidi u sljedećem primjeru:

#### Primjer 5.

$$E_{KCP} = \{K_1, K_2, K_3, K_4, K_5\}$$

$$E_{REL} = \{isa, inst, part, slot, filler\}$$

#### Mogućnost 1 – ulančavanje

Koncept može imati proizvoljni broj atributa iskazanih konceptima koji su s njim povezani relacijom *slot*. Vrijednosti atributa su iskazane konceptima koji su s atributima povezane relacijom *filler*.

$PZ = \{(K_1, slot, K_2), (K_1, slot, K_4), (K_2, filler, K_3), (K_4, filler, K_5)\}$  jednoznačno opisuje okvir:

$K_1$
$K_2$   $K_3$
$K_4$   $K_5$

Problem nastaje kad isti atribut, ali s različitim vrijednostima, imaju različiti koncepti, jer ovakav zapis ne rezultira jednoznačnim interpretiranjem okvira:

$PZ = \{(K_1, slot, K_2), (K_2, filler, K_3), (K_4, slot, K_2), (K_2, filler, K_5)\}$

$K_1$
$K_2$   $K_3$

$K_4$
$K_2$   $K_5$

ili

$K_1$
$K_2$   $K_5$

$K_4$
$K_2$   $K_3$

Navedeni problem pokušavamo riješiti drugačijim pristupom.

#### Mogućnost 2 – uparivanje

Koncept može imati proizvoljni broj atributa iskazanih konceptima koji su s njim povezani relacijom *slot*. Vrijednosti atributa su iskazane konceptima koji su s konceptom povezane relacijom *filler*.

$PZ = \{(K_1, slot, K_2), (K_1, filler, K_3), (K_4, slot, K_2), (K_4, filler, K_5)\}$  jednoznačno opisuje okvire:

$K_1$
$K_2$   $K_3$

$K_4$
$K_2$   $K_5$

Problem nastaje kad jedan koncept ima više različitih atributa kojima pripadaju različite vrijednostima, jer ovakav zapis ne rezultira jednoznačnim interpretiranjem okvira:

$$PZ = \{(K_1, slot, K_2), (K_1, filler, K_3), (K_1, slot, K_4), (K_1, filler, K_5)\}$$

K <sub>1</sub>	
K <sub>2</sub>	K <sub>3</sub>
K <sub>4</sub>	K <sub>5</sub>

ili

K <sub>1</sub>	
K <sub>2</sub>	K <sub>5</sub>
K <sub>4</sub>	K <sub>3</sub>

Oba problema mogu se riješiti trećim pristupom.

**Mogućnost 3 – trokut**

Koncept može imati proizvoljni broj atributa iskazanih konceptima koji su s njim povezani relacijom *slot*. Vrijednosti atributa su iskazane konceptima koji su s atributima i konceptima povezane relacijom *filler*.

$$PZ = \{(K_1, slot, K_2), (K_1, filler, K_3), (K_2, filler, K_3), (K_1, slot, K_4), (K_1, filler, K_5), (K_4, filler, K_5)\}$$

jednoznačno određuje okvir:

K <sub>1</sub>	
K <sub>2</sub>	K <sub>3</sub>
K <sub>4</sub>	K <sub>5</sub>

$$PZ = \{(K_1, slot, K_2), (K_1, filler, K_3), (K_2, filler, K_3), (K_4, slot, K_2), (K_4, filler, K_5), (K_2, filler, K_5)\}$$

jednoznačno određuje okvire:

K <sub>1</sub>	
K <sub>2</sub>	K <sub>3</sub>

K <sub>4</sub>	
K <sub>2</sub>	K <sub>5</sub>

U Primjeru 6. je prikazan zapis proizvoljnog područnog znanja.

<b>Primjer 6.</b>
$E_{KCP} = \{K_1, K_2, K_3, K_4, K_5, K_6, K_7, K_8, K_9, K_{10}, K_{11}, K_{12}, K_{13}, K_{14}, K_{15}, K_{16}, K_{17}, K_{18}, K_{19}\}$ $E_{REL} = \{r_1, r_2, r_3, r_4, isa, inst, part, slot, filler, \emptyset_E\}$ $PZ = \left\{ \begin{array}{l} (K_1, part, K_2), (K_1, part, K_3), (K_2, isa, K_4), (K_4, inst, K_5), (K_4, part, K_6), (K_4, r_1, K_7), \\ (K_7, r_3, K_8), (K_8, inst, K_{14}), (K_8, inst, K_{15}), (K_8, inst, K_{16}), (K_9, part, K_8), (K_9, r_4, K_{10}), \\ (K_{11}, r_2, K_{12}), (K_{11}, inst, K_{13}), (K_{17}, \emptyset_E, K_{17}), (K_{13}, slot, K_{18}), (K_{18}, filler, K_{19}), (K_{13}, filler, K_{19}) \end{array} \right\}$

Pojmove nadkoncept i podkoncept, kao i oznake  $nK_x$  i  $pK_x$  smo već definirali u drugom poglavlju (Definicija 2.3, Definicija 2.4). Broj  $pK_x$  se naziva i *stupanj vrha*. U Primjer 7. su prikazani skupovi nadkoncepta i podkoncepta od jednog proizvoljnog koncepta područnog znanja.

<b>Primjer 7.</b>
$NadK_4 = \{K_2\}, nK_4 = 1$ $PodK_4 = \{K_5, K_6, K_7\}, pK_4 = 3$

Sljedeće definicije uvodimo zbog kasnijeg jednostavnijeg označavanja skupova koncepata i relacija koje one opisuju.

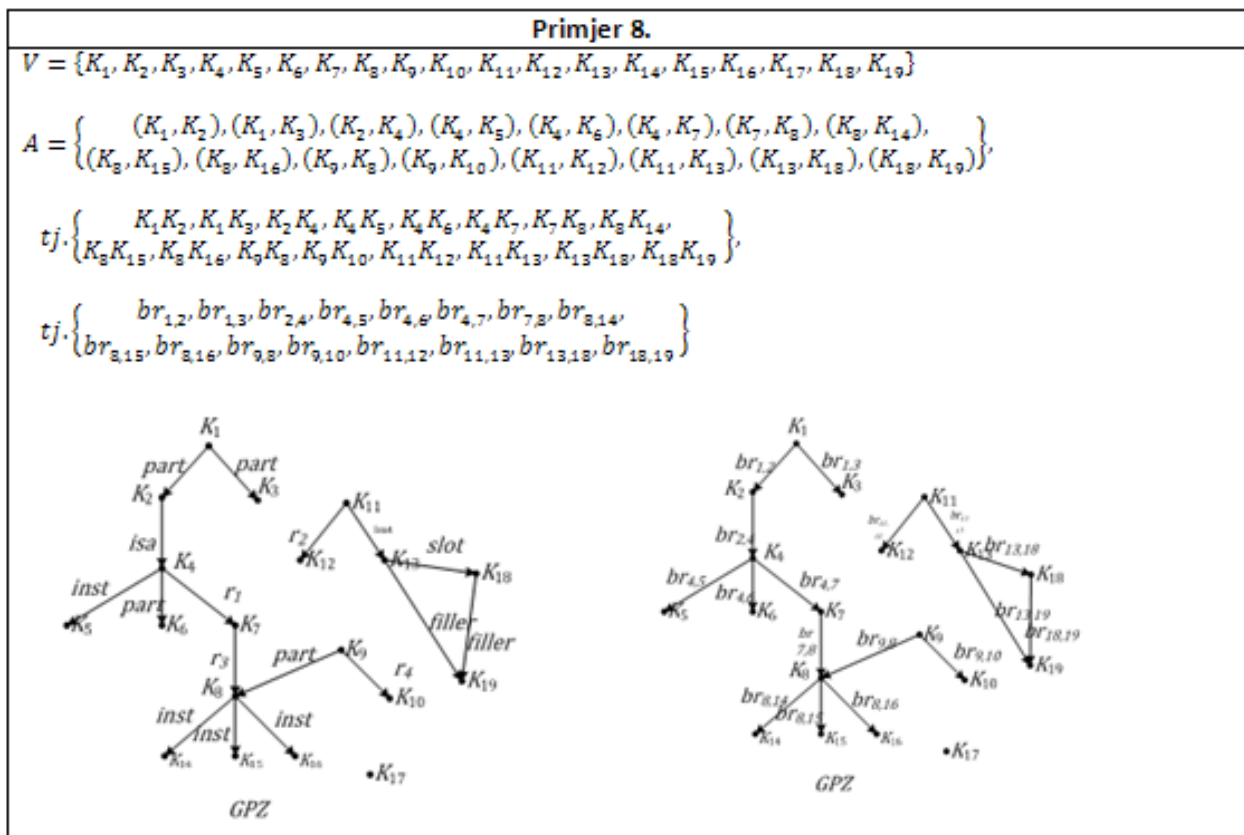
**Definicija 4.1** Definiramo sljedeće međusobno disjunktne skupove koncepata:

- skup svih **koncepata korijena**  $E_K = \{K_x \in E_{KCP} | NadK_x = \emptyset, PodK_x \neq \emptyset\}$
- skup svih **koncepata račvi**  $E_R = \{K_x \in E_{KCP} | NadK_x \neq \emptyset, PodK_x \neq \emptyset\}$
- skup svih **koncepata listova**  $E_L = \{K_x \in E_{KCP} | NadK_x \neq \emptyset, PodK_x = \emptyset\}$
- skup svih **izoliranih koncepata**  $E_I = \{K_x \in E_{KCP} | NadK_x = \emptyset, PodK_x = \emptyset\}$
- skup svih **atributa koncepata**  $E_A = \{K \in E_{KCP} | \exists K_x \in E_{KCP}, (K_x, slot, K) \in PZ\}$
- skup svih **vrijednosti koncepata**  $E_V = \{K \in E_{KCP} | \exists K_x \in E_{KCP}, (K_x, filler, K) \in PZ\}$

### 4.1.2 Graf područnog znanja

Pošto su osnovni elementi trojki područnog znanja koncepti i relacije među njima, koristimo teoriju grafova kao matematičku podlogu koja omogućava upravljanje podskupovima i elementima područnog znanja, kao i vizualizaciju područnog znanja ((Gross & Yellen, 1998), (Veljan, 1989)). Zato smo u drugom poglavlju definirali graf područnog znanja (Definicija 2.2).

Područno znanje iz Primjera 6. ima sljedeći usmjereni graf područnog znanja:



Pošto promatramo usmjerene grafove, možemo promatrati bridove čije strelice idu prema nekom konceptu (podkoncepti) i bridove čije strelice idu od nekog koncepta (nadkoncepti). Navodimo nekoliko definicija kojima pobliže određujemo graf područnog znanja.

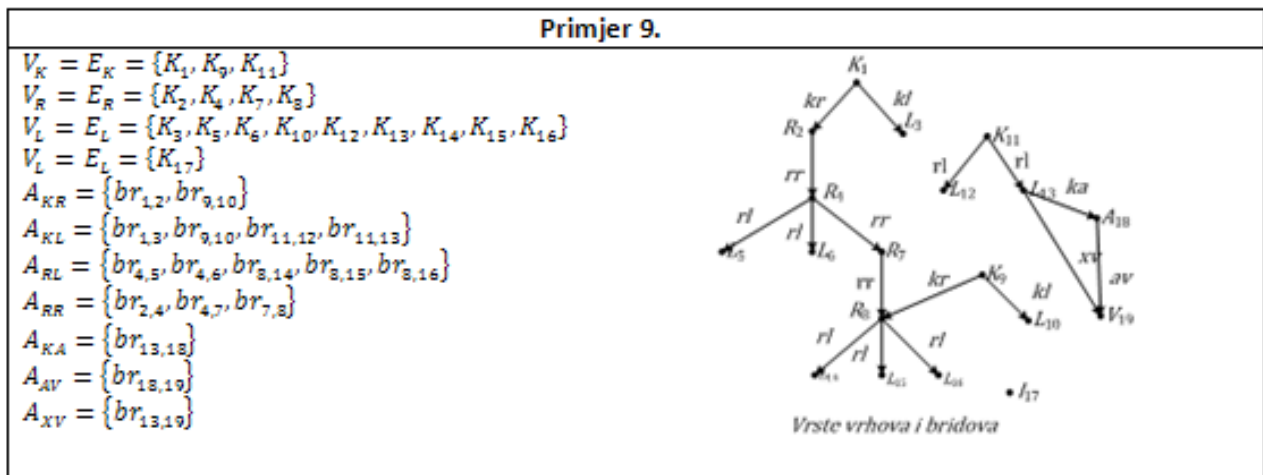
**Definicija 4.2** Definiramo sljedeće međusobno disjunktne skupove vrhova:

- skup svih **vrhova korijena**  $V_K = E_K$
- skup svih **vrhova račvi**  $V_R = E_R$  - unutrašnji vrhovi
- skup svih **vrhova listova**  $V_L = E_L$
- skup svih **izoliranih vrhova**  $V_I = E_I$
- skup svih **vrhova atributa**  $V_A = E_A$
- skup svih **vrhova vrijednosti**  $V_V = E_V$

**Definicija 4.3** Definiramo sljedeće međusobno disjunktne skupove bridova:

- skup svih **bridova između korijena i račve**  $A_{KR} = \{(K_x, K_y) \in A | K_x \in V_K, K_y \in V_R\}$
- skup svih **bridova između korijena i lista**  $A_{KL} = \{(K_x, K_y) \in A | K_x \in V_K, K_y \in V_L\}$
- skup svih **bridova između račve i lista**  $A_{RL} = \{(K_x, K_y) \in A | K_x \in V_R, K_y \in V_L\}$
- skup svih **bridova između račve i račve**  $A_{RR} = \{(K_x, K_y) \in A | K_x, K_y \in V_R\}$
- skup svih **bridova između koncepta i atributa**  $A_{KA} = \{(K_x, K_y) \in A | K_x \in V \setminus V_A \cup V_V, K_y \in V_A\}$
- skup svih **bridova između atributa i vrijednosti**  $A_{AV} = \{(K_x, K_y) \in A | K_x \in V_A, K_y \in V_V\}$
- skup svih **bridova između koncepta i vrijednosti**  $A_{XV} = \{(K_x, K_y) \in A | K_x \in V \setminus V_A, K_y \in V_V\}$

Pokažimo navedene skupove na sljedećem primjeru.



Usmjerenost bridova u grafu područnog znanja omogućava definiranje relacija „neposredni prethodnik“ i „neposredni sljedbenik“ samo na temelju strelice na bridu koji povezuje dva vrha.

**Definicija 4.4** Na skupu V definiramo relacije:

- $K_y$  je **neposredni sljedbenik** od  $K_x \Leftrightarrow K_x$  je **neposredni prethodnik** od  $K_y \Leftrightarrow \exists (K_x, K_y) \in A, K_x \in E_{KCP} \setminus E_A, K_y \in E_{KCP} \setminus E_V$

Jedan koncept može biti neposredni sljedbenik ili neposredni prethodnik drugog koncepta jedino ako nije atribut ili vrijednost atributa. Neposredni sljedbenik ili neposredni prethodnik ne može biti atribut ili vrijednost.

Ove relacije nam omogućavaju definiranje posebne vrste ulančanog podgraфа, tzv. šetnje:

**Definicija 4.5 Šetnja** od  $K_1$  do  $K_p$  u grafu GPZ je podgraf  $SetnjaK_1K_p = (V' = \{K_1, \dots, K_p\}, A')$  gdje je  $K_{i+1}$  neposredni sljedbenik od  $K_i, \forall i = 1, \dots, p - 1$   
(analogno,  $K_i$  neposredni prethodnik od  $K_{i+1}, \forall i = 1, \dots, p - 1$ )

U šetnji nema konceptata koji su atributi ili vrijednosti nekog koncepta, kao ni bridova prema atributima ili vrijednostima. U šetnji svaki vrh osim  $K_p$  ima neposrednog sljedbenika, odnosno svaki vrh osim  $K_1$  ima neposrednog prethodnika. Šetnju kraće označavamo sa  $(K_1, br_1, K_2, \dots, br_p, K_p)$ , gdje je  $br_i$  brid  $(K_{i-1}, K_i)$ , za  $i = 2, \dots, n$ . Duljina šetnje  $l_{K_xK_y}$  je broj bridova u podgrafu koji je definira, tj.  $|A'|$ . Šetnja je zatvorena ako vrijedi  $K_1 = K_p$ .

Put je šetnja u kojoj su svi vrhovi različiti, tj.  $PutK_1K_p = (V' = \{K_1, \dots, K_n\}, A')$  gdje je  $K_{i+1}$  neposredni sljedbenik od  $K_i, \forall i = 1, \dots, p - 1, \forall K_x, K_y \in V', K_x \neq K_y$ . Zatvoreni put zovemo ciklus.

Staza je šetnja u kojoj su svi bridovi međusobno različiti. Zatvorena staza zove se tura. Staza je Eulerova ako se u njoj pojavljuju svi bridovi u grafu, i to točno jedanput. Svaki put je staza.

Udaljenost između vrhova  $K_x$  i  $K_y$   $d(K_x, K_y)$  je duljina najkraćeg puta među njima.

Pošto stalno nastojimo održati izomorfizam između područnog znanja i graфа područnog znanja, potrebno je vidjeti kakvom podskupu područnog znanja odgovora podgraf kojeg čini šetnja između dva vrha.

**Definicija 4.6 Povezani niz konceptata** koji započinje s konceptom  $K_x$ , a završava s konceptom  $K_y$   
 $PNZK_xK_y = \left\{ (K_i, r, K_j) \in PZ \mid \begin{array}{l} K_x = K_1, K_y = K_p, r \neq \text{slot, filler, } \emptyset_E, K_i \neq K_j, \\ K_{i+1} \text{ neposredni sljedbenik od } K_i, \forall i = 1, \dots, p - 1 \end{array} \right\}$  je  
podskup područnog znanja PZ u kojem je podkoncept jednog koncepta ujedno nadkoncept drugog.

Usmjereni graf od  $PNZK_xK_y$  je upravo  $SetnjaK_xK_y$ . Broj elemenata u  $PNZK_xK_y$  je jednak broju bridova u  $SetnjaK_xK_y$ .

<b>Primjer 10.</b>
<p>Šetnja <math>(K_1, br_{1,2}, K_2, br_{2,4}, K_4, br_{4,6}, K_6)</math> je podgraf <math>SetnjaK_1K_6 = (\{K_1, K_2, K_4, K_6\}, \{br_{1,2}, br_{2,4}, br_{4,6}\})</math> duljine <math>l_{K_1K_6} =  \{br_{1,2}, br_{2,4}, br_{4,6}\} =3</math>  <math>PNZK_1K_6 = \{(K_1, part, K_2), (K_2, isa, K_4), (K_4, part, K_6)\}</math></p>



### 4.1.2.1 Komponente i cjeline grafa područnog znanja

U grafu područnog znanja mogu postojati nezavisni podgrafovi čiji vrhovi nisu povezani. Zato definiramo relaciju ekvivalencije koja omogućava definiranje takvih podgrafova.

**Definicija 4.7** Na skupu  $V$  definiramo relaciju ekvivalencije  $\equiv$  sa:

$$K_x \equiv K_y \Leftrightarrow \exists Put K_x K_y$$

Ova relacija ekvivalencije definira jednu particiju skupa  $V$ , pa su komponente grafa GPZ podgrafovi inducirani klasama ekvivalencije. Kažemo da je graf *povezan* ako postoji samo jedna komponenta. Komponente grafa su povezani podgrafovi. Pošto je put jedna vrsta šetnje, a u šetnji nema koncepata koji su atributi ili vrijednosti nekog koncepta, zaključujemo da u komponentama nema koncepata koji su atributi ili vrijednosti nekog koncepta.

Algoritam za rastavljanje grafa na komponente (Algoritam 1):

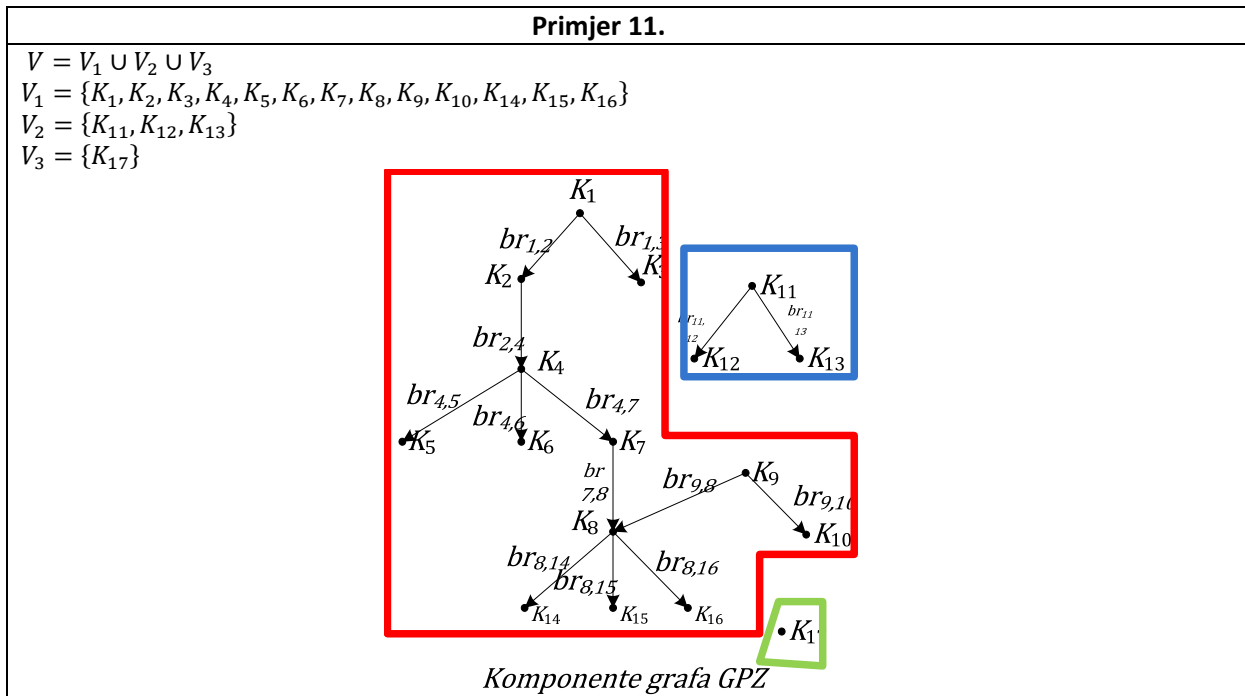
1. Definira se prazni skup  $SPG$  koji će sadržavati sve podgrafove usmjerenog grafa  $GPZ = (V, A)$ .
2. Zatim se za svaki usmjereni brid  $(K_x, K_y) \in A$  i za svaki podgraf  $G_i \in SPG$  provjerava da li skup vrhova  $V_i$  podgraфа  $G_i$  sadrži  $K_x$  ili  $K_y$ .
3. Ako sadrži, onda se podgraf  $G_i$  nadopunjuje vrhovima  $K_x, K_y$  i usmjerenim bridom  $(K_x, K_y)$ .
4. Ako nije pronađen niti jedan takav podgraf, onda se stvara novi graf  $PG = \{(K_x, K_y), (K_x, K_y)\}$  i dodaje se u skup podgrafova  $SPG$ .
5. Na kraju se u definira podgraf od izoliranih vrhova i doda se u skup  $SPG$ .

Algoritam 1.    Algoritam za rastavljanje grafa na komponente
<pre> SPG = ∅ for each (K<sub>x</sub>, K<sub>y</sub>) ∈ A   found = false   for each G<sub>i</sub> ∈ SPG, i=1, ..., n     if K<sub>x</sub> ∈ V<sub>i</sub> ∨ K<sub>y</sub> ∈ V<sub>i</sub> then       V<sub>i</sub> = V<sub>i</sub> ∪ {K<sub>x</sub>} ∪ {K<sub>y</sub>}       A<sub>i</sub> = A<sub>i</sub> ∪ {(K<sub>x</sub>, K<sub>y</sub>)}       found = true     end if   end for   if not found then     PG = {(K<sub>x</sub>, K<sub>y</sub>), (K<sub>x</sub>, K<sub>y</sub>)}     SPG = SPG ∪ {PG}   end if end for PG = (V \ ∪<sub>i=1</sub><sup>n</sup> V<sub>i</sub>, A \ ∪<sub>i=1</sub><sup>n</sup> A<sub>i</sub>) SPG = SPG ∪ {PG} </pre>

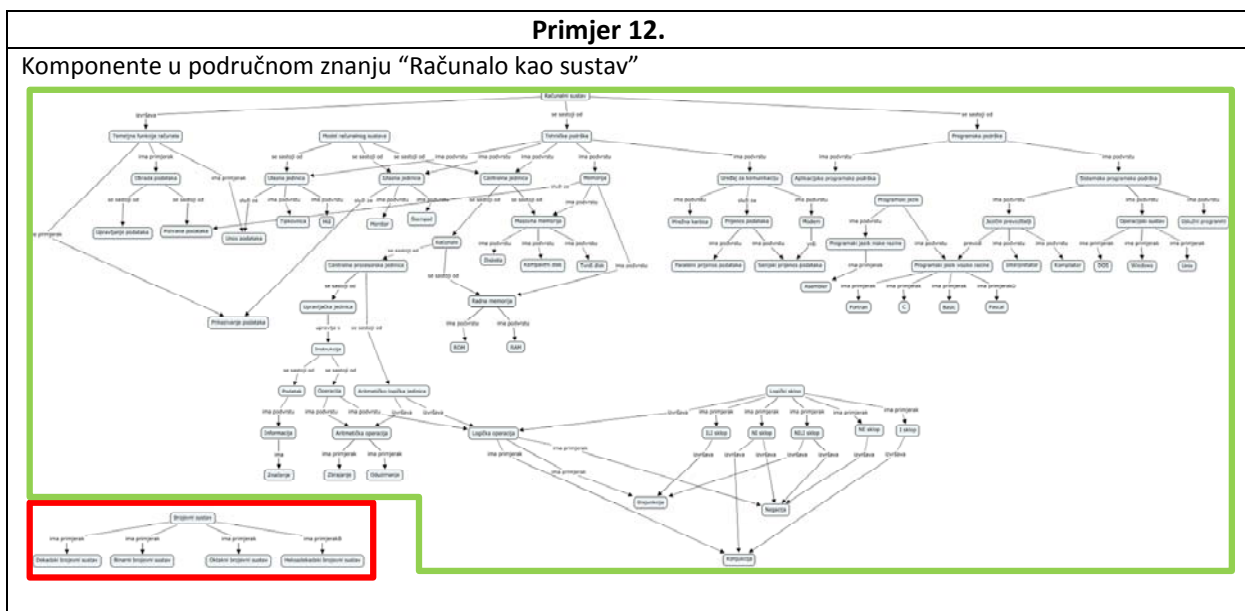
Svaki od algoritama koji su razrađeni u ovom poglavlju popraćeni su s dva primjera na dvije razine općenitosti. Prva razina općenitosti pokazuje malo područnom znanju koje ne

pokazuje semantiku konceptata. Druga razina općenitosti pokazuje semantički definirano područno znanje „Računalo kao sustav“.

Primjer 11 pokazuje primjenu Algoritam 1 na prvoj razini općenitosti:



Primjer 12 pokazuje primjenu Algoritam 1 na drugoj razini općenitosti:



Prethodno navedena relacija ekvivalencije u svojoj definiciji traži utvrđivanje da li između neka dva vrha postoji put. Pošto će nam ispitivanje postojanja puta trebati i kasnije, navodimo *algoritam za ispisivanje puta (ako postoji) između dva vrha  $K_x, K_y$  u grafu GPZ i određivanje njegove duljine  $l_{K_x, K_y}$*  (Algoritam 2):

1. Krećemo od vrha  $K' = K_x$  i postavljamo duljinu puta na 1.
2. Uzimamo jednog sljedbenika  $K_i$  od  $K'$  kojeg odmah brišemo iz skupa sljedbenika od  $K'$ .
3. Ako je  $K_i = K_y$ , onda smo pronašli put između  $K_x, K_y$ .
4. Ako  $K_i$  ima sljedbenike (dakle, nije list ili ima još „neposječenih“ sljedbenika), onda prvo dodajemo u varijablu *Put* brid  $K'K_i$  i vrh  $K_i$ , zatim zapamtimo njegovog prethodnika  $PrethodnikK_i = K'$  i postavimo  $K_i$  za sljedeći vrh od kojeg nastavljamo postupak u točki 2 za  $K' = K_i$ . Ovdje povećavamo duljinu puta za 1.
5. Ako  $K_i$  nema sljedbenike, a promatrani vrh  $K'$  nema više nijednog sljedbenika na kojeg bi moglo prijeći pretraživanje, onda prvo uklanjamo iz varijable *Put* zadnji brid i vrh, zatim se vraćamo na prethodnike od  $K'$  dok god ne pronađemo onog  $K' = PrethodnikK'$  koji ima još sljedbenika na koje možemo prebaciti pretraživanje. Ovdje smanjujemo duljinu puta za 1. Tada se vraćamo na točku 2.

<b>Algoritam 2. Algoritam za ispisivanje puta između dva vrha i određivanje njegove duljine</b>
<pre> found = false K' = K<sub>x</sub> Put = ({K<sub>x</sub>}, ∅) len=1 for each K<sub>i</sub> ∈ SljedK', K ≠ K<sub>x</sub>     SljedK' = SljedK' \ {K<sub>i</sub>}     if K<sub>i</sub> = K<sub>y</sub> then         Put = Put ∪ ({K<sub>i</sub>}, {K'K<sub>i</sub>})         len=len+1         found = true         exit for     end if     if SljedK<sub>i</sub> ≠ ∅ then         Put = Put ∪ ({K<sub>i</sub>}, {K'K<sub>i</sub>})         PrethodnikK<sub>i</sub> = K'         K' = K<sub>i</sub>         len=len+1     else         while SljedK' = ∅             Put = Put \ ({PrethodnikK'}, {PrethodnikK'K'})             K' = PrethodnikK'             len=len-1         loop     end if end for if found then ∃PutK<sub>x</sub>K<sub>y</sub> Put duljine len </pre>

Bitno je naglasiti da je u algoritmu važno pamtit i modificirane skupove sljedbenika  $SljedK'$  svakog promatranog vrha u onom obliku u kojem su ostali nakon eventualnog brisanja „posječenih“ vrhova (onih sljedbenika vrha  $K'$  za koje promatramo da li čine put do  $K_y$  i koje spremamo u varijablu  $K_i$ ). Varijabla  $PrethodnikK_i$  pamti neposrednog prethodnika vrha  $K_i$  i omogućava nam „vraćanje unatrag“, odnosno prelazak na sljedećeg sljedbenika vrha  $K'$ .

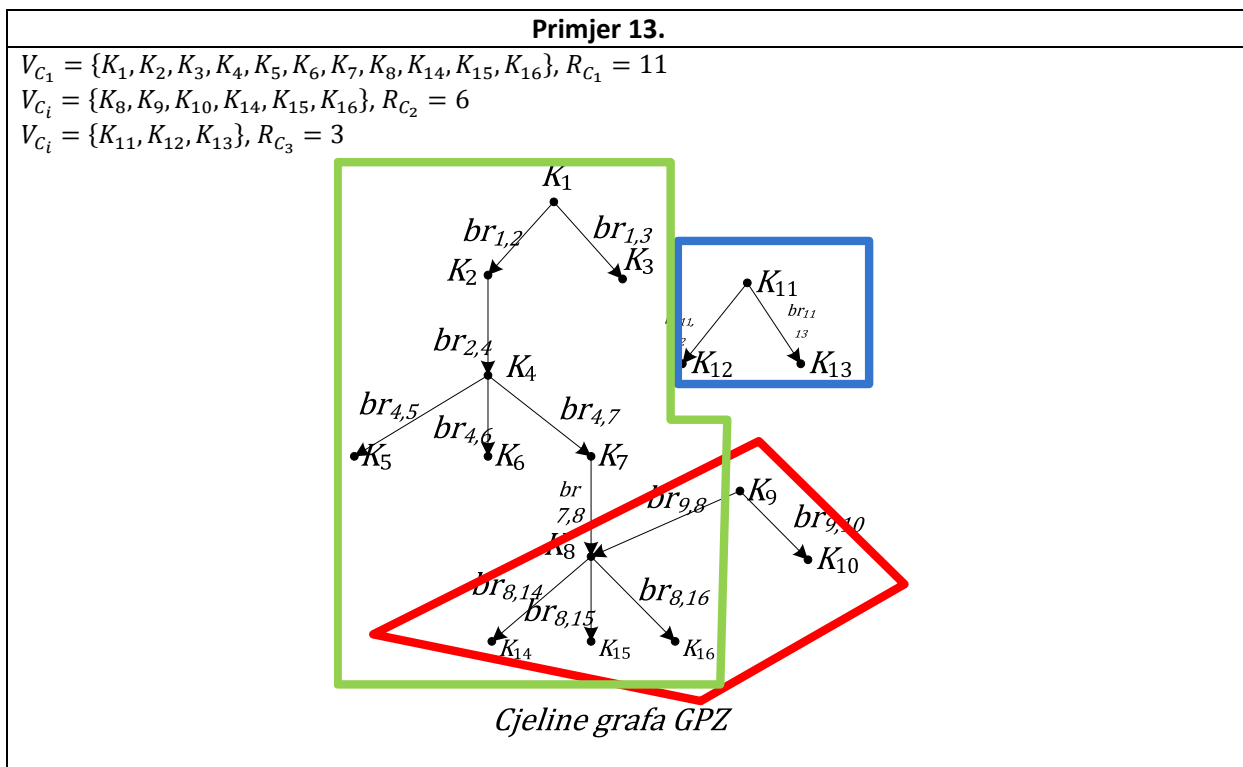
U drugom poglavlju smo definirali jedan vrlo važan pojam na kojem se temelji naš model stručnjaka, a to je cjelina (Definicija 2.6).

Prethodno navedeni algoritam traženja puta između dva vrha (Algoritam 2) predstavlja pretraživanje grafa po dubini počevši od vrha  $K_x$ . On pronalazi sve puteve između zadanih vrhova (putevi su uređeni leksikografski prvo prema nazivima relacija, a zatim prema nazivima koncepata). Algoritam za definiranje cjelina u komponenti grafa GPZ (Algoritam 3):

1. Definira se prazni skup  $C_i$  koji će sadržavati sve cjeline iz komponente  $G_i$  usmjerenog grafa  $GPZ = (V, A)$ .
2. Zatim se za svaki vrh  $K_x \in V_i$  provjerava da li je korijen, tj. da li postoji neki vrh  $K_y \in V_i$  takav da je  $(K_y, K_x) \in A_i$
3. Za svaki korijen  $K_x$  i za  $\forall K_z \in V_i, K_z \neq K_x$  se svi putevi  $Put_{K_x K_z}$  dodaju u cjelinu  $C_{K_x}$
4. Cjelina  $C_{K_x}$  se dodaje u skup  $C_i$ .

<b>Algoritam 3. Algoritam za definiranje cjelina u komponenti grafa</b>
<pre> <math>C_i = \emptyset</math> for each <math>K_x \in V_i</math>   root = true   for each <math>K_y \in V_i, K_y \neq K_x</math>     if <math>(K_y, K_x) \in A_i</math> then root = false   end for   if root then     for each <math>K_z \in V_i, K_z \neq K_x</math>       if <math>\exists Put_{K_x K_z}</math> then <math>C_i = C_i \cup Put_{K_x K_z}</math>     end for   end if end for </pre>

Primjer 13 pokazuje primjenu Algoritam 3 na prvoj razini općenitosti:



Sljedeće relacije „sljedbenik od“ i „prethodnik od“ omogućavaju identificiranje korijena podgrafova koji odgovara cjelini grafa područnog znanja.

**Definicija 4.8** Na skupu vrhova cjeline  $C_i = (V_{C_i}, A_{C_i})$  definiramo relacije parcijalnog uređaja:

$$K_y \text{ je sljedbenik od } K_x \Leftrightarrow \exists \text{Put } K_x K_y \subset C_i$$

$$K_x \text{ je prethodnik od } K_y \Leftrightarrow \exists \text{Put } K_x K_y \subset C_i$$

**Definicija 4.9** Centralni vrh  $K_{C_i}$  cjeline  $C_i$  je vrh za kojeg vrijedi:  $\forall K_x \in V_{C_i}$  vrijedi  $K_x$  je sljedbenik od  $K_{C_i}$ , odnosno  $K_{C_i}$  je prethodnik od  $K_x$ .

Centralni vrh cjeline nema prethodnika i od njega vodi put do svih ostalih vrhova. Bilo koji vrh u cjelini, osim centralnog vrha, ima jednog ili više neposrednih prethodnika. U sljedećem primjeru su navedeni centralni vrhovi cjelina iz prethodnog primjera.

**Primjer 14.**

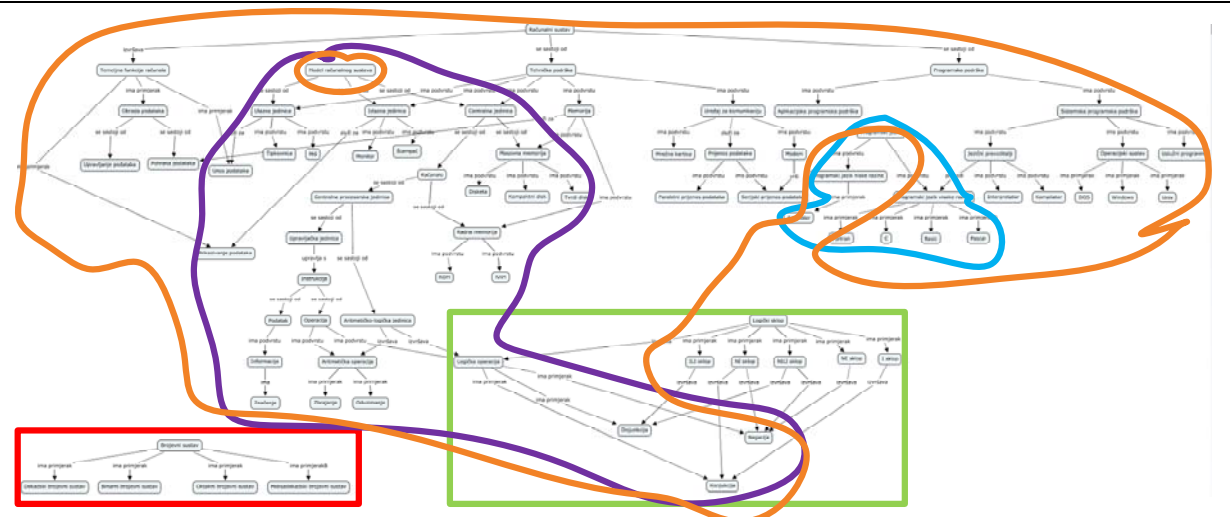
$$K_{C_1} = K_1$$

$$K_{C_2} = K_9$$

$$K_{C_3} = K_{11}$$

Primjer 15 pokazuje primjenu Algoritam 3 na drugoj razini općenitosti:

**Primjer 15.**



Cjeline u područnom znanju “Računalo kao sustav”

Cjelina “Računalni sustav” – narančasta -  $C_1$

Cjelina “Model računalnog sustava” – ljubičasta -  $C_2$

Cjelina “Programski jezik” – plava -  $C_3$

Cjelina “Logički sklop” – zelena -  $C_4$

Cjelina “Brojevni sustav” – crvena -  $C_5$

**Primjer 16.**

$K_{C_1}$  = Računalni sustav  
 $K_{C_2}$  = Model računalnog sustava  
 $K_{C_3}$  = Programski jezik  
 $K_{C_4}$  = Logički sklop  
 $K_{C_5}$  = Brojevni sustav

Cjelina je, dakle povezani podgraf u kojem je svaki vrh povezan s centralnim vrhom (osim njega samog) najmanje jednim putem. U cjelini nema izoliranih vrhova. To znači da nam Algoritam 2. za svaki vrh cjeline  $K_y \neq K_{C_i}$  ispisuje put  $PutK_{C_i}K_y$  i njegovu duljinu. Ovaj put je prvi pronađeni prema leksikografskom uređaju prvo prema nazivu relacija, a zatim prema nazivu koncepata. To nam omogućava stvaranje algoritma za traženje najvećeg puta  $PutK_{C_i}MaxVrh_{C_i}$  u nekoj cjelini, odnosno određivanje vrha  $MaxVrh_{C_i}$  koji definira taj put (Algoritam 4):

1. Odaberemo jednog  $K_z \in V_{C_i}, K_z \neq K_{C_i}$  i stavljamo  $K' = K_{C_i}$  i izvršimo Algoritam 2. koji nam kao izlazne vrijednosti da varijable  $Put$  i  $len$ .
2. Stavimo da je  $MaxPut=PutK_z$ ,  $Maxlen=lenK_z$  i  $MaxVrh=K_z$
3. Za svaki  $K_w \in V_{C_i}, K_w \neq K_{C_i}, K_w \neq K_z$  stavljamo  $K' = K_{C_i}$  i izvršimo Algoritam 2. koji nam kao izlazne vrijednosti da varijable  $Put$  i  $len$ .
4. Ako je  $lenK_y > Maxlen$ , onda je  $MaxPut=PutK_y$ ,  $Maxlen=lenK_y$  i  $MaxVrh=K_w$ .

Ako u cjelini  $C_i$  ima više najdužih puteva, onda dobivene vrijednosti  $MaxPut_{C_i}$ ,  $Maxlen_{C_i}$  i  $MaxVrh_{C_i}$  odgovaraju prvom nađenom od njih (prema leksikografskom uređaju).

**Algoritam 4. Algoritam za traženje najvećeg puta u nekoj cjelini**

```

 $K_z \in V_{C_i}, K_y \neq K_{C_i}$ 
izvrši Algoritam 2. za  $K_x = K_{C_i} i K_y = K_z$ 
 $MaxPut=PutK_z$ 
 $Maxlen=lenK_z$ 
 $MaxVrh=K_z$ 
for each  $K_w \in V_{C_i}, K_w \neq K_{C_i}, K_w \neq K_z$ 
  izvrši Algoritam 2. za  $K_x = K_{C_i} i K_y = K_w$ 
  if  $lenK_y > Maxlen$  then
     $MaxPut=PutK_w$ 
     $Maxlen=lenK_w$ 
     $MaxVrh=K_w$ 
  end if
end for
najduži je  $MaxPut$  duljine  $Maxlen$  do vrha  $MaxVrh$ 

```

Primjer 17 pokazuje primjenu Algoritam 4 na prvoj razini općenitosti:

**Primjer 17.**

$MaxVrh_{C_1} = K_{14}, MaxPut_{C_1} = (K_1, br_{1,2}, K_2, br_{2,4}, K_4, br_{4,7}, K_7, br_{7,8}, K_8, br_{8,14}, K_{14}), Maxlen_{C_1} = 5$   
 $MaxVrh_{C_2} = K_{14}, MaxPut_{C_2} = (K_9, br_{9,8}, K_8, br_{8,14}, K_{14}), Maxlen_{C_2} = 2$   
 $MaxVrh_{C_3} = K_{12}, MaxPut_{C_3} = (K_{11}, br_{11,12}, K_{12}), Maxlen_{C_3} = 1$

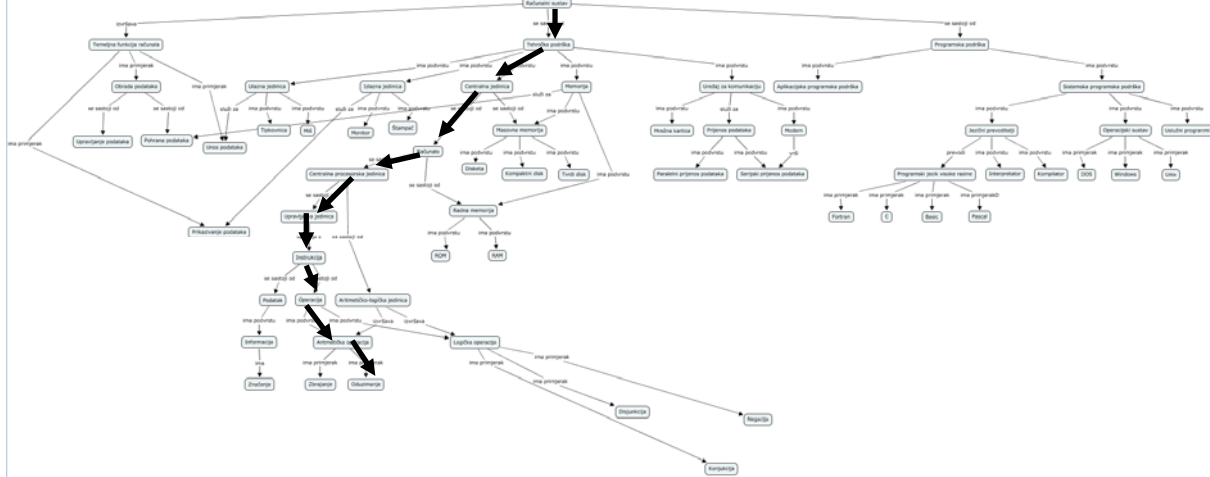
Primjer 18 pokazuje primjenu Algoritam 4 na drugoj razini općenitosti:

**Primjer 18.**

$MaxVrh_{c_1} =$  Oduzimanje

$MaxPut_{c_1} =$  (Računalni sustav, se sastoji od, Tehnička podrška, ima podvrstu, Centralna jedinica, se sastoji od, Računalo, se sastoji od, Centralna procesorska jedinica, se sastoji od, Upravljačka jedinica, upravlja s, Instrukcija, se sastoji od, Operacija, ima podvrstu, Aritmetička operacija, ima primjerak, Oduzimanje)

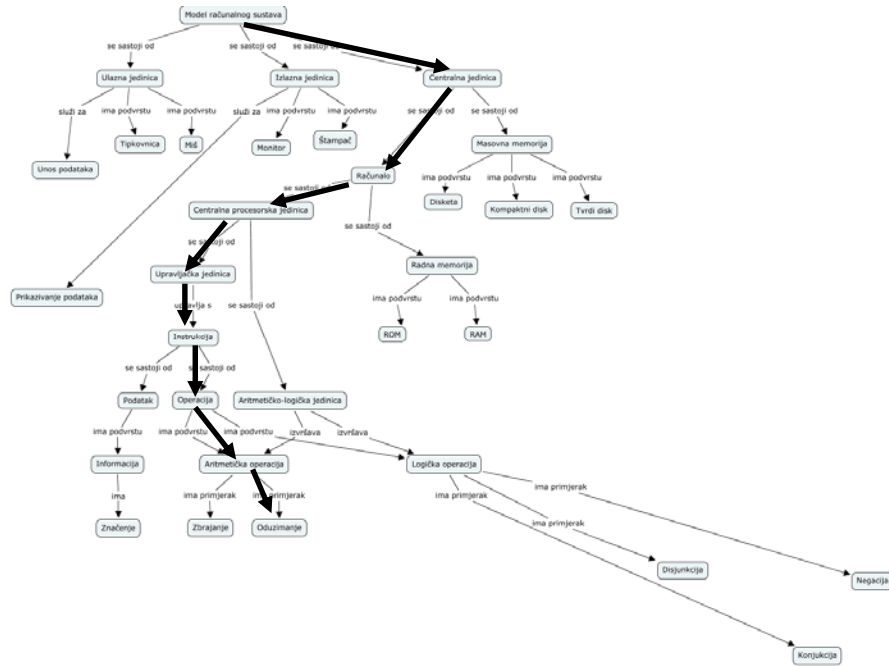
$Maxlen_{c_1} = 9$



$MaxVrh_{c_2} =$  Oduzimanje

$MaxPut_{c_2} =$  (Model računalnog sustava, se sastoji od, Centralna jedinica, se sastoji od, Računalo, se sastoji od, Centralna procesorska jedinica, se sastoji od, Upravljačka jedinica, upravlja s, Instrukcija, se sastoji od, Operacija, ima podvrstu, Aritmetička operacija, ima primjerak, Oduzimanje)

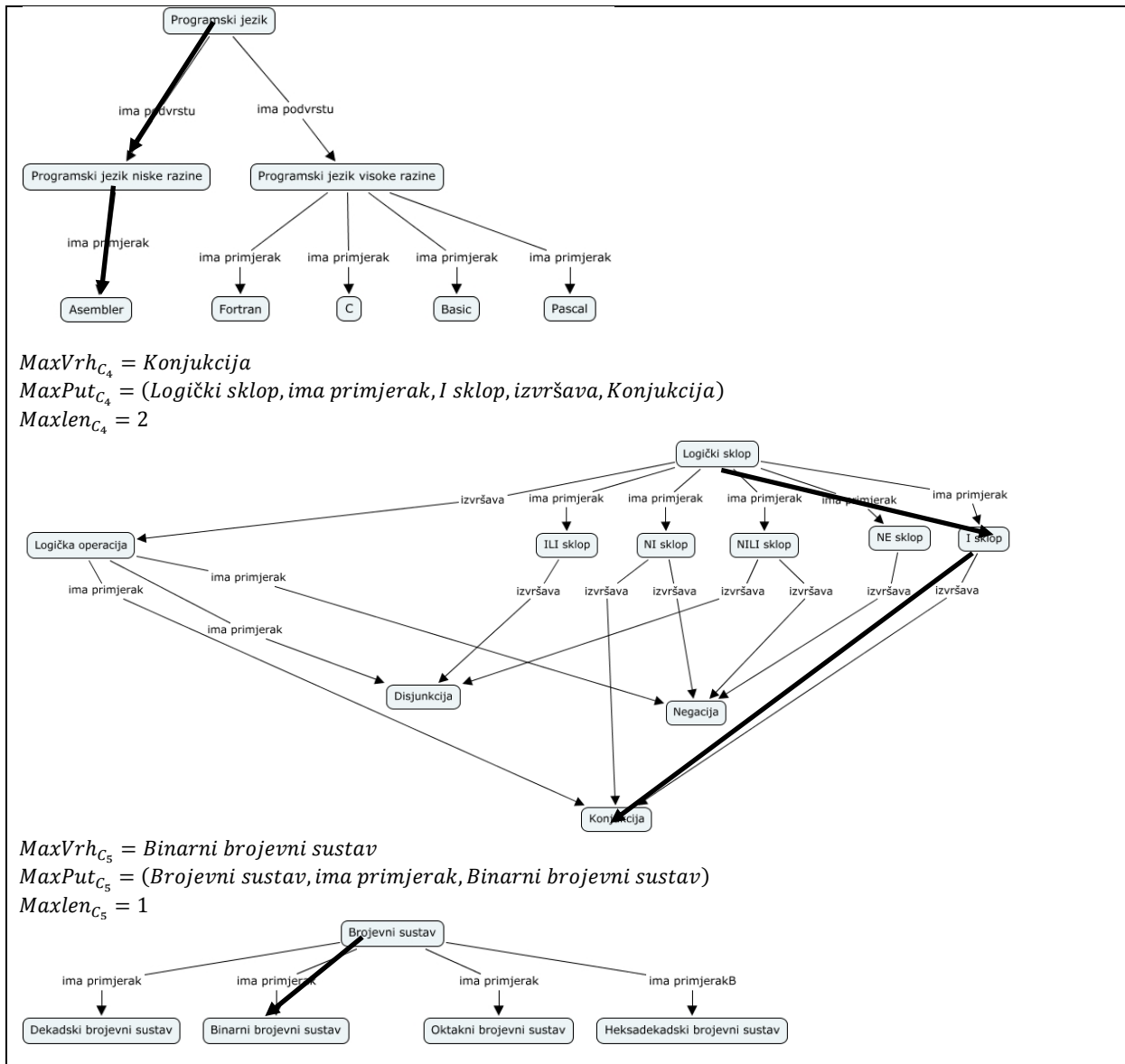
$Maxlen_{c_2} = 8$



$MaxVrh_{c_3} =$  Asembler

$MaxPut_{c_3} =$  (Programski jezik, ima podvrstu, Programski jezik niske razine, ima primjerak, Asembler)

$Maxlen_{c_3} = 2$



### 4.1.3 Reprezentacija područnog znanja

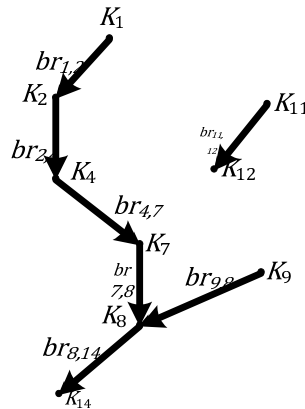
Pošto područno znanje može biti prilično veliko, potrebno je definirati takav podskup područnog znanja koji će dostojno reprezentirati cijelo područno znanje. Dakle, reprezentacijom područnog znanja obuhvaćamo sve koncepte i relacije koji su relevantni za neko područno znanje. U drugom poglavlju smo definirali reprezentaciju područnog znanja kao i skup relevantnih koncepata i relacija (Definicija 2.7, Definicija 2.8).

Pogledajmo kako izgleda reprezentacija na prvoj razini općenitosti u Primjer 19:



**Primjer 19.**

$$\begin{aligned}
 RepGPZ &= \bigcup_{i=1}^3 PutK_{C_i}MaxVrh_{C_i} \\
 &= (K_1, br_{1,2}, K_2, br_{2,4}, K_4, br_{4,7}, K_7, br_{7,8}, K_8, br_{8,14}, K_{14}) \cup (K_9, br_{9,8}, K_8, br_{8,14}, K_{14}) \cup (K_{11}, br_{11,12}, K_{12}) \\
 &= (\{K_1, K_2, K_4, K_7, K_8, K_{14}, K_9, K_{11}, K_{12}\}, \{br_{1,2}, br_{2,4}, br_{4,7}, br_{7,8}, br_{8,14}, br_{9,8}, br_{11,12}\})
 \end{aligned}$$

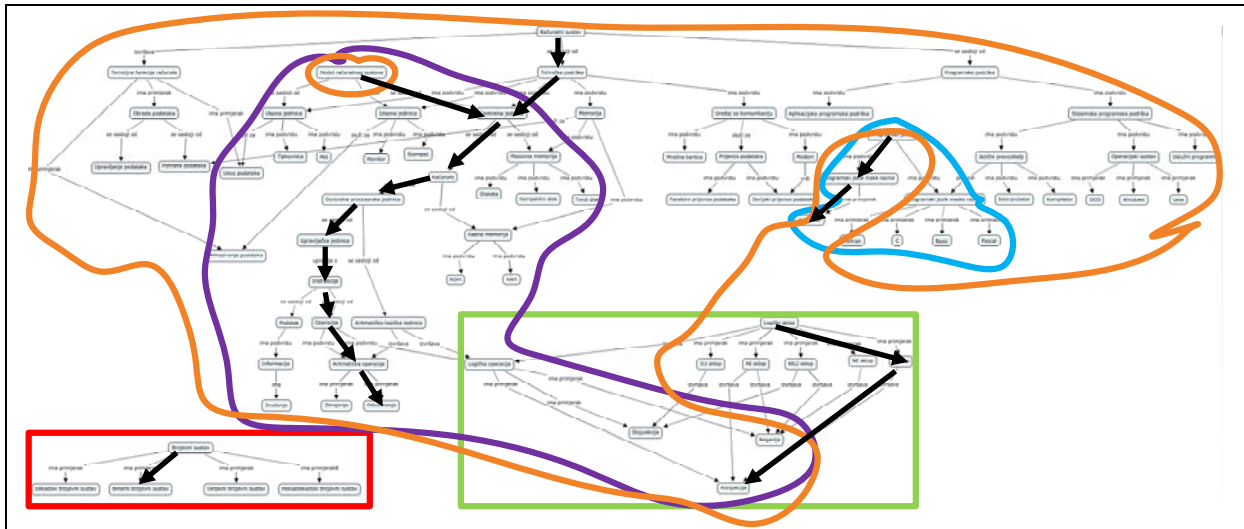


Reprezentativni graf grafa GPZ

Pogledajmo kako izgleda reprezentacija na drugoj razini općenitosti u Primjer 20:

**Primjer 20.**

$$\begin{aligned}
 RepGPZ &= \bigcup_{i=1}^5 PutK_{C_i}MaxVrh_{C_i} \\
 &= \left( \begin{array}{l} \text{Računalni sustav, se sastoji od, Tehnička podrška, ima podvrstu, Centralna jedinica,} \\ \text{se sastoji od, Računalo, se sastoji od, Centralna procesorska jedinica, se sastoji od,} \\ \text{Upravljačka jedinica, upravlja s, Instrukcija, se sastoji od, Operacija, ima podvrstu,} \\ \text{Aritmetička operacija, ima primjerak, Oduzimanje} \end{array} \right) \\
 &\cup \left( \begin{array}{l} \text{Model računalnog sustava, se sastoji od, Centralna jedinica, se sastoji od, Računalo,} \\ \text{se sastoji od, Centralna procesorska jedinica, se sastoji od, Upravljačka jedinica,} \\ \text{upravlja s, Instrukcija, se sastoji od, Operacija, ima podvrstu,} \\ \text{Aritmetička operacija, ima primjerak, Oduzimanje} \end{array} \right) \\
 &\cup \left( \begin{array}{l} \text{Programski jezik, ima podvrstu, Programski jezik niske razine,} \\ \text{ima primjerak, Asembler} \end{array} \right) \\
 &\cup \left( \begin{array}{l} \text{Logički sklop, ima primjerak, I sklop, izvršava, Konjukcija} \end{array} \right) \\
 &\cup \left( \begin{array}{l} \text{Brojevni sustav, ima primjerak, Binarni brojevni sustav} \end{array} \right) \\
 &= \left( \left\{ \begin{array}{l} \text{Računalni sustav, Tehnička podrška, Model računalnog sustava, Centralna jedinica,} \\ \text{Računalo, Centralna procesorska jedinica, Upravljačka jedinica, Instrukcija, Operacija,} \\ \text{Aritmetička operacija, Oduzimanje, Programski jezik, Programski jezik niske razine,} \\ \text{Asembler, Logički sklop, I sklop, Konjukcija, Brojevni sustav, Binarni brojevni sustav} \end{array} \right\}, \left. \begin{array}{l} \text{\{se sastoji od, ima podvrstu, upravlja s, ima primjerak, izvršava\}} \end{array} \right\}
 \end{aligned}$$



Smatramo da činjenica što algoritam za izbor najdužeg puta (Algoritam 4) uzima prvi takav pronađeni, ne umanjuje vrijednost cijelog algoritma, jer su svi putevi u grafu koji su jednake duljine za naš pristup potpuno izomorfni. To znači da ukoliko postoji u svakoj cjelini više najdužih puteva, onda njihovim kombiniranjem možemo dobiti više izomorfnih reprezentativnih podgrafova područnog znanja. Dakle, potpuno je svejedno da li je izbor najdužeg puta postavljen na prvi ili na posljednji takav nađen ili se pak radi o slučajno odabranom najvećem putu u nekoj cjelini, jer su svi tako dobiveni reprezentativni podgrafovi izomorfni.

**Primjer 21.**

$RepPZ = \{(K_1, part, K_2), (K_2, isa, K_4), (K_4, r_1, K_7), (K_7, r_3, K_8), (K_8, inst, K_{14}), (K_9, part, K_8), (K_{11}, r_2, K_{12})\}$

**Primjer 22.**

$RepPZ = \left\{ \begin{array}{l} (Računalni\ sustav, se\ sastoji\ od, Tehnička\ podrška), \\ (Tehnička\ podrška, ima\ podvrstu, Centralna\ jedinica), \\ (Centralna\ jedinica, se\ sastoji\ od, Računalo), \\ (Računalo, se\ sastoji\ od, Centralna\ procesorska\ jedinica), \\ (Centralna\ procesorska\ jedinica, se\ sastoji\ od, Upravljačka\ jedinica), \\ (Upravljačka\ jedinica, upravlja\ s, Instrukcija), (Instrukcija, se\ sastoji\ od, Operacija), \\ (Operacija, ima\ podvrstu, Aritmetička\ operacija), \\ (Aritmetička\ operacija, ima\ primjerak, Oduzimanje), \\ (Model\ računalnog\ sustava, se\ sastoji\ od, Centralna\ jedinica), \\ (Programski\ jezik, ima\ podvrstu, Programski\ jezik\ niske\ razine), (I\ sklop, izvršava, Konjukcija) \\ (Programski\ jezik\ niske\ razine, ima\ primjerak, Asembler), (Logički\ sklop, ima\ primjerak, I\ sklop), \\ (Brojevi\ sustav, ima\ primjerak, Binarni\ brojevi\ sustav) \end{array} \right\}$

Reprezentacija područnog znanja nam služi da bi odredili koncepte područnog znanja na temelju kojih će se generirati *ulazni test* kojim se provjerava predznanje učenika o nekom područnom znanju. Ovako određeni ulazni test garantira da će ispitivanje znanja učenika obuhvatiti sve koncepte i relacije koji su relevantni za neko područno znanje. Pošto smo

spomenuli mogućnost postojanja izomorfnih reprezentativnih podgrafova područnog znanja, jednako je moguće postojanje izomorfnih reprezentacija područnog znanja, ovisno o najvećem putu koji je odabran u svakoj cjelini grafa područnog znanja. Dakle, postoji do na izomorfizam jedinstvena reprezentacija područnog znanja.

U teoriji grafova postoji pojam razapinjuće stablo grafa. Razapinjuće stablo od  $G$  je razapinjući podgraf od  $G$  oblika  $G' = (V, A')$  (dakle sadrži sve vrhove, ali su neki bridovi izbačeni) koji je stablo. Nadalje, u literaturi se spominje i pojam *minimalno razapinjuće stablo*. To je pojam koji bi mogao poslužiti za provedbu ulaznog testa, ali smatramo da on to ne može biti jer bi onda ulazni test ispitivao poznavanje povezanosti svakog para koncepata iz područnog znanja. Ako je područno znanje veliko, onda bi ulazni test bio prevelik. Zato smo pristupili definiranju reprezentacije područnog znanja koja svodi broj pitanja u ulaznom testu na minimum, koji je ipak dovoljan da se dobije prikladna predodžba o predznanju učenika o nekom područnom znanju.

## 4.2 Generiranje, odabir, nizanje i prezentiranje elemenata računalom oblikovanog nastavnog sadržaja

U ovom radu se pažnja fokusira na modul učitelja, te posebno na način *generiranja, odabira, nizanja i prezentiranja* računalom oblikovanog nastavnog sadržaja.

Većina ITS-ova omogućava okruženje za učenje koje se temelji na „slobodnom“ pristupu učenju, tj. učenici samostalno i samovoljno biraju svoju putanju učenja (eng. learning path) u računalom oblikovanom nastavnog sadržaju. Na taj način mogu preskočiti usvajanje pojedinih elemenata računalom oblikovanog nastavnog sadržaja ili učiti nešto pogrešnim redoslijedom ili učiti nešto što nema njima odgovarajuću težinu (ili preteško ili prelagano). To je razlog zašto sustavi ITS-ovi moraju učenika sami voditi u procesu učenja i poučavanja i prezentirati im samo onaj računalom oblikovani nastavni sadržaj koji ima prikladni obim i težinu. Stoga ITS-ovi, prilikom odabira, nizanja i prezentiranja materijala za učenje učeniku, trebaju voditi računa o trenutnom znanju učenika i o složenosti elemenata računalom oblikovanog nastavnog sadržaja. Ovakvim pristupom će se smanjiti kognitivno preopterećivanje te će se omogućiti individualizirano vođenje procesa učenja i poučavanja.

Mnogi autori definiraju računalom oblikovani nastavni sadržaj (eng. *courseware*) kao vrstu nastavne ili obrazovne programske podrške ([www.cybermediacreations.com](http://www.cybermediacreations.com)) ili kao obrazovnu programsku podršku koja isporučuje nastavne materijale putem računala ([www.worldwidelearn.com](http://www.worldwidelearn.com)). Mi se ne slažemo s tim definicijama jer smatramo da je računalom oblikovani nastavni sadržaj samo dio sustava e-učenja, te se stoga ne može

smatrati programskom podrškom. Mi promatramo računalom oblikovane nastavne sadržaje kao sadržaje za učenje koji su stvoreni i prezentirani uz pomoć računala.

Pobliže ćemo objasniti svaki segment izgradnje računalom oblikovanog nastavnog sadržaja prilagođenog stereotipovima po znanju. Dakle, pobliže ćemo objasniti proces generiranja elemenata računalom oblikovanog nastavnog sadržaja, njihov odabira i nizanje, te prezentiranje.

U Definicija 2.6 smo definirali pojam *cjelina*. Taj pojam ćemo koristiti kao polaznu točku za definiranje elemenata računalom oblikovanog nastavnog sadržaja. Elementi računalom oblikovanog nastavnog sadržaja su *nastavne cjeline* (NC), *nastavne teme* (NT) i *nastavne jedinice* (NJ), kao i *testovi* (T) i *kvizovi* (K). Nastavne cjeline, teme i jedinice su elementi za učenje, a testovi i kvizovi su elementi za provjeravanje znanja.

U drugom poglavlju smo definirali pojmove računalom oblikovani nastavni sadržaj, element računalom oblikovanog nastavnog sadržaja za učenje i rang elementa (Definicija 2.9, Definicija 2.10, Definicija 2.11).

Dakle, za svaki element za učenje potrebno je samo definirati koncept  $K_x$  koji će biti korijen njegovog stabla. Ostali vrhovi podgrafa su svi oni  $K_y$  do kojih postoji put od  $K_x$ , a bridovi su svi oni bridovi iz svih mogućih puteva od  $K_x$  do  $K_y$ . Stoga uvodimo sljedeću oznaku  $NE_{K_x}$  za nastavni element čiji je korijen koncept  $K_x$ , tj. koji počinje u  $K_x$ , odnosno  $K_x$  je početak od  $NE_{K_x}$ .

Element računalom oblikovanog nastavnog sadržaja je jedinstveno određen svojim korijenom i rangom. To znači da možemo npr. imati nastavnu jedinicu s istim korijenom, ali različitom oznakom cjeline (prvi član u uređenoj trojci).

Definiramo atribut elementa računalom oblikovanog nastavnog sadržaja za učenje kojim ćemo označiti učenikovo poznavanje tog elementa, odnosno koncepta i relacija koji mu pripadaju:

**Definicija 4.10** Element računalom oblikovanog nastavnog sadržaja za učenje je **usvojen na razini**  $n$ ,  $n=0, 1, 2, 3, 4$ , ako  $\forall K_x K_y \in A', X_A(K_x K_y)=n$ . Ako je  $\forall K_x K_y \in A', X_A(K_x K_y)=-1$ , onda element računalom oblikovanog nastavnog sadržaja **nije usvojen**.

Prilikom **inicijalizacije modela učenika**, svaki element računalom oblikovanog nastavnog sadržaja u nekom područnom znanju se smatra da nije usvojen, odnosno vrijedi  $\forall K_x K_y \in A', X_A(K_x K_y)=-1$ .

## 4.2.1 Elementi za učenje

Za svaku vrstu elemenata računalom oblikovanog nastavnog sadržaja za učenje navodimo algoritme za generiranje. Algoritmi će biti pobliže objašnjeni primjerima na dvije razine općenitosti. Prva razina općenitosti pokazuje generiranje određene vrste elemenata na malom područnom znanju koje ne pokazuje semantiku koncepata. Druga razina općenitosti pokazuje generiranje određene vrste elemenata na semantički definiranom područnom znanju „Računalo kao sustav“. Pošto je ovo područno znanje veliko, njegov cijeli graf je prikazan u Prilogu 8.2.2 u formatu A3, a u primjeru je moguće vidjeti položaj određenih elemenata računalom oblikovanog nastavnog sadržaja u samom grafu.

### Faza 1 - Generiranje nastavnih cjelina

Za svaki korijen  $K_x$  u grafu područnog znanja koji je centralni vrh neke cjeline  $C_i$  ( $i$  je sortiran prema rangu cjeline) ponavljamo sljedeći algoritam za određivanje elemenata učenja:

Algoritam 5. Generiranje nastavnih cjelina
1. Ako u toj cjelini postoji put duljine veće od 3, onda definiramo $NC_{K_x}$ , tj. nastavnu cjelinu s početkom u $K_x$ . Rang $R_{NC_{K_x}} = (i, 0, 1)$ .
2. Inače, ako u toj cjelini postoji put duljine veće od 1, onda definiramo $NT_{K_x}$ , tj. nastavnu temu s početkom u $K_x$ . $R_{NT_{K_x}} = (i, 0, 2)$ .
3. Inače, definiramo $NJ_{K_x}$ , tj. nastavnu jedinicu s početkom u $K_x$ . $R_{NJ_{K_x}} = (i, 0, 3)$ .
4. Na kraju se u elemente računalom oblikovanog nastavnog sadržaja dodaju atributi i njihove vrijednosti (jer smo ih izbacili prilikom definiranja cjelina područnog znanja).

Primjer 23 pokazuje primjenu Algoritam 5 na prvoj razini općenitosti:

Primjer 23.
Cjelina $C_1$ s centralnim vrhom $K_1$ sadrži put duljine veće od 3, pa je ona nastavna cjelina, tj. $NC_{K_1} = C_1, R_{NC_{K_1}} = (1, 0, 1)$
Cjelina $C_2$ s centralnim vrhom $K_9$ ne sadrži put duljine veće od 3, ali sadrži put duljine veće od 1, pa je ona nastavna tema, tj. $NT_{K_9} = C_2, R_{NT_{K_9}} = (2, 0, 2)$
Cjelina $C_3$ s centralnim vrhom $K_{11}$ ne sadrži put duljine veće od 3, niti put duljine veće od 1, pa je ona nastavna jedinica, tj. $NJ_{K_{11}} = C_3, R_{NJ_{K_{11}}} = (3, 0, 3)$

Primjer 24 pokazuje primjenu Algoritam 5 na drugoj razini općenitosti:

**Primjer 24.**

Nastavne cjeline  
Nastavne teme  
Nastavne jedinice

Cjelina  $C_1$  s centralnim vrhom **Računalski sustav** sadrži put duljine veće od 3, pa je ona nastavna cjelina, tj.  $NC_{Računalski\ sustav} = C_1, R_{NC_{Računalski\ sustav}} = (1,0,1)$

Cjelina  $C_2$  s centralnim vrhom **Model računalnog sustava** sadrži put duljine veće od 3, pa je ona nastavna cjelina, tj.  $NC_{Model\ računalnog\ sustava} = C_1, R_{NC_{Model\ računalnog\ sustava}} = (2,0,1)$

Cjelina  $C_3$  s centralnim vrhom **Logički sklop** ne sadrži put duljine veće od 3, ali sadrži put duljine veće od 1, pa je ona nastavna tema, tj.  $NT_{Logički\ sklop} = C_3, R_{NT_{Logički\ sklop}} = (3,0,2)$

Cjelina  $C_4$  s centralnim vrhom **Programski jezik** ne sadrži put duljine veće od 3, ali sadrži put duljine veće od 1, pa je ona nastavna tema, tj.  $NT_{Programski\ jezik} = C_3, R_{NT_{Programski\ jezik}} = (4,0,2)$

Cjelina  $C_5$  s centralnim vrhom **Brojevi** ne sadrži put duljine veće od 3, niti put duljine veće od 1, pa je ona nastavna jedinica, tj.  $NJ_{Brojevi\ sustav} = C_3, R_{NT_{Brojevi\ sustav}} = (5,0,3)$

## Faza 2 - Generiranje nastavnih tema

Za svaku nastavnu cjelinu  $NT_{K_x}$  ponavljamo sljedeći algoritam za određivanje njenih nastavnih tema (nastavnu temu čine koncept i njegovi podkoncepti i njihovi podkoncepti; ako podkoncept ima samo jedan svoj podkoncept, onda se uzimaju i podkoncepti od tog podkoncepta):

<b>Algoritam 6. Generiranje nastavnih tema</b>
1. Odaberemo početak $K_x$ nastavne cjeline $NC_{K_x}$ .
2. Skup $V'$ vrhova nastavne teme $NT_{K_x}$ čini početak $K_x$ i njegovi podkoncepti (ali samo ako pripadaju odabranoj nastavnoj cjelini).
3. Ako podkoncept od $K_x$ ima samo jedan podkoncept, onda u $V'$ ulazi i taj podkoncept i njegovi podkoncepti (dakle koncepti koji su za 3 udaljeni od $K_x$ , ali samo ako pripadaju odabranoj nastavnoj cjelini ili ako su listovi).
4. Ako podkoncept od $K_x$ ima više podkonceptata, onda u $V'$ ulaze samo ti podkoncepti (dakle koncepti koji su za 2 udaljeni od $K_x$ , ali samo ako pripadaju odabranoj nastavnoj cjelini).
5. Ako nema nijedan koncept u točkama 3 i 4, onda ne možemo konstruirati nastavnu temu. Ako $K_x$ ima podkoncepte, onda konstruiramo nastavnu jedinicu od tog koncepta i njegovih

podkonceptata te od podpodkonceptata (ako postoje).

6. Za svaki list nastavne teme  $NT_{K_x}$  provjeravamo da li možemo konstruirati nastavnu temu koja počinje u njemu (mora postojati put u cjelini  $NC_{K_x}$  koji počinje u tom listu, a ima duljinu barem 2, a taj list mora imati više od jednog podkoncepta) na način opisan u koracima 2,3 i 4.
7. Korak 6 ponavljamo za svaku konstruiranu nastavnu temu.
8. Na kraju se u elemente računalom oblikovanog nastavnog sadržaja dodaju atributi i njihove vrijednosti (jer smo ih izbacili prilikom definiranja cjelina područnog znanja).

Primjer 25 pokazuje primjenu Algoritam 6 na prvoj razini općenitosti:

**Primjer 25.**

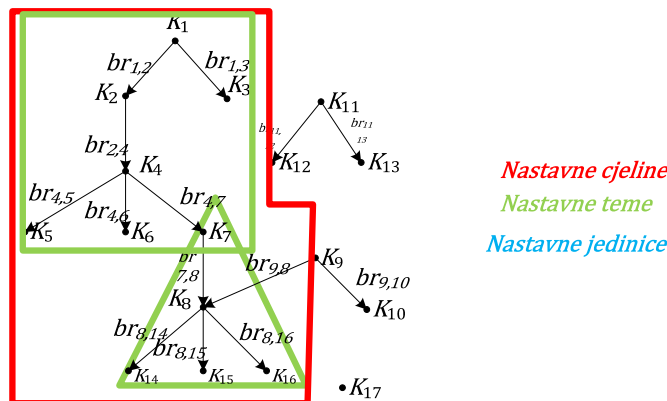
Definirajmo nastavne teme unutar nastavne cjeline  $NC_{K_1}$

Krećemo od centralnog vrha  $K_1$

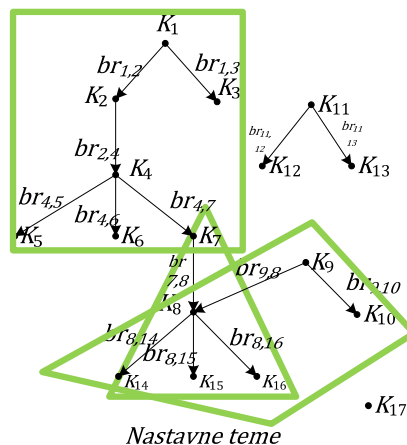
$$V_{NT_{K_1}} = \{K_1, K_2, K_3, K_4, K_5, K_6, K_7\}, R_{NT_{K_1}} = (1,0,2)$$

Promotrimo sada listove nastavne teme  $NT_{K_1}$  -  $K_5, K_6, K_7$ . Za koncepte  $K_5, K_6$  se ne može konstruirati niti nastavna tema niti nastavna jedinica jer nemaju podkonceptata. Za koncept  $K_7$  se može konstruirati nastavna tema jer postoji put u cjelini  $NC_{K_1}$  koji počinje u tom listu, a ima duljinu barem 2

$$V_{NT_{K_7}} = \{K_7, K_{14}, K_{15}, K_{16}\}, R_{NT_{K_7}} = (1,3,2)$$



Analogno definiramo ostale nastavne teme:



Primjer 26 pokazuje primjenu Algoritam 6 na drugoj razini općenitosti:

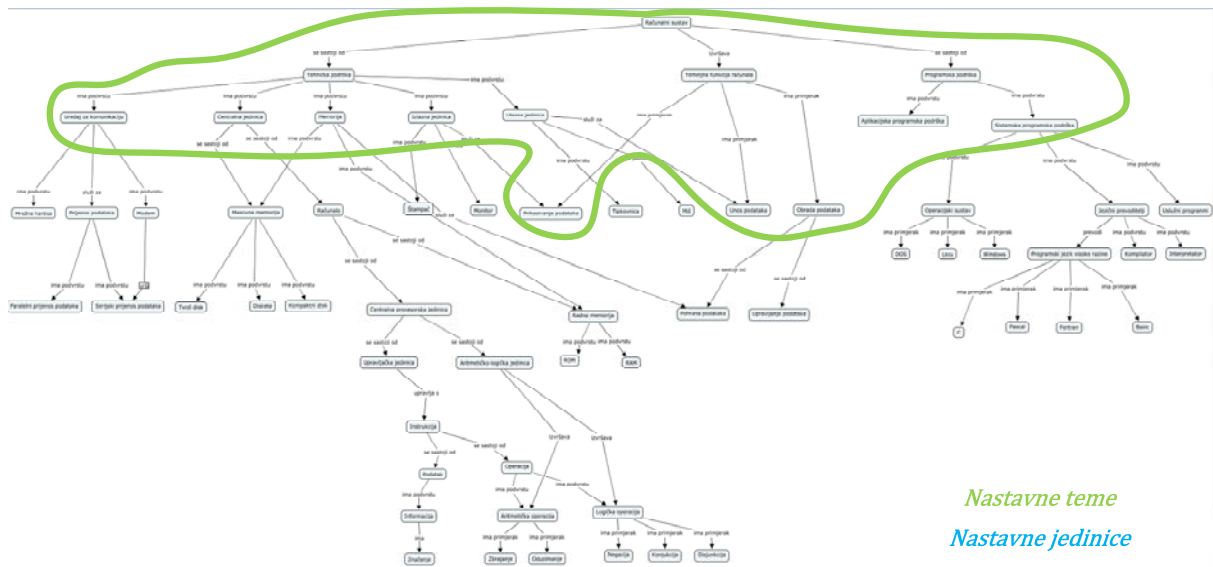
**Primjer 26.**

Definirajmo nastavne teme unutar nastavne cjeline  $NC_{Računalni\ sustav}$

Krećemo od centralnog vrha **Računalni sustav**

$$V_{NT_{Računalni\ sustav}} = \left\{ \begin{array}{l} Računalni\ sustav, Temeljna\ funkcija\ računala, Tehnička\ podrška, \\ Programska\ podrška, Prikazivanje\ podataka, Unos\ podataka, \\ Obrada\ podataka, Centralna\ jedinica, Izlazna\ jedinica, Memorija, \\ Ulazna\ jedinica, Uređaj\ za\ komunikaciju, \\ Aplikacijska\ programska\ podrška, Sistemska\ programska\ podrška \end{array} \right\}$$

$$R_{NT_{Računalni\ sustav}} = (1,0,2)$$



Promotrimo sada listove nastavne teme  $NT_{Računalni\ sustav}$  - **Prikazivanje podataka, Unos podataka, Obrada podataka, Centralna jedinica, Izlazna jedinica, Memorija, Ulazna jedinica, Uređaj za komunikaciju, Aplikacijska programska podrška, Sistemska programska podrška**

Za koncepte **Prikazivanje podataka, Unos podataka** i **Aplikacijska programska podrška** se ne može konstruirati niti nastavna tema niti nastavna jedinica.

Za koncepte **Obrada podataka, Izlazna jedinica** i **Ulazna jedinica** možemo konstruirati nastavne jedinice:

$$V_{NJ_{Obrada\ podataka}} = \{Obrada\ podataka, Pohrana\ podataka, Upravljanje\ podataka\},$$

$$R_{NJ_{Obrada\ podataka}} = (1,2,3)$$

$$V_{NJ_{Izlazna\ jedinica}} = \{Izlazna\ jedinica, Monitor, Prikazivanje\ podataka, Štampač\},$$

$$R_{NJ_{Izlazna\ jedinica}} = (1,2,3)$$

$$V_{NJ_{Ulazna\ jedinica}} = \{Ulazna\ jedinica, Miš, Unos\ podataka, Tipkovnica\}, R_{NJ_{Ulazna\ jedinica}} = (1,2,3)$$

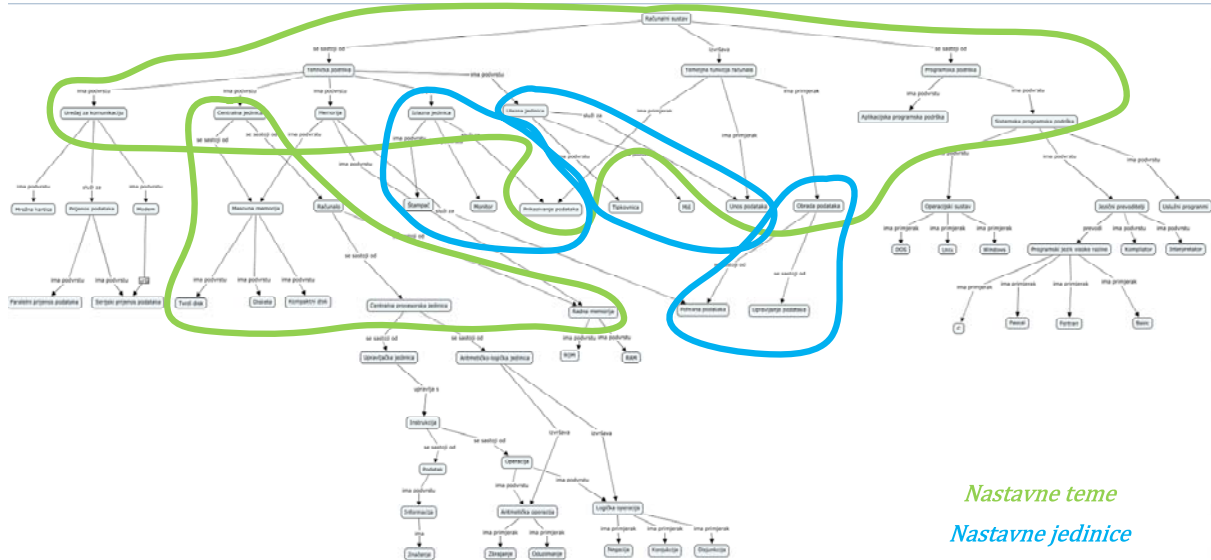
Za koncept **Centralna jedinica** se može konstruirati nastavna tema jer postoji put u cjelini  $NC_{Računalni\ sustav}$  koji



počinje u tom listu, a ima duljinu barem 2

$$V_{NT_{Centralna\ jedinica}} = \{Centralna\ jedinica, Masovna\ memorija, Računalo, Tvrdi\ disk, Disketa, Kompaktni\ disk, Radna\ memorija, Centralna\ procesorska\ jedinica\}$$

$$R_{NT_{Centralna\ jedinica}} = (1,2,2)$$



Konstruiranje ostalih elemenata računalom oblikovanog nastavnog sadržaja možete vidjeti u Prilogu 8.4.1.

### Faza 3 - Generiranje nastavnih jedinica

Za svaku nastavnu temu  $NT_{K_x}$  ponavljamo sljedeći algoritam za određivanje njenih nastavnih jedinica (nastavnu jedinicu čine koncept i njegovi podkoncepti; ako koncept ima samo jedan podkoncept, onda se uzimaju i podkoncepti od podkoncepta):

<b>Algoritam 7. Generiranje nastavnih jedinica</b>
1. Odaberemo početak $K_x$ nastavne teme $NT_{K_x}$ .
2. Skup $V'$ vrhova nastavne jedinice $NJ_{K_x}$ čini početak $K_x$ i njegovi podkoncepti (ali samo ako pripadaju odabranoj nastavnoj temi).
3. Ako $K_x$ ima samo jedan podkoncept, onda u $V'$ ulazi i njegovi podkoncepti (dakle koncepti koji su za 2 udaljeni od $K_x$ , ali samo ako pripadaju odabranoj nastavnoj temi ili ako su listovi).
4. Za svaki list nastavne jedinice $NJ_{K_x}$ provjeravamo da li možemo konstruirati nastavnu jedinicu koja počinje u njemu (mora postojati put u odabranoj temi koji počinje u tom listu, a ima duljinu barem 1) na način opisan u koracima 2 i 3.
5. Korak 4 ponavljamo za svaku konstruiranu nastavnu jedinicu.
6. Svi izolirani vrhovi iz grafa područnog znanja čine nastavne jedinice same za sebe (specijalni oblik nastavnih jedinica).
7. Na kraju se u elemente računalom oblikovanog nastavnog sadržaja dodaju atributi i njihove vrijednosti (jer smo ih izbacili prilikom definiranja cjelina područnog znanja).

Primjer 27 pokazuje primjenu Algoritam 7 na prvoj razini općenitosti:

**Primjer 27.**

Definirajmo nastavne jedinice unutar nastavne teme  $NT_{K_1}$

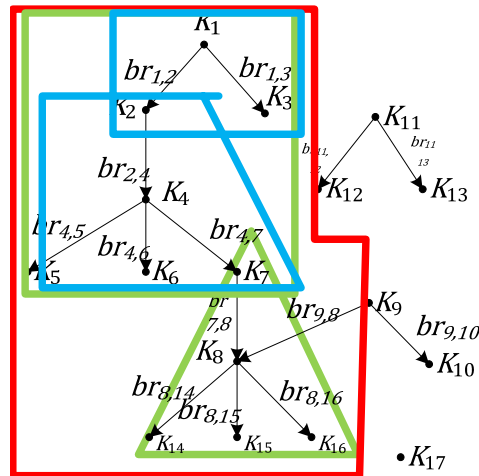
Krećemo od centralnog vrha  $K_1$

$$V_{NJ_{K_1}} = \{K_1, K_2, K_3\}, R_{NT_{K_1}} = (1,0,3)$$

Promotrimo sada listove nastavne jedinice  $NJ_{K_1} - K_2, K_3$ . Za koncept  $K_3$  se ne može konstruirati nastavna jedinica jer nema podkonceptata. Za koncept  $K_2$  se može konstruirati nastavna jedinica jer postoji put u temi  $NT_{K_1}$  koji počinje u tom listu, a ima duljinu barem 1

$$V_{NJ_{K_2}} = \{K_2, K_4, K_5, K_6, K_7\}, R_{NJ_{K_2}} = (1,1,3)$$

Promotrimo sada listove nastavne jedinice  $NJ_{K_2} - K_5, K_6, K_7$ . Ne mogu se konstruirati nastavne jedinice jer ne postoji put u temi  $NT_{K_1}$  koji počinje u tim listovima, a ima duljinu barem 1.

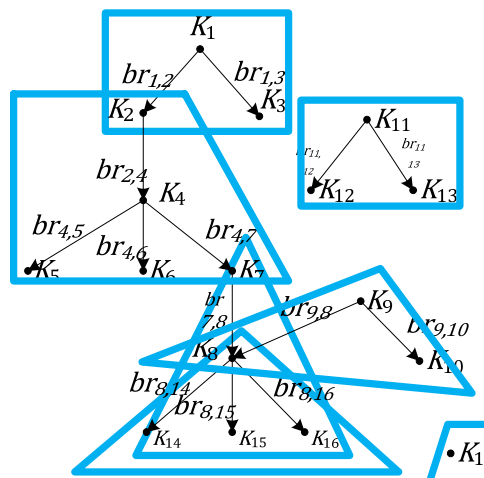


*Nastavne cjeline*

*Nastavne teme*

*Nastavne jedinice*

Analogno definiramo ostale nastavne jedinice:



*Nastavne jedinice*

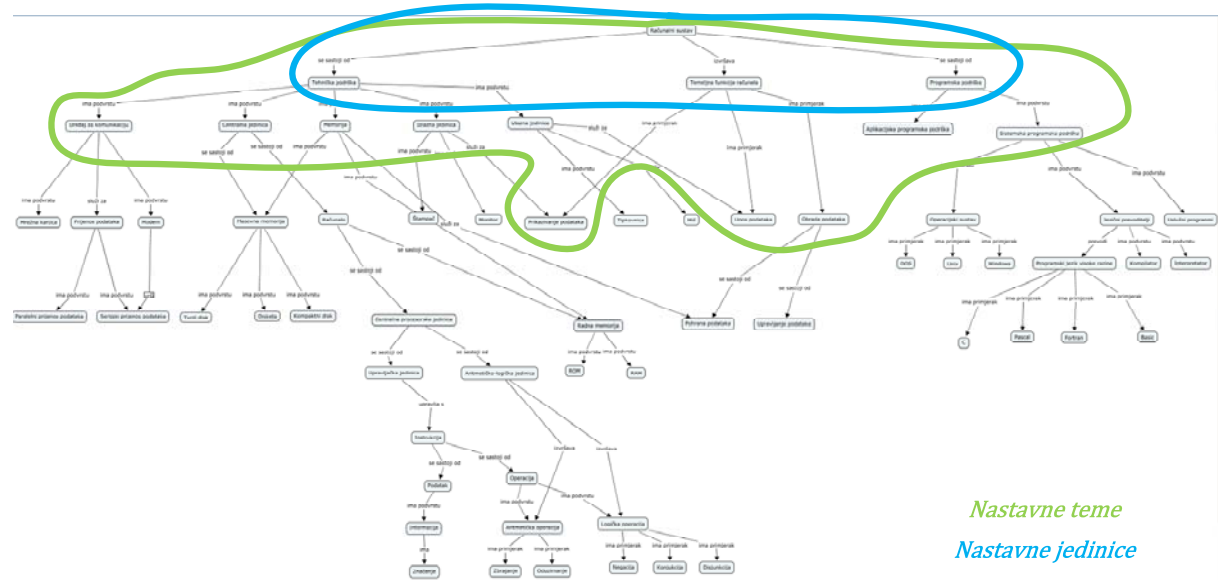
Primjer 28 pokazuje primjenu Algoritam 7 na drugoj razini općenitosti:

**Primjer 28.**

Definirajmo nastavne jedinice unutar nastavne teme  $NT_{Računalni\ sustav}$

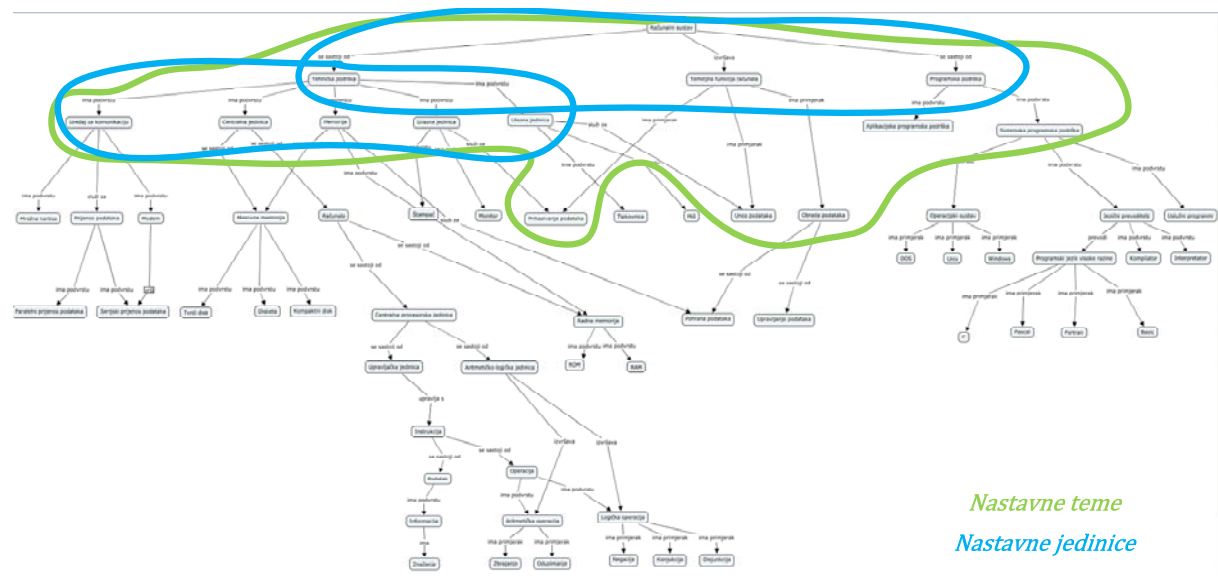
Krećemo od centralnog vrha **Računalni sustav**

$$V_{NJ_{Računalni\ sustav}} = \left\{ \begin{array}{l} Računalni\ sustav, Temeljna\ funkcija\ računala, \\ Tehnička\ podrška, Programska\ podrška \end{array} \right\}, R_{NJ_{Računalni\ sustav}} = (1,0,3)$$



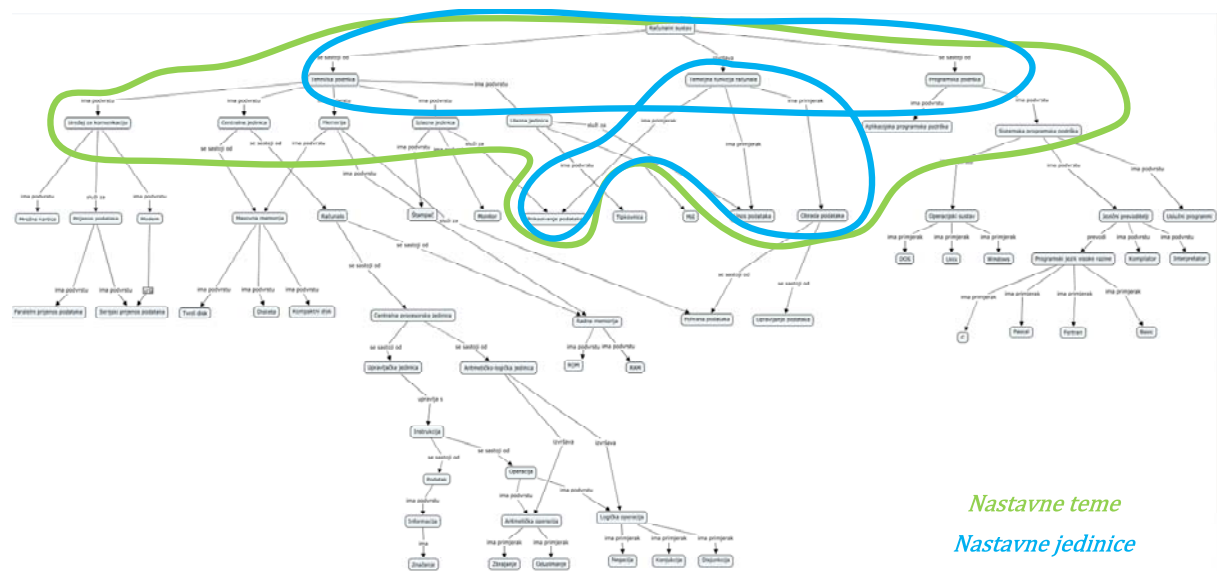
Promotrimo sada listove nastavne jedinice  $NJ_{Računalni\ sustav}$  - **Tehnička podrška, Temeljna funkcija računala, Programska podrška**. Za svih se može konstruirati nastavna jedinica jer postoji put u temi  $NT_{Računalni\ sustav}$  koji počinje u tom listu, a ima duljinu barem 1.

$$V_{NJ_{Tehnička\ podrška}} = \left\{ \begin{array}{l} Tehnička\ podrška, Centralna\ jedinica, \\ Memorija, Izlazna\ jedinica, Ulazna\ jedinica, \\ Uređaj\ za\ komunikaciju \end{array} \right\}, R_{NJ_{Tehnička\ podrška}} = (1,1,3)$$



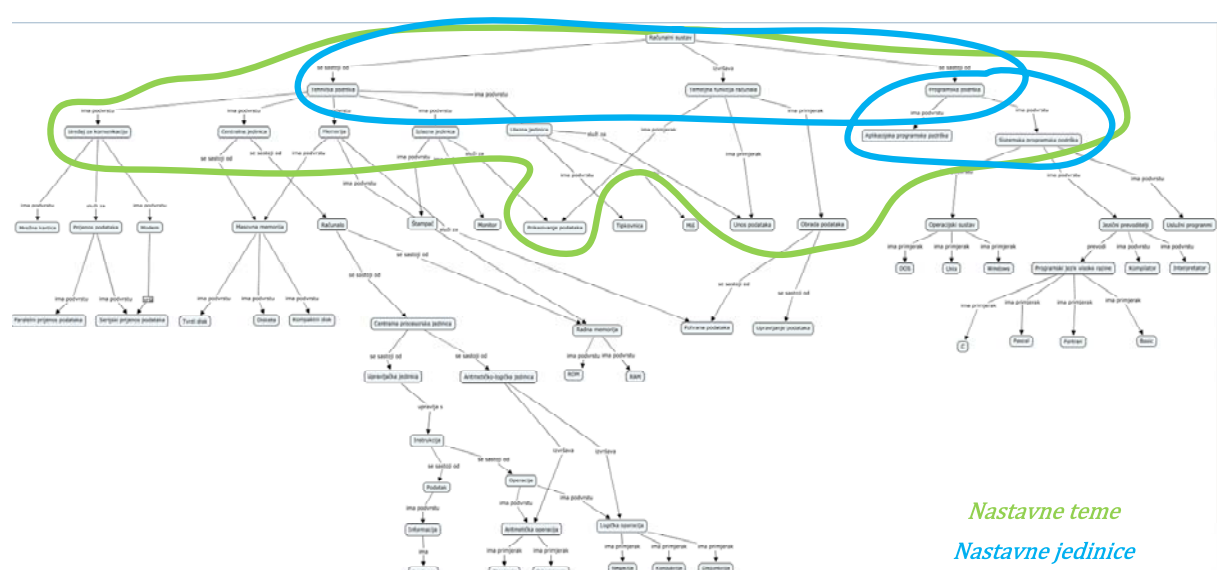
Promotrimo sada listove nastavne jedinice  $NJ_{Tehnička\ podrška}$ . Za nijednog se ne može konstruirati nastavna jedinica jer ne postoji put u temi  $NT_{Računalni\ sustav}$  koji počinje u tom listu, a ima duljinu barem 1.

$$V_{NJ_{Tehnička\ podrška}} = \{ \text{Temeljna funkcija računala, Unos podataka, Prikazivanje podataka, Obrada podataka} \}, R_{NJ_{Tehnička\ podrška}} = (1,1,3)$$



Promotrimo sada listove nastavne jedinice  $NJ_{Temeljna\ funkcija\ računala}$ . Za nijednog se ne može konstruirati nastavna jedinica jer ne postoji put u temi  $NT_{Računalni\ sustav}$  koji počinje u tom listu, a ima duljinu barem 1.

$$V_{NJ_{Programska\ podrška}} = \{ \text{Programska podrška, Aplikacijska programska podrška, Sistemska programska podrška} \}, R_{NJ_{Programska\ podrška}} = (1,1,3)$$



Promotrimo sada listove nastavne jedinice  $NJ_{Programska\ podrška}$ . Za nijednog se ne može konstruirati nastavna jedinica jer ne postoji put u temi  $NT_{Računalni\ sustav}$  koji počinje u tom listu, a ima duljinu barem 1.

Konstruiranje ostalih elemenata računalom oblikovanog nastavnog sadržaja možete vidjeti u Prilogu 8.4.1.

Algoritmi za generiranje nastavnih cjelina, tema i jedinica (Algoritam 5, Algoritam 6, Algoritam 7) osiguravaju sljedeće:

- presjek svake dvije nastavne teme je najviše jedan koncept (list jedne nastavne teme je korijen druge)
- presjek svake dvije nastavne jedinice je najviše jedan koncept (list jedne nastavne jedinice je korijen druge)
- svaka nastavna cjelina se sastoji od barem jedne nastavne teme i jedne nastavne jedinice
- svaka nastavna tema se sastoji od barem dvije nastavne jedinice
- jedan koncept može biti korijen različitih vrsta elemenata računalom oblikovanog nastavnog sadržaja u istoj cjelini, ali i u različitim cjelinama
- jedan koncept može biti korijen iste vrste elemenata računalom oblikovanog nastavnog sadržaja u različitim cjelinama

Na primjer, **Aritmetičko-Logička jedinica** može biti korijen i nastavne teme i nastavne jedinice. **Centralna jedinica** je korijen nastavne teme u cjelini **Računalni sustav**, a korijen nastavne jedinice u cjelini **Model računalnog sustava**. **Izlazna jedinica** je korijen nastavne cjeline i u cjelini **Računalni sustav** i u cjelini **Model računalnog sustava**.

Minimalni uvjeti za generiranje elemenata računalom oblikovanog nastavnog sadržaja za učenje su sljedeći:

- nastavna cjelina odgovara cjelini područnog znanja → postojanje barem jednog puta duljine 4 i više od početka elementa
- nastavna tema → postojanje barem jednog puta duljine 2 od početka elementa i barem dva podpodkoncepta početka ili postojanje barem jednog puta duljine 3 od početka elementa
- nastavna jedinica → postojanje barem jednog puta duljine 1 od početka elementa i barem dva podkoncepta početka ili postojanje barem jednog puta duljine 2 od početka elementa

## 4.2.2 Elementi za provjeravanje znanja

Proces učenja i poučavanja je iterativan proces u kojem se stalno izmjenjuju učenje i provjeravanje znanja. Provjerava se ono znanje koje se prethodno učilo. U našoj pedagoškoj praksi je uvriježena uporaba termina kao što su nastavna cjelina, tema i jedinica koje čine organizacijsku strukturu računalom oblikovanog nastavnog sadržaja. Dakle, može se provjeravati znanje učenika na razini nastavne cjeline, teme ili jedinice i jasno je da obim znanja koji se ispituje na različitim razinama nije jednak. Stoga razlikujemo: (i) test kao takav skup pitanja kojim se provjerava znanje učenika o većem skupu koncepata koji odgovaraju

---

nastavnoj cjelini i (ii) kviz koji obuhvaća pitanja o konceptima koji odgovaraju nastavnoj temi ili jedinici i obično je vremenski ograničen na petnaest ili manje minuta.

Oba načina ispitivanja znanja uključuju sljedeće tipove pitanja (Gronlund, 1985):

1. Dosjećanje (eng. short-answer)

Sastoji se od pitanja ili nedovršene tvrdnje na koja treba dati kratak odgovor (npr. jedna riječ) – pitanje otvorenog tipa.

2. Jednostruki izbor (eng. true-false)

Predstavlja tvrdnju za koje se traži procjena točnosti. Tvrdnje moraju biti nedvosmislene i ne smiju se sastojati od dijelova koji su točni i dijelova koji nisu točni. Preporuča se izbjegavanje složenih rečenica.

3. Dopunjavanje (eng. completion)

Postoje prazna mjesta na koja treba upisati riječi koje nedostaju. Može biti otvorenog ili zatvorenog tipa.

4. Višestruki izbor (eng. multiple-choice)

Može tražiti točan odgovor među netočnima, netočan među točnima, uzrok među posljedicama i dr. Svi predloženi odgovori moraju izgledati podjednako prihvatljivi i traženi odgovor ne smije odskakati ni po jednoj svojoj značajki (dužinom ili značenjem). Mjesto traženog odgovora treba varirati.

Na ova pitanja odgovara se biranjem nekoliko od više ponuđenih odgovora od kojih su neka točna. Za razliku od pitanja jednostrukog izbora, ova pitanja su bolji pokazatelj znanja učenika jer se povećanjem broja odgovora smanjuje mogućnost pogađanja.

5. Sređivanje (eng. ranking)

Sastoji se od niza povezanih pojmova i činjenica koje se prezentiraju nesređeno. Treba ih srediti po određenom kriteriju.

6. Povezivanje (eng. matching)

Sastoji se od dva niza riječi ili rečenica (ili nekih drugih simbola) koje treba međusobno povezati. Traže povezivanje članova dvaju nizova riječi ili rečenica. Dužina nizova ne smije biti jednaka jer je visoka vjerojatnost pogađanja pri povezivanju posljednjih članova nizova. Pri izradi zadataka povezivanja treba voditi brigu o tome da broj elemenata koji se povezuju ne bude manji od četiri para ni veći od šesnaest parova.

Ova pitanja omogućuju ispitivanje znanja učenika kroz sparivanje danih pojmova. Posebno su korisni za testiranje onih područja znanja u kojima postoji veliki broj pojmova i činjenica. Ponuđene opcije, u obliku riječi, učenik treba upariti s nekom drugom riječi

Kategorija pitanja, odnosno vrsta znanja koju oni ispituju, je određena prema Bloomovoj taksonomiji prihvaćenoj u cijelom svijetu (Bloom, 1956). Definiramo četiri težinske kategorije pitanja. U Tablica 4.1. navodimo koja težinska kategorija odgovara na koja pitanja:

Tablica 4.1. Težinske kategorije pitanja u kvizu

Kategorija pitanja	Pitanja
1 znanje na razini reprodukcije činjenica	nabrojati prepoznati definirati imenovati poredati
2 znanje na razini razumijevanja	objasniti označiti proširiti razlikovati razvrstati
3 znanje na razini primjene	zaključiti izdvojiti raščlaniti promijeniti odrediti
4 znanje na razini analize, sinteze i procjene	usporediti povezati razlikovati analizirati upotpuniti

Različitim vrstama pitanja provjeravamo različite razine znanja učenika. Ovisno o načinu postavljanja pitanja različitim zadacima se mogu ispitivati i druge razine znanja po Bloomu, pa na primjer u zadatku povezivanja možemo zadati nekoliko parova koji su svi lančano povezani, time je pitanje dodatno otežano pa se na taj način mogu ispitivati i više razine znanja. U Tablica 4.2. vidimo kojom vrstom pitanja ispitujemo koju vrstu znanja ((eLAN, 2004), (HEC-PIEAS, 2009), (Notar et al., 2004), (Roberson-Scott, 2005)).

Tablica 4.2. Bloomova taksonomija i vrste pitanja

Kategorija pitanja	Vrsta pitanja
1 znanje na razini reprodukcije činjenica	Jednostruki izbor Višestruki izbor Jednostavno dosjećanje Povezivanje
2 znanje na razini razumijevanja	Jednostruki izbor Dopunjavanje Višestruki izbor Jednostavno dosjećanje Povezivanje
3 znanje na razini primjene	Višestruki izbor Jednostavno dosjećanje Povezivanje
4 znanje na razini analize, sinteze i procjene	Višestruki izbor Jednostavno dosjećanje Povezivanje Sređivanje

U drugom poglavlju smo definirali element računalom oblikovanog nastavnog sadržaja za provjeravanje znanja (Definicija 2.12). Pošto element za provjeravanje znanja čine pitanja, u nastavku ćemo pokazati kako se ta pitanja automatski i dinamički generiraju.

### 4.2.2.1 Predložci pitanja

Pošto se znanje provjerava iterativno u procesu učenja i poučavanja, za generiranje testova je potrebno znati samo nad kojim skupom koncepata i relacija je potrebno izvršiti provjeru znanja. Dakle, ulaz za svako generiranje testa je određeni podskup područnog znanja koji odgovara reprezentaciji, nastavnoj cjelini, jedinici ili temi. Zato uvodimo sljedeće oznake:

- podskup područnog znanja nad kojim se vrši testiranje  
 $PZ' = \{(K_y, r, K_x) | (K_y, r, K_x) \in PZ\}$
- graf podskupa područnog znanja  $GPZ' = (V', A')$  nastao na temelju  $PZ'$

Ako se radi o ulaznom testu, onda je  $PZ' = RepPZ, GPZ' = RepGPZ$ , tj.  $V' = V_{RepGPZ}, A' = A_{RepGPZ}$ .

Element za provjeravanje znanja, odnosno test, se sastoji od određenog broja pitanja koja se stvaraju prema predlošcima. Da bi se odredilo kojim predloškom provjeravamo koju kategoriju znanja, navodimo predložke pitanja prema težinskim kategorijama. Ne može se svakim predloškom provjeravati poznavanje povezanosti bilo kojeg podskupa koncepata, već je kod svakog predložka naveden i uvjet koji mora biti ispunjen da bi se moglo generirati pitanje prema tom predlošku.

#### Prva težinska kategorija

Predložak1: Jednostruki izbor	
Uvjet za generiranje	$K_x, K_y \in V' \setminus (V_A \cup V_V) - K_x$ zadan, $K_y$ slučajno izabran iz $GPZ'$
Pitanje	Da li su $K_x$ i $K_y$ neposredno povezani?
Mogući odgovori	a) Da b) Ne c) Ne znam
Točan odgovor	Da, ako $\exists (K_x, K_y) \vee \exists (K_y, K_x) \in A'$ , inače je točan odgovor Ne
Napomena	Odabiru se dva proizvoljna koncepta koja mogu, ali ne moraju biti neposredno povezana. Jedino je važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta.
Primjer	Da li su koncepti Model računalnog sustava i Aritmetička operacija neposredno povezani? a) Da b) Ne c) Ne Znam

Predložak2: Jednostruki izbor	
Uvjet za generiranje	$K_x, K_y \in V' \setminus (V_A \cup V_V), r \in E_{REL} \setminus \{slot, filler, \emptyset\} - K_x$ zadan, $K_y$ i $r$ slučajno izabrani iz $GPZ'$
Pitanje	Da li su $K_x$ i $K_y$ neposredno povezani s $r$ ?
Mogući odgovori	a) Da b) Ne c) Ne znam
Točan odgovor	Da, ako $\exists (K_x, r, K_y) \vee \exists (K_y, r, K_x) \in PZ'$ , inače je točan odgovor Ne
Napomena	Odabiru se dva proizvoljna koncepta koja mogu, ali ne moraju biti neposredno povezana zadanom relacijom. Jedino je važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta. Također, relacija ne smije biti <i>slot</i> ili <i>filler</i> .
Primjer	<b>Da li su</b> Centralna jedinica <b>i</b> Upravljačka jedinica <b>neposredno povezani sa</b> služi za? a) Da b) Ne c) Ne Znam



Predložak3: Jednostruki izbor	
Uvjet za generiranje	$K_x, K_y \in V' \setminus (V_A \cup V_V), K_x \in V' \setminus V_K, K_y \in V' \setminus V_L - K_x$ zadan, $K_y$ slučajno izabran iz GPZ'
Pitanje	Da li je $K_x$ podkoncept od $K_y$ ?
Mogući odgovori	a) Da b) Ne c) Ne znam
Točan odgovor	Da, ako $\exists(K_y, K_x) \in A'$ , inače je točan odgovor Ne
Napomena	Odabiru se dva proizvoljna koncepta koja mogu, ali ne moraju biti neposredno povezana. Jedino je važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta. Također, koncept $K_x$ ne smije biti korijen, dok koncept $K_y$ ne smije biti list. Razlog tomu je uklanjanje očiglednosti ukoliko bi se provjeravalo da li korijen može biti podkoncept ili da li list ima podkoncept.
Primjer	<b>Da li je</b> Računalo <b>podkoncept od</b> Centralna procesorska jedinica? a) Da b) Ne c) Ne Znam

Predložak4: Jednostruki izbor	
Uvjet za generiranje	$K_x, K_y \in V' \setminus (V_A \cup V_V), K_x \in V' \setminus V_L, K_y \in V' \setminus V_K - K_x$ zadan, $K_y$ slučajno izabran iz GPZ'
Pitanje	Da li je $K_x$ nadkoncept od $K_y$ ?
Mogući odgovori	a) Da b) Ne c) Ne znam
Točan odgovor	Da, ako $\exists(K_x, K_y) \in A'$ , inače je točan odgovor Ne
Napomena	Odabiru se dva proizvoljna koncepta koja mogu, ali ne moraju biti neposredno povezana. Jedino je važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta. Također, koncept $K_x$ ne smije biti list, dok koncept $K_y$ ne smije biti korijen. Razlog tomu je uklanjanje očiglednosti ukoliko bi se provjeravalo da li list može biti nadkoncept ili da li korijen ima nadkoncept.
Primjer	<b>Da li je</b> Instrukcija <b>nadkoncept od</b> Upravljačka jedinica? a) Da b) Ne c) Ne Znam

Predložak5: Višestruki izbor	
Uvjet za generiranje	$K_x, K_y \in V' \setminus (V_A \cup V_V), \exists(K_x, r_{tocna}, K_y) \in PZ' - K_x$ zadan, $(K_x, r_{tocna}, K_y)$ slučajno izabran iz PZ', od ponuđenih odgovora jedna relacija mora biti $r_{tocna}, r_{odgi} \in E_{REL}$
Pitanje	Što neposredno povezuje $K_x$ i $K_y$ ?
Mogući odgovori	a) $r_{odg1}$ b) $r_{odg2}$ c) $r_{odg3}$ d) $r_{tocna}$ e) Ne znam
Točan odgovor	$r_{tocna}$
Napomena	Odabiru se dva proizvoljna koncepta koja moraju biti neposredno povezana nekom relacijom. Također je važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta.
Primjer	<b>Što neposredno povezuje</b> Instrukcija <b>i</b> Operacija? a) se sastoji od b) vrši c) ima primjerak d) služi za e) Ne znam

**Druga težinska kategorija**

Predložak6: Jednostruki izbor	
Uvjet za generiranje	$K_x, K_y \in V' \setminus (V_A \cup V_V), \exists (K_y, K_x) \in A - K_x$ zadan, $K_y$ slučajno izabran iz GPZ'
Pitanje	Da li su $K_x$ i $K_y$ posredno povezani?
Mogući odgovori	a) Da b) Ne c) Ne znam
Točan odgovor	Da, ako je $\exists \text{Setnja } K_x, K_y \subset GPZ', l_{K_x, K_y} > 3$ , inače je točan odgovor Ne
Napomena	Odabiru se dva proizvoljna koncepta koja ne smiju biti neposredno povezana. Također je važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta.
Primjer	<b>Da li su</b> Instrukcija <b>i</b> Upravljačka jedinica <b>posredno povezani</b> ? a) Da b) Ne c) Ne Znam

Predložak7: Višestruki izbor	
Uvjet za generiranje	$K_x \in V' \setminus (V_A \cup V_V \cup V_L), \exists r \in E_{REL} (\exists (K_x, r, K_{tocalan}) \in PZ') - K_x$ zadan, $(K_x, r, K_{tocalan})$ slučajno izabran iz PZ', od ponuđenih odgovora jedan koncept mora biti $K_{tocalan}, K_{odgj} \in V' \setminus (V_A \cup V_V)$
Pitanje	Koji podkoncept je neposredno povezan vezom $r$ s $K_x$ ?
Mogući odgovori	a) $K_{odg1}$ b) $K_{odg2}$ c) $K_{odg3}$ d) $K_{tocalan}$ e) Ne znam
Točan odgovor	$K_{tocalan}$
Napomena	Odabire se jedan proizvoljni koncept koji ne smije biti atribut ili vrijednost nekog drugog koncepta. Također treba imati podkoncept $K_{tocalan}$ s kojim je povezan zadanom relacijom u podskupu područnog znanja koje se provjerava. Ostali ponuđeni odgovori ne smiju biti atributi ili vrijednosti nekog drugog koncepta.
Primjer	<b>Koji podkoncept neposredno povezuje vezom</b> se sastoji od <b>s</b> Instrukcija? a) Upravljačka jedinica b) Operacija c) Centralna jedinica d) Aritmetička operacija e) Ne znam

Predložak8: Višestruki izbor	
Uvjet za generiranje	$K_x \in V' \setminus (V_A \cup V_V \cup V_K), \exists r \in E_{REL} (\exists (K_{tocalan}, r, K_x) \in PZ') - K_x$ zadan, $(K_{tocalan}, r, K_x)$ slučajno izabran iz PZ', od ponuđenih odgovora jedan koncept mora biti $K_{tocalan}, K_{odgj} \in V' \setminus (V_A \cup V_V)$
Pitanje	Koji nadkoncept je neposredno povezan vezom $r$ s $K_x$ ?
Mogući odgovori	a) $K_{odg1}$ b) $K_{odg2}$ c) $K_{odg3}$ d) $K_{tocalan}$ e) Ne znam
Točan odgovor	$K_{tocalan}$
Napomena	Odabire se jedan proizvoljni koncept koji ne smije biti atribut ili vrijednost nekog drugog koncepta. Također treba imati nadkoncept $K_{tocalan}$ s kojim je povezan zadanom relacijom u podskupu područnog znanja koje se provjerava. Ostali ponuđeni odgovori ne smiju biti atributi ili vrijednosti nekog drugog koncepta.
Primjer	<b>Koji nadkoncept je neposredno povezan vezom</b> se sastoji od <b>s</b> Računalo? a) Centralna jedinica b) Računalo c) Centralna procesorska jedinica d) Instrukcija e) Ne znam

### Treća težinska kategorija

Predložak9: Višestruki izbor	
Uvjet za generiranje	$K_x, K_y \in V' \setminus (V_A \cup V_V) - K_x$ zadan, $K_y$ slučajno izabran iz GPZ'
Pitanje	$K_x$ je _____ od $K_y$
Mogući odgovori	a) podkoncept b) nadkoncept c) nisu povezani d) ne znam
Točan odgovor	podkoncept ako $\exists(K_y, K_x) \in A'$ , nadkoncept ako $\exists(K_x, K_y) \in A'$ , nisu povezani ako $\nexists(K_y, K_x) \wedge \nexists(K_x, K_y) \in A'$
Napomena	Odabiru se dva proizvoljna koncepta koja mogu, ali ne moraju biti neposredno povezana. Jedino je važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta.
Primjer	Centralna procesorska jedinica <u>je</u> _____ <u>od</u> Tehnička podrška a) podkoncept b) nadkoncept c) nisu povezani d) ne znam

Predložak10: Dopunjavanje	
Uvjet za generiranje	$K_x \in V' \setminus (V_A \cup V_V) - K_x$ zadan
Pitanje	S koliko konceptata je povezan koncept $K_x$ ?
Mogući odgovori	upisati
Točan odgovor	$nK_x + pK_x$
Napomena	Odabire se jedan proizvoljni koncept koji ne smije biti atribut ili vrijednost nekog drugog koncepta.
Primjer	<b>S koliko konceptata je povezan koncept</b> Računalo?

Predložak11: Dopunjavanje	
Uvjet za generiranje	$K_x \in V' \setminus (V_A \cup V_V) - K_x$ zadan
Pitanje	Koliko podkonceptata ima $K_x$ ?
Mogući odgovori	upisati
Točan odgovor	$pK_x$
Napomena	Odabire se jedan proizvoljni koncept koji ne smije biti atribut ili vrijednost nekog drugog koncepta.
Primjer	<b>Koliko podkonceptata ima</b> Centralna jedinica?

Predložak12: Dopunjavanje	
Uvjet za generiranje	$K_x \in V' \setminus (V_A \cup V_V) - K_x$ zadan
Pitanje	Koliko nadkonceptata ima $K_x$ ?
Mogući odgovori	upisati
Točan odgovor	$nK_x$
Napomena	Odabire se jedan proizvoljni koncept koji ne smije biti atribut ili vrijednost nekog drugog koncepta.
Primjer	<b>Koliko nadkonceptata ima</b> Računalni sustav?

Predložak13: Povezivanje	
Uvjet za generiranje	$\forall(K_{odgai}, K_{odgbi}) \in A', K_{odgai}, K_{odgbi} \in V' \setminus (V_A \cup V_V) -$ odabrati od 4 do 5 uređenih trojki iz PZ'
Pitanje	Poveži zadane koncepte (nadkoncept – podkoncept) a) $K_{odga1}$ a) $K_{odgb1}$ b) $K_{odga2}$ b) $K_{odgb2}$ c) $K_{odga3}$ c) $K_{odgb3}$ d) $K_{odga4}$ d) $K_{odgb4}$
Mogući odgovori	Treba navest parove rednih brojeva konceptata koji su povezani
Točan odgovor	$\forall(K_{odgai}, K_{odgbi}) \in A'$
Napomena	Odabiru se svi parovi povezanih koncepta koji ne smiju biti atribut ili vrijednost nekog drugog koncepta. Slučajnim odabirom se jedan koncept iz para postavlja u lijevu listu, a drugi u desnu listu.
Primjer	<b>Poveži zadane koncepte</b> a) Logička operacija a) Konjukcija b) Upravljačka jedinica b) Instrukcija c) Instrukcija c) Operacija d) Operacija d) Logička operacija

### Četvrta težinska kategorija

Predložak14: Dopunjavanje	
Uvjet za generiranje	$K_x, K_y \in V' \setminus (V_A \cup V_V), \exists (K_x, r_{tocna}, K_y) \in PZ' - K_x$ zadan, $(K_x, r_{tocna}, K_y)$ slučajno izabran
Pitanje	$K_x$ i $K_y$ su neposredno povezani s _____
Mogući odgovori	upisati
Točan odgovor	<p>Ispravnost odgovora se procjenjuje ispitivanjem sličnosti unesene riječi s nazivom relacije <math>r_{tocna}</math>. Smatramo da je odgovor točan ako postoji sličnost od 80%. <i>Sličnost</i> se provjerava sljedećim algoritmom (Algoritam 8.):</p> <ol style="list-style-type: none"> <li>1. Tražimo niz znakova koji postoje u unesenoj riječi u redoslijedu u kojem su zapisani u toj riječi, a koji postoje i u nazivu relacije <math>r_{tocna}</math>.</li> <li>2. Ako takav niz postoji, onda je sličnost jednaka broj znakova u nizu podijeljen s brojem znakova u nazivu relacije <math>r_{tocna}</math>.</li> <li>3. Ako takav niz ne postoji, onda je sličnost 0.</li> </ol> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p style="text-align: center;"><b>Algoritam 8. Algoritam za provjeravanje sličnosti dvije riječi</b></p> <pre> a\$=odgovor b\$=naziv od r niz\$="" for each znakb\$ u b\$   for each znaka\$ u a\$     if znaka\$=znakb\$ then       niz\$=niz\$+znaka\$     exit for   end if end for end for sličnost=(LEN(niz\$)/LEN(b\$))*10                     </pre> </div>
Napomena	Odabiru se dva proizvoljna koncepta koja moraju biti neposredno povezana nekom relacijom. Također je važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta.
Primjer	Logička operacija i Konjunkcija <b><u>su neposredno povezani s relacijom</u></b> ima primjerak.

Predložak15: Sređivanje	
Uvjet za generiranje	$K_{odgj} \in V' \setminus (V_A \cup V_V), \exists K_{odgj+1} \in A' -$ odabire se najduži put u $GPZ'$ za svako pitanje, $j>1$
Pitanje	Poredaj: <ol style="list-style-type: none"> <li>1. <math>K_{odg1}</math></li> <li>2. <math>K_{odg2}</math></li> <li>3. ...</li> <li>4. <math>K_{odgn}</math></li> </ol>
Mogući odgovori	Treba navesti redne brojeve koncepta
Točan odgovor	$K_{odgj}K_{odgj+1} \in A'$
Napomena	Odabiru se proizvoljni koncepti koji moraju činiti povezani niz koncepta. Također je važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta.
Primjer	<b><u>Poredaj:</u></b> <ol style="list-style-type: none"> <li>1. Instrukcija</li> <li>2. Upravljačka jedinica</li> <li>3. Centralna procesorska jedinica</li> <li>4. Operacija</li> </ol>

Predložak16: Jednostruki izbor	
Uvjet za generiranje	$K_x \in V' \setminus (V_A \cup V_V), \exists K_z \in V' \setminus (V_A \cup V_V),$ $K_{slot} \in V_A, K_{filler} \in V_V ((K_x, slot, K_{slot}), (K_z, filler, K_{filler}) \in PZ')$ - odabire se $K_x$ za koji postoji $K_{slot}$ takav da postoji $(K_x, slot, K_{slot})$ , kao i neki $K_{filler}$ takav da postoji $(K_z, filler, K_{filler})$
Pitanje	Da li $K_x$ za $K_{slot}$ ima $K_{filler}$ ?
Mogući odgovori	a) Da b) Ne c) Ne znam
Točan odgovor	Da, ako $\exists (K_x, slot, K_{slot}), (K_{slot}, filler, K_{filler}), (K_x, filler, K_{filler}) \in PZ'$ , inače je točan odgovor Ne

Napomena	Odabiru se jedan proizvoljni koncept $K_x$ , jedan njegov proizvoljni atribut $K_{slot}$ i jedna proizvoljna vrijednost $K_{filler}$ . Vrijednost se bira tako da nije nužno da zajedno s odabranim atributom čini okvir zadanog koncepta $K_x$ .
Primjer	<b>Da li</b> Disketa <b>za</b> Kapacitet <b>ima</b> 2? a) Da b) Ne c) Ne Znam

Predložak17: Višestruki izbor	
Uvjet za generiranje	$K_x \in V' \setminus (V_A \cup V_V), K_{slot} \in V_A, \exists (K_x, slot, K_{slot}) \in PZ', K_{odgi} \in V_V$ - odabire se $K_x$ za koji postoji $K_{slot}$ takav da postoji $(K_x, slot, K_{slot})$
Pitanje	Koja je $K_{slot}$ od $K_x$ ?
Mogući odgovori	a) $K_{odg1}$ b) $K_{odg2}$ c) $K_{odg3}$ d) $K_{tocan}$ e) Ne znam
Točan odgovor	ona vrijednost $K_{tocan}$ za koju vrijedi da atribut $K_{slot}$ i vrijednost $K_{tocan}$ čine okvir koncepta $K_x$ , tj. $K_{tocan} \in V_V((K_x, slot, K_{slot}), (K_{slot}, filler, K_{tocan}), (K_x, filler, K_{tocan}) \in PZ')$
Napomena	Odabiru se jedan proizvoljni koncept $K_x$ i jedan njegov proizvoljni atribut $K_{slot}$ .
Primjer	<b>Koja je</b> Kapacitet <b>od</b> Disketa? a) 2 b) 8 c) 1.44MB d) 01 e) Ne Znam

Predložak18:	
Uvjet za generiranje	$K_{slot} \in V_A, K_{filler} \in V_V, \exists (K_{slot}, filler, K_{filler}) \in PZ'$ - odabire se $K_{slot}$ takav da postoji $K_{filler}$ takav da postoji $(K_{slot}, filler, K_{filler})$
Pitanje	Čija je $K_{slot}$ $K_{filler}$ ?
Mogući odgovori	a) $K_{odg1}$ b) $K_{odg2}$ c) $K_{odg3}$ d) $K_{tocan}$ e) Ne znam
Točan odgovor	onaj koncept $K_{tocan}$ čiji okvir čine atribut $K_{slot}$ i vrijednost $K_{filler}$ , tj. $\exists K_{tocan} \in V' \setminus (V_A \cup V_V) ((K_{tocan}, slot, K_{slot}), (K_{slot}, filler, K_{filler}), (K_{tocan}, filler, K_{filler}) \in PZ')$
Napomena	Odabiru se jedan proizvoljni atribut $K_{slot}$ i vrijednost $K_{filler}$ tako da oni čine okvir traženog koncepta u podskupu područnog znanja koje se provjerava.
Primjer	<b>Čija je</b> 2 <b>od</b> Baza? a) Disketa b) Binarni brojevnj sustav c) Oktalni brojevnj sustav d) Dekadski brojevnj sustav e) Ne Znam

#### 4.2.2.2 Ocjenjivanje poznavanja povezanosti

Svako pitanje omogućuje ocjenjivanje poznavanja povezanosti između dva koncepta, kao i poznavanje samih koncepata (Tablica 4.3.). Zato je potrebno odrediti koliko je bodova moguće dobiti za odgovor na pitanje pojedinog predloška. Bez obzira na pitanje, ako je odgovor „Ne znam“, onda je ostvareni broj bodova na tom pitanju jednak 0. Netočan odgovor donosi 0 bodova. Točan odgovor donosi onoliko bodova kolika je kategorija kojoj pripada.

Tablica 4.3. Predložci i koncepti na koje se odnose

Kategorija pitanja	Broj bodova za točno odgovoreno pitanje	Predložak	Uređena trojka područnog znanja na koju se bodovi odnose	Relacija na koju se bodovi odnose
1	bod=1	Predložak1	$(K_x, r, K_y) \vee (K_y, r, K_x)$	$K_x K_y \vee K_y K_x$
		Predložak2	$(K_x, r, K_y) \vee (K_y, r, K_x)$	$K_x K_y \vee K_y K_x$
		Predložak3	$(K_y, r, K_x)$	$K_x K_y$
		Predložak4	$(K_x, r, K_y)$	$K_x K_y$
		Predložak5	$(K_x, r_{tocna}, K_y)$	$K_x K_y$
2	bod=2	Predložak6	$\forall (K_x, r_z, K_y) \in PNZK_x K_y$	$\forall K_x K_y \in G_{PNZK_x K_y}$
		Predložak7	$(K_{tocan}, r, K_x)$	$K_{tocan} K_x$
		Predložak8	$(K_x, r, K_{tocan})$	$K_{tocan} K_x$
3	bod=3	Predložak9	$(K_y, r, K_x)$	$K_x K_y$
		Predložak10	$\forall (K_x, r, K_y), \forall (K_z, r, K_x) \in PZ'$	$\forall K_x K_y, \forall K_z K_x$
		Predložak11	$\forall (K_x, r, K_y) \in PZ'$	$\forall K_x K_y$
		Predložak12	$\forall (K_z, r, K_x) \in PZ'$	$\forall K_z K_x$
		Predložak13	$\forall i (K_{odgai}, r, K_{odgbi})$	$\forall i, K_{odgai} K_{odgbi}$
4	bod=4	Predložak14	$\exists (K_x, r_{tocna}, K_y) \in PZ'$	$K_x K_y$
		Predložak15	$\forall i, j, k (K_{odgai}, r_k, K_{odgbj})$	$\forall i, j, K_{odgai} K_{odgbj}$
		Predložak16	$\exists (K_x, slot, K_{slot})$ $\wedge (K_{slot}, filler, K_{filler})$ $\wedge (K_x, filler, K_{filler}) \in PZ'$	$K_x K_{slot}, K_x K_{filler}, K_{slot} K_{filler}$
		Predložak17	$(K_x, slot, K_{slot}),$ $(K_{slot}, filler, K_{tocan}),$ $(K_x, filler, K_{tocan}) \in PZ'$	$K_x K_{slot}, K_x K_{tocan}, K_{slot} K_{tocan}$
		Predložak18	$(K_{tocan}, slot, K_{slot}),$ $(K_{slot}, filler, K_{filler}),$ $(K_{tocan}, filler, K_{filler}) \in PZ'$	$K_{tocan} K_{slot}, K_{tocan} K_{filler},$ $K_{slot} K_{filler}$

U drugom poglavlju smo definirali bridno-težinsku funkciju  $X_A$  na grafu područnog znanja koje omogućava pridjeljivanje bodova relacijama na koje se određeno pitanje odnosi (Definicija 2.14).

Pošto funkcija  $X_A$  pridjeljuje vrijednost bridovima grafa područnog znanja, nailazimo na problem ako ne postoji brid kojem bi ova funkcija trebala pridijeliti vrijednosti. Ovakve situacije se mogu dogoditi kod pitanja koja su generirana prema predlošcima 1, 2, 3, 4, 6, 9 i 16, odnosno kod pitanja koja za uvjet izvedivosti nemaju navedeno postojanje određene uređene trojke u  $PZ'$ , odnosno brida u  $GPZ'$ . U tim situacijama učenik za dva koncepta  $K_x, K_y$  može tvrditi da su povezani, iako oni to nisu. Takav odgovor interpretiramo na način da kažemo da učenik ima *krivo poimanje* (eng. misconception) i učenik dobiva 0 bodova na tom pitanju. Pošto ne postoji brid  $K_x K_y$ , ne možemo pridijeliti vrijednost funkcije  $X_A$  nečemu što ne postoji. U istim situacijama, ako učenik potvrdi da koncepti nisu povezani, učenik je pokazao znanje i dobiva odgovarajući broj bodova na tom pitanju, ali opet nemamo brid kojem bi funkcija  $X_A$  pridijelila vrijednost.

Ako već postoji pozitivna vrijednost funkcije  $X_A$  za određeni brid  $K_x K_y$ , a dobiveni bodovi na trenutnom pitanju su 0, onda ostavljamo postojeću pozitivnu vrijednost. Za odgovor „Ne znam“ vrijedi  $X_A(K_x K_y) = 0$ , odnosno broj bodova je  $bod=0$ .

Algoritam za određivanje vrijednosti funkcije  $X_A$ , odnosno za interpretiranje i bodovanje odgovora na pitanja generirana prema predlošcima za pitanja, slijedi:

**Algoritam 9. Algoritam za određivanje vrijednosti funkcije  $X_A$** **Predložak1:**

Ako  $\exists(K_x, K_y) \vee \exists(K_y, K_x) \in A'$ , onda

Ako je učenik odgovorio Da, onda  $X_A(K_x, K_y) = bod = 1$

Inače  $X_A(K_x, K_y) = bod = 0$

Inače, Ako je učenik odgovorio Ne, onda

Zaključujemo da učenik nema krivih poimanja, ali ne možemo pridieliti težinu nijednom bridu u grafu područnog znanja, jer koncepti nisu povezani, pa brid ne postoji

$bod = 1$

Inače

učenik ima krivo poimanje da su povezani koncepti koji nisu povezani

$bod = 0$

**Predložak2:**

Ako  $\exists(K_x, K_y) \exists(K_x, r, K_y) \vee \exists(K_y, r, K_x) \in PZ'$ , onda

Ako je učenik odgovorio Da, onda  $X_A(K_x, K_y) = bod = 1$

Inače  $X_A(K_x, K_y) = bod = 0$

Inače, Ako je učenik odgovorio Ne, onda

Zaključujemo da učenik nema krivih poimanja, ali ne možemo pridieliti težinu nijednom bridu u grafu područnog znanja, jer koncepti nisu povezani, pa brid ne postoji

$bod = 1$

Inače

učenik ima krivo poimanje da su povezani koncepti koji nisu povezani

$bod = 0$

**Predložak3:**

Ako  $\exists(K_y, K_x) \in A'$ , onda

Ako je učenik odgovorio Da, onda  $X_A(K_x, K_y) = bod = 1$

Inače  $X_A(K_x, K_y) = bod = 0$

Inače, Ako je učenik odgovorio Ne, onda

Zaključujemo da učenik nema krivih poimanja, ali ne možemo pridieliti težinu nijednom bridu u grafu područnog znanja, jer koncepti nisu povezani, pa brid ne postoji

$bod = 1$

Inače

učenik ima krivo poimanje da su povezani koncepti koji nisu povezani

$bod = 0$

**Predložak4:**

Ako  $\exists(K_x, K_y) \in A'$ , onda

Ako je učenik odgovorio Da, onda  $X_A(K_x, K_y) = bod = 1$

Inače  $X_A(K_x, K_y) = bod = 0$

Inače, Ako je učenik odgovorio Ne, onda

Zaključujemo da učenik nema krivih poimanja, ali ne možemo pridieliti težinu nijednom bridu u grafu područnog znanja, jer koncepti nisu povezani, pa brid ne postoji

$bod = 1$

Inače

učenik ima krivo poimanje da su povezani koncepti koji nisu povezani

$bod = 0$

**Predložak5:**

Ako je učenik točno odgovorio na pitanje (odabrao je za odgovor  $r_{točna}$ ), onda je  $X_A(K_x, K_y) = bod = 1$

Inače  $X_A(K_x, K_y) = bod = 0$

**Predložak6:**

Ako  $\exists SetnjaK_xK_y \subset GPZ', |SetnjaK_xK_y| \geq 2$ , onda

Ako je učenik odgovorio Da, onda  $\forall K_xK_y \in A_{SetnjaK_xK_y}, X_A(K_x, K_y) = bod = 2$

Inače  $\forall K_xK_y \in A_{SetnjaK_xK_y}, X_A(K_x, K_y) = bod = 0$

Inače, Ako je učenik odgovorio Ne, onda

Zaključujemo da učenik nema krivih poimanja, ali ne možemo pridieliti težinu nijednom bridu u grafu područnog znanja, jer koncepti nisu povezani, pa brid ne postoji.

$bod = 2$

Inače

učenik ima krivo poimanje da su povezani koncepti koji nisu povezani

$bod = 0$

**Predložak7:**

Ako je učenik točno odgovorio na pitanje (odabrao je za odgovor  $K_{točan}$ ), onda je  $X_A(K_x, K_{točan}) =$

$bod = 2$

Inače  $X_A(K_x K_{tocan}) = bod = 0$

**Predložak8:**

Ako je učenik točno odgovorio na pitanje (odabrao je za odgovor  $K_{tocan}$ ), onda je  $X_A(K_{tocan} K_x) = bod = 2$

Inače  $X_A(K_{tocan} K_x) = bod = 0$

**Predložak9:**

Ako  $\exists(K_y, K_x) \in A'$ , onda

Ako je učenik odgovorio podkoncept, onda  $X_A(K_y K_x) = bod = 3$

Inače  $X_A(K_y K_x) = bod = 0$

Inače, Ako  $\exists(K_x, K_y) \in A'$ , onda

Ako je učenik odgovorio nadkoncept, onda  $X_A(K_x K_y) = bod = 3$

Inače  $X_A(K_x K_y) = bod = 0$

Inače, Ako je učenik odgovorio nisu povezani

Zaključujemo da učenik nema krivih poimanja, ali ne možemo pridijeliti težinu nijednom bridu u grafu područnog znanja, jer koncepti nisu povezani, pa brid ne postoji.

$bod = 3$

Inače

učenik ima krivo poimanje da su povezani koncepti koji nisu povezani

$bod = 0$

**Predložak10:**

Ako je učenik odgovorio  $nK_x + pK_x$ , onda  $\forall(K_x K_y), \forall(K_z K_x) \in A', X_A(K_x K_y) = bod = 3, X_A(K_z K_x) = bod = 3$

Inače  $\forall(K_x K_y), \forall(K_z K_x) \in A', X_A(K_x K_y) = bod = 0, X_A(K_z K_x) = bod = 0$

**Predložak11:**

Ako je učenik odgovorio  $pK_x$ , onda  $\forall(K_x K_y) \in A', X_A(K_x K_y) = bod = 3$

Inače  $\forall(K_x K_y) \in A', X_A(K_x K_y) = bod = 0$

**Predložak12:**

Ako je učenik odgovorio  $nK_x$ , onda  $\forall(K_z K_x) \in A', X_A(K_z K_x) = bod = 3$

Inače  $\forall(K_z K_x) \in A', X_A(K_z K_x) = bod = 0$

**Predložak13:**

Za one parove rednih brojeva koji su točno navedeni je  $X_A(K_{odgai}, K_{odgbi}) = bod = 3$

Inače  $X_A(K_{odgai}, K_{odgbi}) = bod = 0$

**Predložak14:**

Ako je učenik točno odgovorio na pitanje (naveo je za odgovor  $r_{tocna}$ ), onda je  $X_A(K_x K_y) = bod = 4$

Inače  $X_A(K_x K_y) = bod = 0$

**Predložak15:**

Ako je učenik točno odgovorio na pitanje (naveo je točan redoslijed koncepata), onda je

$X_A(K_{odgj} K_{odgj+1}) = bod = 4$

Inače  $X_A(K_{odgj} K_{odgj+1}) = bod = 0$

**Predložak16:**

Ako  $\exists(K_x, slot, K_{slot}), (K_{slot}, filler, K_{filler}), (K_x, filler, K_{filler}) \in PZ'$ , onda

Ako je učenik odgovorio Da, onda  $X_A(K_x K_{slot}) = X_A(K_x K_{filler}) = X_A(K_{slot} K_{filler}) = bod = 4$

Inače  $X_A(K_x K_{slot}) = X_A(K_x K_{filler}) = X_A(K_{slot} K_{filler}) = bod = 0$

Inače, Ako je učenik odgovorio Ne, onda

Zaključujemo da učenik nema krivih poimanja, ali ne možemo pridijeliti težinu nijednom bridu u grafu područnog znanja, jer koncepti nisu povezani, pa brid ne postoji.

$bod = 4$

Inače

učenik ima krivo poimanje da su povezani koncepti koji zapravo nisu povezani

$bod = 0$

**Predložak17:**

Ako je učenik točno odgovorio na pitanje (odabrao je za odgovor  $K_{tocan}$ ), onda je  $X_A(K_x K_{slot}) =$

$X_A(K_x K_{tocan}) = X_A(K_{slot} K_{tocan}) = bod = 4$

Inače  $X_A(K_x K_{slot}) = X_A(K_x K_{tocan}) = X_A(K_{slot} K_{tocan}) = bod = 0$

**Predložak18:**

Ako je učenik točno odgovorio na pitanje (odabrao je za odgovor  $K_{tocan}$ ), onda je

$X_A(K_{tocan} K_{slot}) = X_A(K_{tocan} K_{filler}) = X_A(K_{slot} K_{filler}) = bod = 4$

Inače  $X_A(K_{tocan} K_{slot}) = X_A(K_{tocan} K_{filler}) = X_A(K_{slot} K_{filler}) = bod = 0$



U drugom poglavlju smo također definirali i vršno-težinsku funkciju  $X_V$  (Definicija 2.15) čija vrijednost  $X_V(K_x)$  predstavlja normiranu sumu vrijednosti funkcije  $X_A$  bridova prema nadkonceptima koncepta  $K_x$  i bridova prema podkonceptima koncepta  $K_x$ .

Rezultati provjeravanja znanja, odnosno vrijednosti težinske funkcije  $X_V$  omogućavaju definiranje pet podskupova koncepata područnog znanja. Svaki podskup obuhvaća koncepte koje učenik poznaje na različitim razinama usvojenosti (Definicija 2.16). Navodimo spomenute skupove i razine znanja prema Bloomovoj taksonomiji koje ih određuju u sljedećoj tablici:

Tablica 4.4. Podskupovi područnog znanja i odgovarajuće razine znanja prema Bloomovoj taksonomiji

Podskup područnog znanja	Razina znanja
$Z_{Razina_0}$	Nema znanja
$Z_{Razina_1}$	Reprodukcija
$Z_{Razina_2}$	Razumijevanje
$Z_{Razina_3}$	Primjena
$Z_{Razina_4}$	Analiza, sinteza i vrednovanje

#### 4.2.2.3 Izvedivost i generiranje testa

Testom provjeravamo kakvo je učenikovo poznavanje povezanosti koncepata koji pripadaju određenom podskupu područnog znanja  $PZ'$ . Pošto se područno znanje sastoji od uređenih trojki koje govore o tome koja dva koncepta su povezana kojom relacijom, podskup područnog znanja  $PZ'$  se sastoji od istih uređenih trojki. Pri generiranju pitanja promatramo i graf podskupa područnog znanja  $GPZ'$  i njegov skup vrhova  $V'$  kao i pripadajući skup bridova  $A'$ .

Spomenuli smo da se pitanje može generirati prema nekom predlošku samo ako su ispunjeni uvjeti izvodivosti pitanja. Dakle, za dva koncepta koji pripadaju skup vrhova  $V'$  ne moraju se generirati pitanja prema svim predlošcima. U drugom poglavlju smo definirali što je to provjerljivi podskup područnog znanja, te kada je element računalom oblikovanog nastavnog sadržaja za provjeravanje znanja izvediv (Definicija 2.13)

Svaki test se dinamički generira u trenutku izvršavanja i to jedino ako se utvrdi da je izvediv. U protivnom se iz podskupa područnog znanja moraju izbaciti uređene trojke čija se povezanost ne može provjeriti.

Prije generiranja testa potrebno je utvrditi kojem stereotipu učenik pripada (definirani su u drugom poglavlju), odnosno njegovu razinu znanja *razina* tako što se odredi koji skup  $Z_{Razina_i}$ ,  $i = 0, \dots, 4$  ima najviše članova (Definicija 2.16). Zatim se  $\forall K_x, K_y \in V'$ , koje učenik poznaje na razini jednakoj ili manjoj od one kojoj on pripada, generira po jedno pitanje prema predlošcima iz težinske kategorije koja je za jedan veća od one kojoj učenik pripada, tj. *razina* + 1 (osim ako je učenik *stručnjak*, tj. *razina* = 4, pa se tada kreće od četvrte težinske kategorije). Ovakvim pristupom odmah možemo utvrditi da li je došlo da napretka u znanju učenika (ako riješi pitanja iz veće težinske kategorije nego što je njegovo prethodno

iskazano znanje). Također ovakav pristup povećava pozitivnu korelaciju između motivacije i rezultata (Pintrich, 1999, prema (Ullrich, 2008)).

Nad konceptima koji su obuhvaćeni s pitanjima na koja učenik nije točno odgovorio, generiraju se pitanja iz manje težinske kategorije (osim ako je nova razina 0, tj. ako su nad njima već bila generirana pitanja iz prve težinske kategorije, pa ne postoji manja težinska kategorija, tada se za te koncepte više ne generiraju pitanja), a nad konceptima koji su obuhvaćeni s pitanjima na koja je učenik točno odgovorio, generiraju se pitanja iz veće težinske kategorije (osim ako je nova razina 4 tj. ako su nad njima već bila generirana pitanja iz četvrte težinske kategorije, pa ne postoji veća težinska kategorija, tada se za te koncepte više ne generiraju pitanja).

Provjeravanje znanja u ulaznom testu koji se generira nad reprezentacijom područnog znanja, započinje s pitanjima iz *treće* kategorije. Nad konceptima koji su obuhvaćeni s pitanjima na koja učenik nije točno odgovorio, generiraju se pitanja iz *druge* težinske kategorije, a nad konceptima koji su obuhvaćeni s pitanjima na koja je učenik točno odgovorio, generiraju se pitanja iz *četvrte* težinske kategorije. Nad konceptima koji su obuhvaćeni s pitanjima iz druge težinske kategorije na koja učenik nije točno odgovorio, generiraju se pitanja iz *prve* težinske kategorije.

Navodimo algoritam za generiranje testa nad podskupom područnog znanja *PZ'* ako je poznata razina znanja *razina*:

<b>Algoritam 10. Algoritam za generiranje testa nad podskupom područnog znanja <i>PZ'</i> i grafom <i>GPZ'</i></b>	
1.	Poznata je vrijednost varijable <i>razina</i> i podskup područnog znanja <i>PZ'</i> (odnosno <i>GPZ'</i> )
2.	Ako je <i>razina</i> = 0, onda se pitanja u testu (u prvoj iteraciji) mogu generirati samo prema predlošcima iz prve težinske kategorije <ol style="list-style-type: none"> <li>a. Sortirajte koncepte iz <i>GPZ'</i> u niz T koristeći algoritam za pretraživanje grafa po dubini (Algoritam 2)</li> <li>b. Uzmite prvi element u nizu T</li> <li>c. Slučajno odaberite jedan od predložaka iz <u>prve</u> težinske kategorije kojem je uvjet za generiranje ispunjen</li> <li>d. <b>Ako</b> ste odabrali predložak i generirali pitanje, onda               <ol style="list-style-type: none"> <li>i. <b>Ako</b> je učenik točno odgovorio na pitanje, onda sve uređene trojke na koje se generirano pitanje odnosi dodajte u podskup područnog znanja <i>PZ'_TO</i> i izbacite ih iz <i>PZ'</i></li> <li>ii. <b>Inače</b> ih dodajte u podskup područnog znanja <i>PZ'_NO</i> i izbacite ih iz <i>PZ'</i></li> </ol> </li> <li>e. <b>Ako</b> je <math> T =1</math>, onda prijedite na <b>točku 6</b></li> <li>f. <b>Inače</b> idite na <b>točku 2.a</b></li> </ol>
3.	Ako je <i>razina</i> = 1, onda se pitanja u testu (u prvoj iteraciji) mogu generirati samo prema predlošcima iz druge težinske kategorije <ol style="list-style-type: none"> <li>a. Sortirajte koncepte iz <i>GPZ'</i> u niz T koristeći algoritam za pretraživanje grafa po dubini (Algoritam 2)</li> <li>b. Uzmite prvi element u nizu T</li> <li>c. Slučajno odaberite jedan od predložaka iz <u>druge</u> težinske kategorije kojem je uvjet za generiranje ispunjen</li> <li>d. <b>Ako</b> ste odabrali predložak i generirali pitanje, onda               <ol style="list-style-type: none"> <li>i. <b>Ako</b> je učenik točno odgovorio na pitanje, onda sve uređene trojke na koje se generirano pitanje odnosi dodajte u podskup područnog znanja <i>PZ'_TO</i> i izbacite ih iz <i>PZ'</i></li> <li>ii. <b>Inače</b> ih dodajte u podskup područnog znanja <i>PZ'_NO</i> i izbacite ih iz <i>PZ'</i></li> </ol> </li> <li>e. <b>Ako</b> je <math> T =1</math>, onda prijedite na <b>točku 6</b></li> <li>f. <b>Inače</b> idite na <b>točku 3.a</b></li> </ol>
4.	Ako je <i>razina</i> = 2, onda se pitanja u testu (u prvoj iteraciji) mogu generirati samo prema predlošcima iz treće težinske kategorije <ol style="list-style-type: none"> <li>a. Generirajte sva moguća pitanja prema predlošku 13.</li> <li>b. <b>Ako</b> ste generirali pitanje, onda               <ol style="list-style-type: none"> <li>i. <b>Ako</b> je učenik točno odgovorio na pitanje, onda sve uređene trojke na koje se generirano pitanje odnosi dodajte u podskup područnog znanja</li> </ol> </li> </ol>

- PZ'\_TO i izbacite ih iz PZ' ,
- ii. **Inače** ih dodajte u podskup područnog znanja PZ'\_NO i izbacite ih iz PZ'
- c. Sortirajte koncepte iz GPZ' u niz T koristeći algoritam za pretraživanje grafa po dubini (Algoritam 2)
- d. Uzmite prvi element u nizu T
- e. Slučajno odaberite jedan od predložaka iz treće težinske kategorije kojem je uvjet za generiranje ispunjen (osim predložka 13)
- f. **Ako** ste odabrali predložak i generirali pitanje, onda
- i. **Ako** je učenik točno odgovorio na pitanje, onda sve uređene trojke na koje se generirano pitanje odnosi dodajte u podskup područnog znanja PZ'\_TO i izbacite ih iz PZ' ,
- ii. **Inače** ih dodajte u podskup područnog znanja PZ'\_NO i izbacite ih iz PZ'
- g. **Ako** je  $|T|=1$ , onda prijedite na **točku 6**
- h. **Inače** idite na **točku 4.c**
5. Ako je *razina* = 3 ili 4, onda se pitanja u testu (u prvoj iteraciji) mogu generirati samo prema predlošcima iz četvrte težinske kategorije
- a. Generirajte sva moguća pitanja prema predlošku 15.
- b. **Ako** ste generirali pitanje, onda
- i. **Ako** je učenik točno odgovorio na pitanje, onda sve uređene trojke na koje se generirano pitanje odnosi dodajte u podskup područnog znanja PZ'\_TO i izbacite ih iz PZ' ,
- ii. **Inače** ih dodajte u podskup područnog znanja PZ'\_NO i izbacite ih iz PZ'
- c. Sortirajte koncepte iz GPZ' u niz T koristeći algoritam za pretraživanje grafa po dubini (Algoritam 2)
- d. Uzmite prvi element u nizu T
- e. Slučajno odaberite jedan od predložaka iz četvrte težinske kategorije kojem je uvjet za generiranje ispunjen (osim predložka 15)
- f. **Ako** ste odabrali predložak i generirali pitanje, onda
- i. **Ako** je učenik točno odgovorio na pitanje, onda sve uređene trojke na koje se generirano pitanje odnosi dodajte u podskup područnog znanja PZ'\_TO i izbacite ih iz PZ' ,
- ii. **Inače** ih dodajte u podskup područnog znanja PZ'\_NO i izbacite ih iz PZ'
- g. **Ako** je  $|T|=1$ , onda prijedite na **točku 6**
- h. **Inače** idite na **točku 5.c**
6. Ako je *razina* = 0, onda
- a. Ako je  $PZ'_TO \neq \emptyset$ , onda
- i. Sortirajte koncepte iz GPZ' u niz T koristeći algoritam za pretraživanje grafa po dubini (Algoritam 2) za  $PZ'=PZ'_TO$
- ii. Uzmite prvi element u nizu T
- iii. Slučajno odaberite jedan od predložaka iz druge težinske kategorije kojem je uvjet za generiranje ispunjen
- iv. **Ako** ste odabrali predložak i generirali pitanje, onda **ako** je učenik točno odgovorio na pitanje, sve uređene trojke na koje se generirano pitanje odnosi dodajte u podskup područnog znanja PZ'\_TO i izbacite ih iz PZ'
- v. **Ako** je  $|T|=1$ , onda prijedite na **točku 7.a**
- vi. **Inače** idite na **točku 6.a.i**
7. Ako je *razina* = 1, onda
- a. Ako je  $PZ'_TO \neq \emptyset$ , onda
- i. Generirajte sva moguća pitanja prema predlošku 13.
- ii. **Ako** ste generirali pitanje, onda **ako** je učenik točno odgovorio na pitanje, onda sve uređene trojke na koje se generirano pitanje odnosi dodajte u podskup područnog znanja PZ'\_TO i izbacite ih iz PZ'
- iii. Sortirajte koncepte iz GPZ' u niz T koristeći algoritam za pretraživanje grafa po dubini (Algoritam 2) za  $PZ'=PZ'_TO$
- iv. Uzmite prvi element u nizu T
- v. Slučajno odaberite jedan od predložaka iz treće težinske kategorije kojem je uvjet za generiranje ispunjen (osim predložka 13)
- vi. **Ako** ste odabrali predložak i generirali pitanje, onda **ako** je učenik točno odgovorio na pitanje, sve uređene trojke na koje se generirano pitanje odnosi dodajte u podskup područnog znanja PZ'\_TO i izbacite ih iz PZ'
- vii. **Ako** je  $|T|=1$ , onda prijedite na **točku 8.a**
- viii. **Inače** idite na **točku 7.a.iii**
- b. Ako je  $PZ'_NO \neq \emptyset$ , onda
- i. Sortirajte koncepte iz GPZ' u niz T koristeći algoritam za pretraživanje grafa po dubini (Algoritam 2) za  $PZ'=PZ'_NO$
- ii. Uzmite prvi element u nizu T
- iii. Slučajno odaberite jedan od predložaka iz prve težinske kategorije kojem je uvjet za generiranje ispunjen
- iv. **Ako** je  $|T|=1$ , onda prijedite na **točku 10**
- v. **Inače** idite na **točku 7.b.i**
8. Ako je *razina* = 2, onda
- a. Ako je  $PZ'_TO \neq \emptyset$ , onda
- i. Generirajte sva moguća pitanja prema predlošku 15.
- ii. **Ako** ste generirali pitanje, onda **ako** je učenik točno odgovorio na pitanje, onda sve uređene trojke na koje se generirano pitanje odnosi dodajte u

- podskup područnog znanja  $PZ'_{TO}$  i izbacite ih iz  $PZ'$
- iii. Sortirajte koncepte iz  $GPZ'$  u niz  $T$  koristeći algoritam za pretraživanje grafa po dubini (Algoritam 2) za  $PZ'=PZ'_{TO}$
  - iv. Uzmite prvi element u nizu  $T$
  - v. Slučajno odaberite jedan od predložaka iz četvrte težinske kategorije kojem je uvjet za generiranje ispunjen (osim predloška 15)
  - vi. **Ako** je  $|T|=1$ , onda prijedite na **točku 10**
  - vii. **Inače** idite na **točku 8.a.iii**
- b. Ako je  $PZ'_{NO} \neq \emptyset$ , onda
- i. Sortirajte koncepte iz  $GPZ'$  u niz  $T$  koristeći algoritam za pretraživanje grafa po dubini (Algoritam 2) za  $PZ'=PZ'_{NO}$
  - ii. Uzmite prvi element u nizu  $T$
  - iii. Slučajno odaberite jedan od predložaka iz druge težinske kategorije kojem je uvjet za generiranje ispunjen
  - iv. **Ako** ste odabrali predložak i generirali pitanje, onda **ako** je učenik netočno odgovorio na pitanje, sve uređene trojke na koje se generirano pitanje odnosi dodajte u podskup područnog znanja  $PZ'_{NO}$  i izbacite ih iz  $PZ'$
  - v. **Ako** je  $|T|=1$ , onda prijedite na **točku 7.b**
  - vi. **Inače** idite na **točku 8.b.i**
9. Ako je *razina* = 3 ili 4, onda
- a. Ako je  $PZ'_{NO} \neq \emptyset$ , onda
    - i. Generirajte sva moguća pitanja prema predlošku 13.
    - ii. **Ako** ste generirali pitanje, onda **ako** je učenik točno odgovorio na pitanje, onda sve uređene trojke na koje se generirano pitanje odnosi dodajte u podskup područnog znanja  $PZ'_{NO}$  i izbacite ih iz  $PZ'$
    - iii. Sortirajte koncepte iz  $GPZ'$  u niz  $T$  koristeći algoritam za pretraživanje grafa po dubini (Algoritam 2) za  $PZ'=PZ'_{NO}$
    - iv. Uzmite prvi element u nizu  $T$
    - v. Slučajno odaberite jedan od predložaka iz treće težinske kategorije kojem je uvjet za generiranje ispunjen (osim predloška 13)
    - vi. **Ako** ste odabrali predložak i generirali pitanje, onda **ako** je učenik netočno odgovorio na pitanje, sve uređene trojke na koje se generirano pitanje odnosi dodajte u podskup područnog znanja  $PZ'_{NO}$  i izbacite ih iz  $PZ'$
    - vii. **Ako** je  $|T|=1$ , onda prijedite na **točku 8.b**
    - viii. **Inače** idite na **točku 10**
10. Kraj

Proces provjeravanja znanja učenika prikazan je strojem s konačnim brojem stanja na Slika 2.14.

Vidimo da se stereotip učenika može promijeniti samo nakon provedenog testiranja znanja. Određivanje stereotipa nakon testa se radi na način da se pronade koji skup  $Z_{Razina_i}$ ,  $i = 0, \dots, 4$  ima najviše članova (Definicija 2.16, Definicija 2.17).

Tablica 4.5. Stereotipovi određeni skupom koji ima najviše elemenata

Stereotip prema znanju	Skup
Novak	$Z_{Razina_0}$
Početak	$Z_{Razina_1}$
Osrednji	$Z_{Razina_2}$
Napredni	$Z_{Razina_3}$
Stručnjak	$Z_{Razina_4}$

Proces učenja i poučavanja se prilagođava dobivenom stereotipu načinom odabira, nizanja i prezentiranja elemenata računalom oblikovanog nastavnog sadržaja.

### 4.2.3 Odabir i nizanje elemenata računalom oblikovanog nastavnog sadržaja prilagođeno stereotipovima

Nakon što smo definirali stereotipove prema znanju (drugo poglavlje) i način na koji ćemo odrediti kojem stereotipu učenik pripada (Definicija 2.17), potrebno je definirati način na koji će se određenom stereotipu odabrati i nizati elementi računalom oblikovanog nastavnog sadržaja.

*Odabir elemenata* računalom oblikovanog nastavnog sadržaja se vrši prema razini elemenata. Razina elementa nam određuje količinu znanja koju će učenik učiti u jednom ciklusu učenja i poučavanja. U drugom poglavlju u Tablica 2.8 smo naveli kojem stereotipu odgovara koja razina elementa.

Nakon što se za određeni stereotip odabrali svi elementi računalom oblikovanog nastavnog sadržaja određene razine, *nizanje elemenata* je dodavanje sortiranih elemenata u NS i dodavanje elemenata za provjeravanje znanja.

*Sortiranje* elemenata računalom oblikovanog nastavnog sadržaja se vrši prema rangu elemenata: prvo prema oznaci cjeline, a onda rastući prema udaljenosti početaka elemenata od korijena cjeline, zatim prema razini elementa, te na kraju abecedno prema nazivu korijena. Ovim definiramo uređaj na skupu NS za određeni stereotip.

Nakon dodavanja sortiranih elemenata u NS, potrebno definirati kada će učenik *provjeravati* svoje znanje u ovisnosti o stereotipu kojem pripada. Za sve stereotipove je zajedničko da zadnji element računalom oblikovanog nastavnog sadržaja mora biti test. Uvažavajući razinu elementa računalom oblikovanog nastavnog sadržaja koja je pridijeljena kojem stereotipu, u drugom poglavlju u Tablica 2.9 smo naveli pravila za dodavanje elemenata za provjeravanje znanja. Važno je napomenuti da što je niža razina znanja koju učenik posjeduje, to će se njegovo znanje provjeravati češće nad manjim podskupom područnog znanja. Kod učenika koji posjeduju više razine znanja, provjeravanje znanja se vrši rjeđe nad većim podskupom područnog znanja.

U skladu s tim, navodimo algoritme za odabir i nizanje (sortiranje i dodavanje testova) elemenata računalom oblikovanog nastavnog sadržaja za svaki od pet stereotipova. Algoritmi su prikazani u dvije razine detalja: prva razina daje tekstualni opis koraka u algoritmu, a druga razina povezuje te korake s algoritamskim strukturama.

#### **Stereotip *novak***

Ako učenik pripada stereotipu *novak*, onda će mu se generirati NS prema sljedećem algoritmu:

**Algoritam 11. Novak – prva razina**

1. Svaku nastavnu jedinicu dodaj u NS, osim ako je nastavna jedinica s istim korijenom već usvojena na razini 1, 2, 3 ili 4.
2. Sortiraj sve nastavne jedinice u NS prema njihovom rangu (prvo prema oznaci cjeline, a onda rastući prema udaljenosti početaka nastavnih jedinica od korijena cjeline, te na kraju abecedno prema nazivu korijena). Ovim definiramo uređaj na skupu NS za stereotip *novak*.
3. Između svake dvije nastavne jedinice u nizu koje imaju različite oznake cjelina stavi test.
4. Između svake dvije nastavne jedinice u nizu koje imaju različite udaljenosti od korijena stavi test, osim ako bi testu prethodila samo jedna nastavna jedinica ili ako bi iza testa ostala samo jedna nastavna jedinica.
5. Nakon tri nastavne jedinice u nizu koja nisu test dodaj test, osim ako bi testu prethodila samo jedna nastavna jedinica ili ako bi iza testa ostala samo jedna nastavna jedinica.
6. Na kraj računalom oblikovanog nastavnog sadržaja stavi test.

**Algoritam 12. Novak – druga razina**

```

a= cijeli dio(broj nastavnih jedinica*1.3) + broj cjelina
DIM NS(a) = "", j=0
for each  $C_i$  /*cjeline su sortirane prema rangu
  for each  $NJ_{k_x}, R_{NJ_{k_x}} = (i, y, 3)$ 
    j=j+1
    NS(j) =  $NJ_{k_x}$ 
  end for
  j=j+1
  NS(j)=test/*dodavanje testova na kraj cjelina
end for
for each k=1 to j-1 /*sortiranje elemenata po udaljenosti unutar cjelina
  if NS(k)=test then k=k+1
  for each m=k+1 to j
    if NS(m)=test then m=m+1
    ck=oznaka cjeline k-tog elementa
    cm=oznaka cjeline m-tog elementa
    uk=y1 /* $R_{NS(k)} = (i, y1, 3)$  - udaljenost od korijena k-tog elementa
    um=y2 /* $R_{NS(m)} = (i, y2, 3)$  - udaljenost od korijena m-tog elementa
    if ck=cm and uk>um then
      temp=NS(k)
      NS(k)=NS(m)
      NS(m)=temp
    end if
  end for
end for
for each k=2 to j /*dodavanje testova kod promjena udaljenosti od korijena
  if NS(k)=test or NS(k+1)=test then k=k+2
  c1=oznaka cjeline k-tog elementa
  c2=oznaka cjeline k+1-tog elementa
  u1=y1 /* $R_{NS(k)} = (i, y1, 3)$  - udaljenost od korijena k-tog elementa
  u2=y2 /* $R_{NS(k+1)} = (i, y2, 3)$  - udaljenost od korijena k+1-tog elementa
  if c1=c2 and u1<u2 then
    if NS(k-1)<>test and NS(k+2)<>test then /*da ne bi
      testu prethodila samo jedna nastavna jedinica i da iza testa ne bi
      ostala samo jedna nastavna jedinica
      for m=j+1 to k+2 STEP -1 /*sve elemente iza k+1
        pomičemo za jedan unaprijed
        NS(m)=NS(m-1)
      end for
      NS(k+1)=test /*ubacujemo test na k+1 poziciju
      j=j+1
    end if
  end if
end for
br=0
for each k=1 to j /*dodavanje testova nakon tri nastavne cjeline
  if NS(k)<>test and NS(k+1)<>test and NS(k+2)<>test then
    if NS(k+3)<>test and NS(k+4)<>test /*da ne
      bi iza testa ostala samo jedna nastavna jedinica
      for m=j+1 to k+4 STEP -1 /*sve elemente iza k+3-tog
        pomičemo za jedan unaprijed
        NS(m)=NS(m-1)

```

```

                end for
                NS(k+3)=test /*ubacujemo test na k+3 poziciju
                j=j+1
            end if
        end if
    end for

```

### Stereotip *početnik*

Ako učenik pripada stereotipu *početnik*, onda će mu se generirati NS prema sljedećem algoritmu:

#### Algoritam 13. Početnik – prva razina

1. Svaku nastavnu jedinicu dodaj u NS, osim ako je nastavna jedinica s istim korijenom već usvojena na razini 1, 2, 3 ili 4.
2. Sortiraj sve nastavne jedinice u NS prema njihovom rangu (prvo prema oznaci cjeline, a onda rastući prema udaljenosti početaka nastavnih jedinica od korijena cjeline, te na kraju abecedno prema nazivu korijena). Ovim definiramo uređaj na skupu NS za stereotip *početnik*.
3. Između svake dvije nastavne jedinice u nizu koje imaju različite oznake cjelina stavi test, ali jedino ako se ispred zadnje nastavne jedinice u cjelini nalaze još barem dvije nastavne jedinice u nizu.
4. Između svake dvije nastavne jedinice u nizu koje imaju različite udaljenosti od korijena stavi test, osim ako bi testu prethodile samo dvije nastavne jedinice ili ako bi iza testa ostale samo dvije nastavne jedinice
5. Na kraj računalom oblikovanog nastavnog sadržaja stavi test.

#### Algoritam 14. Početnik – druga razina

```

a= cijeli dio(broj nastavnih jedinica*1.3) + broj cjelina
DIM NS(a) = "" , j=0
for each  $C_i$  /*cjeline su sortirane prema rangu
    br=0
    for each  $NJ_{K_x}, R_{NJ_{K_x}} = (i,y,3)$ 
        j=j+1
        br=br+1
        NS(j) =  $NJ_{K_x}$ 
    end for
    if br>3 then
        j=j+1
        NS(j)=test/*dodavanje testova na kraj cjelina, ako u cjelini
            ima više od tri nastavne jedinice
    end if
end for
if NS(j)<> test then
    NS(j+1)=test /*zadnji element mora biti test
    j=j+1
end if
for each k=1 to j-1 /*sortiranje elemenata po udaljenosti unutar cjelina
    if NS(k)=test then k=k+1
    for each m=k+1 to j
        if NS(m)=test then m=m+1
        ck=oznaka cjeline k-tog elementa
        cm=oznaka cjeline m-tog elementa
        uk=y1 /* $R_{NS(k)} = (i,y1,3)$  - udaljenost od korijena k-tog elementa
        um=y2 /* $R_{NS(m)} = (i,y2,3)$  - udaljenost od korijena m-tog elementa
        if ck=cm and uk>um then
            temp=NS(k)
            NS(k)=NS(m)
            NS(m)=temp
        end if
    end for
end for
for each k=3 to j /*dodavanje testova kod promjena udaljenosti od korijena
    if NS(k)=test or NS(k+1)=test then k=k+2
    c1=oznaka cjeline k-tog elementa
    c2=oznaka cjeline k+1-tog elementa
    u1=y1 /* $R_{NS(k)} = (i,y1,3)$  - udaljenost od korijena k-tog elementa
    u2=y2 /* $R_{NS(k+1)} = (i,y2,3)$  - udaljenost od korijena k+1-tog elementa

```

```

    if c1=c2 and u1<u2 then
      if NS(k-2)<>test and NS(k-1)<>test and NS(k+2)<>test and NS(k+3)<>test
        then /*da ne bi testu prethodile samo dvije
              nastavne jedinica i da iza testa ne bi ostale samo dvije
              nastavne jedinica
              for m=j+1 to k+2 STEP -1 /*sve elemente iza k+1
                pomičemo za jedan unaprijed
                NS(m)=NS(m-1)
              end for
              NS(k+1)=test /*ubacujemo test na k+1 poziciju
              j=j+1
            end if
          end if
        end for

```

## Stereotip *osrednji*

Ako učenik pripada stereotipu *osrednji*, onda će mu se generirati NS prema sljedećem algoritmu:

### Algoritam 15. Osrednji – prva razina

1. Svaku nastavnu temu dodaj u NS, osim ako je nastavna tema s istim korijenom već usvojena na razini 2, 3 ili 4. Ako nema više tema, a postoje koncepti koji se ne nalaze u SK, onda dodaj po barem jednu nastavnu jedinicu koja ih sadrži.
2. Sortiraj sve nastavne jedinice u NS prema njihovom rangu (prvo prema oznaci cjeline, a onda rastući prema udaljenosti početaka elemenata od korijena cjeline, zatim prema vrsti elementa, te na kraju abecedno prema nazivu korijena). Ovim definiramo uređaj na skupu NS za stereotip *osrednji*.
3. Između svaka dva elementa u nizu koja imaju različite oznake cjelina stavi test, ali jedino ako se ispred zadnjeg elementa u cjelini nalaze još barem dva elementa u nizu koja nisu test.
4. Između svaka dva elementa u nizu koja imaju različite udaljenosti od korijena stavi test, osim ako bi testu prethodila samo dva elementa koja nisu test ili ako bi iza testa ostale samo dva elementa koja nisu test.
5. Nakon tri elementa u nizu koja nisu test dodaj test, osim ako bi testu prethodila samo jedna nastavna tema ili jedinica ili ako bi iza testa ostala samo jedna nastavna tema ili jedinica.
6. Na kraj računalom oblikovanog nastavnog sadržaja stavi test.

### Algoritam 16. Osrednji – druga razina

```

a= cijeli dio(broj nastavnih jedinica*1.3) + broj cjelina
DIMNS(a) = "" , j=0
for each  $C_i$  /*cjeline su sortirane prema rangu
  br=0
  for each  $NT_{K_x}, R_{NT_{K_x}} = (i, y, 2)$ 
    j=j+1
    br=br+1
    NS(j) =  $NT_{K_x}$ 
  end for
  while  $V_{C_i} \cap SK \neq \emptyset$  /*dodavanje jedinica
    for each  $NJ_{K_x}, R_{NJ_{K_x}} = (i, y, 3), V_{NJ_{K_x}} \cap SK \neq \emptyset$ 
      j=j+1
      br=br+1
      NS(j) =  $NJ_{K_x}$ 
    end for
  end while
  if br>3 then
    j=j+1
    NS(j)=test/*dodavanje testova na kraj cjelina, ako u cjelini
    ima više od tri nastavne teme ili jedinice
  end if
end for
if NS(j)<> test then
  NS(j+1)=test /*zadnji element mora biti test
  j=j+1
end if

```



```

for each k=1 to j-1 /*sortiranje elemenata po udaljenosti unutar cjelina
  if NS(k)=test then k=k+1
  for each m=k+1 to j
    if NS(m)=test then m=m+1
    ck=oznaka cjeline k-tog elementa
    cm=oznaka cjeline m-tog elementa
    uk=y1 /* $R_{NS(k)} = (i, y1, v1)$  -udaljenost od korijena k-tog elementa
    um=y2 /* $R_{NS(m)} = (i, y2, v2)$  -udaljenost od korijena m-tog elementa
    if ck=cm and uk>um then
      temp=NS(k)
      NS(k)=NS(m)
      NS(m)=temp
    end if
  end for
end for
for each k=3 to j /*dodavanje testova kod promjena udaljenosti od korijena
  if NS(k)=test or NS(k+1)=test then k=k+2
  c1=oznaka cjeline k-tog elementa
  c2=oznaka cjeline k+1-tog elementa
  u1=y1 /* $R_{NS(k)} = (i, y1, v1)$  - udaljenost od korijena k-tog elementa
  u2=y2 /* $R_{NS(k+1)} = (i, y2, v2)$  - udaljenost od korijena k+1-tog elementa
  if c1=c2 and u1<u2 then
    if NS(k-2)<>test and NS(k-1)<>test and NS(k+2)<>test and NS(k+3)<>test
    then /*da ne bi testu prethodile samo dvije nastavne teme ili jedinice i
      da iza testa ne bi ostale samo dvije nastavne teme ili jedinice
      for m=j+1 to k+2 STEP -1 /*sve elemente iza k+1
        pomičemo za jedan unaprijed
        NS(m)=NS(m-1)
      end for
      NS(k+1)=test /*ubacujemo test na k+1 poziciju
      j=j+1
    end if
  end if
end for
br=0
for each k=1 to j /*dodavanje testova nakon tri nastavne cjeline
  if NS(k)<>test and NS(k+1)<>test and NS(k+2)<>test then
    if NS(k+3)<>test and NS(k+4)<>test /*da ne bi iza testa ostala samo
      jedna nastavna tema ili jedinica
      for m=j+1 to k+4 STEP -1 /*sve elemente iza k+3-tog
        pomičemo za jedan unaprijed
        NS(m)=NS(m-1)
      end for
      NS(k+3)=test /*ubacujemo test na k+3 poziciju
      j=j+1
    end if
  end if
end for
end for

```

## Stereotip *napredni*

Ako učenik pripada stereotipu *napredni*, onda će mu se generirati NS prema sljedećem algoritmu:

### Algoritam 17. Napredni – prva razina

1. Svaku nastavnu temu dodaj u NS, osim ako je nastavna tema s istim korijenom već usvojena na razini 3 ili 4. Ako nema više tema, a postoje koncepti koji se ne nalaze u SK, onda dodaj po barem jednu nastavnu jedinicu koja ih sadrži.
2. Sortiraj sve nastavne jedinice u NS prema njihovom rangu (prvo prema oznaci cjeline, a onda rastući prema udaljenosti početaka elemenata od korijena cjeline, zatim prema vrsti elementa, te na kraju abecedno prema nazivu korijena). Ovim definiramo uređaj na skupu NS za stereotip *napredni*.
3. Između svaka dva elementa u nizu koja imaju različite oznake cjelina stavi test, ali jedino ako se ispred zadnjeg elementa u cjelini nalaze još barem dva elementa u nizu koja nisu test.
4. Između svaka dva elementa u nizu koja imaju različite udaljenosti od korijena stavi test, osim ako bi testu prethodila samo dva elementa koja nisu test ili ako bi iza testa ostale samo dva elementa koja nisu test.
5. Na kraj računalom oblikovanog nastavnog sadržaja stavi test.

**Algoritam 18. Napredni – druga razina**

```

a= cijeli dio(broj nastavnih jedinica*1.3) + broj cjelina
DIM NS(a) = ""
j=0
for each  $C_i$  /*cjeline su sortirane prema rangu
  br=0
  for each  $NT_{K_x}, R_{NT_{K_x}} = (i, y, 2)$ 
    j=j+1
    br=br+1
    NS(j) =  $NT_{K_x}$ 
  end for
  while  $V_{C_i} \cap SK \neq \emptyset$  /*dodavanje jedinica
    for each  $NJ_{K_x}, R_{NJ_{K_x}} = (i, y, 3), V_{NJ_{K_x}} \cap SK \neq \emptyset$ 
      j=j+1
      br=br+1
      NS(j) =  $NJ_{K_x}$ 
    end for
  end while
  if br>3 then
    j=j+1
    NS(j)=test/*dodavanje testova na kraj cjelina, ako u cjelini
    ima više od tri nastavne teme ili jedinice
  end if
end for
if NS(j)<> test then
  NS(j+1)=test /*zadnji element mora biti test
  j=j+1
end if
for each k=1 to j-1 /*sortiranje elemenata po udaljenosti unutar cjelina
  if NS(k)=test then k=k+1
  for each m=k+1 to j
    if NS(m)=test then m=m+1
    ck=oznaka cjeline k-tog elementa
    cm=oznaka cjeline m-tog elementa
    uk=y1 /* $R_{NS(k)} = (i, y1, v1)$  -udaljenost od korijena k-tog elementa
    um=y2 /* $R_{NS(m)} = (i, y2, v2)$  -udaljenost od korijena m-tog elementa
    if ck=cm and uk>um then
      temp=NS(k)
      NS(k)=NS(m)
      NS(m)=temp
    end if
  end for
end for
for each k=3 to j /*dodavanje testova kod promjena udaljenosti od korijena
  if NS(k)=test or NS(k+1)=test then k=k+2
  c1=oznaka cjeline k-tog elementa
  c2=oznaka cjeline k+1-tog elementa
  u1=y1 /* $R_{NS(k)} = (i, y1, v1)$  - udaljenost od korijena k-tog elementa
  u2=y2 /* $R_{NS(k+1)} = (i, y2, v2)$  - udaljenost od korijena k+1-tog elementa
  if c1=c2 and u1<u2 then
    if NS(k-2)<>test and NS(k-1)<>test and NS(k+2)<>test and NS(k+3)<>test
    then /*da ne bi testu prethodile samo dvije
      nastavne teme ili jedinice i da iza testa ne bi ostale
      samo dvije nastavne teme ili jedinice
      for m=j+1 to k+2 STEP -1 /*sve elemente iza k+1
        pomičemo za jedan unaprijed
        NS(m)=NS(m-1)
      end for
      NS(k+1)=test /*ubacujemo test na k+1 poziciju
      j=j+1
    end if
  end if
end for
end if
end for

```

## Stereotip *stručnjak*

Ako učenik pripada stereotipu *stručnjak*, onda će mu se generirati NS prema sljedećem algoritmu:

### Algoritam 19. Stručnjak – prva razina

1. Svaku nastavnu cjelinu dodaj u NS, osim ako je nastavna cjelina s istim korijenom već usvojena na razini 4. Ako u cjelini nema više nastavnih cjelina, a u cjelini još postoje koncepti koji se ne nalaze u SK, onda dodaj po barem jednu nastavnu temu koja ih sadrži. Ako u cjelini nema više tema, a u cjelini još postoje koncepti koji se ne nalaze u SK, onda dodaj po barem jednu nastavnu jedinicu koja ih sadrži.
2. Sortiraj sve nastavne jedinice u NS prema njihovom rangu (prvo prema oznaci cjeline, a onda rastući prema udaljenosti početaka elemenata od korijena cjeline, zatim prema vrsti elementa, te na kraju abecedno prema nazivu korijena). Ovim definiramo uređaj na skupu NS za stereotip *stručnjak*.
3. Na kraj računalom oblikovanog nastavnog sadržaja stavi test.

### Algoritam 20. Stručnjak – druga razina

```

a= cijeli dio(broj nastavnih jedinica*1.3) + broj cjelina
DIMNS(a) = "" , j=0
for each  $C_i$  /*cjeline su sortirane prema rangu
  br=0
  for each  $NC_{K_x}, R_{NC_{K_x}} = (i, y, 1)$ 
    j=j+1
    br=br+1
     $NS(j) = NC_{K_x}$ 
  end for
  while  $V_{C_i} \cap SK \neq \theta$  /*dodavanje tema
    for each  $NT_{K_x}, R_{NT_{K_x}} = (i, y, 2), V_{NT_{K_x}} \cap SK \neq \theta$ 
      j=j+1
      br=br+1
       $NS(j) = NT_{K_x}$ 
    end for
  end while
  while  $V_{C_i} \cap SK \neq \theta$  /*dodavanje jedinica
    for each  $NJ_{K_x}, R_{NJ_{K_x}} = (i, y, 3), V_{NJ_{K_x}} \cap SK \neq \theta$ 
      j=j+1
      br=br+1
       $NS(j) = NJ_{K_x}$ 
    end for
  end while
end for
NS(j+1)=test /*zadnji element mora biti test
j=j+1
for each k=1 to j-1 /*sortiranje elemenata po udaljenosti unutar cjelina
  if NS(k)=test then k=k+1
  for each m=k+1 to j
    if NS(m)=test then m=m+1
    ck=oznaka cjeline k-tog elementa
    cm=oznaka cjeline m-tog elementa
    uk=y1 /* $R_{NS(k)} = (i, y1, v1)$  -udaljenost od korijena k-tog elementa
    um=y2 /* $R_{NS(m)} = (i, y2, v2)$  -udaljenost od korijena m-tog elementa
    if ck=cm and uk>um then
      temp=NS(k)
      NS(k) =NS(m)
      NS(m) =temp
    end if
  end for
end for
end for

```

## 4.2.4 Prezentiranje elemenata računalom oblikovanog nastavnog sadržaja prilagođeno stereotipovima

Nakon što smo definirali način na koji ćemo odrediti kojem stereotipu učenik pripada, način na koji će se određenom stereotipu odabrati i nizati elementi računalom oblikovanog nastavnog sadržaja, potrebno je definirati i način prezentiranja elemenata računalom oblikovanog nastavnog sadržaja koje je prilagođeno stereotipovima.

Prezentiranje elemenata računalom oblikovanog nastavnog sadržaja se vrši s aspekta Bloomove taksonomije. U drugom poglavlju smo u Tablica 2.10 odredili koja razina znanja će se prezentirati kojem stereotipu. Stereotipovima osrednji, napredni i stručnjak znanje prezentira na istoj razini koja je njihova, dok se stereotipovima novak i početnik znanje prezentira na za jedan višoj razini od one koju oni posjeduju jer se zahtijeva njihov što brži pomak prema barem osrednjem stereotipu.

S obzirom da se prezentiranje elemenata računalom oblikovanog nastavnog sadržaja promatra s dva aspekta: prema razini elementa, te prema Bloomovoj, u Tablica 2.11 je navedeno koja razina elementa i koja razina znanja će se prezentirati određenom stereotipu.

Način na koji će se znanje predstaviti učeniku u procesu učenja je definirano predlošcima za generiranje tvrdnji koje učenik mora usvojiti. Pošto smo u poglavlju 4.2.2 definirali četiri težinske kategorije pitanja koje odgovaraju Bloomovoj taksonomiji znanja, definirat ćemo i četiri težinske kategorije tvrdnji koje također odgovaraju Bloomovoj taksonomiji znanja. Svaka težinska kategorija prezentira određenu razinu znanja: prva težinska kategorija sadrži predloške koji prezentiraju znanje na razini reprodukcije, druga na razini razumijevanja, treća na razini primjene, a četvrta na razini analize, sinteze i procjene

### 4.2.4.1 Predlošci tvrdnji za prezentiranje elemenata računalom oblikovanog nastavnog sadržaja

Učenje i poučavanje se vrši prezentiranjem tvrdnji koje se stvaraju prema predlošcima. Da bi se odredilo kojim predloškom poučavamo koju kategoriju znanja, navodimo predloške tvrdnji prema težinskim kategorijama. Ne može se svakim predloškom poučavati povezanosti bilo kojeg podskupa koncepata, već je kod svakog predloška naveden i uvjet koji mora biti ispunjen da bi se mogla generirati tvrdnja prema tom predlošku.

#### Prva težinska kategorija

Predložak1:	
Uvjet	$\exists(K_y, r, K_x) \in PZ'$
Tekst	$K_y$ r $K_x$ . $K_x$ je podkoncept od $K_y$ .
Napomena	Odabiru se dva proizvoljna koncepta koja moraju biti neposredno povezana. Važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta. Također, relacija ne smije biti <i>slot</i> ili <i>filler</i> . Također, koncept $K_x$ ne smije biti korijen, dok koncept $K_y$ ne smije biti list. Razlog tomu je uklanjanje očiglednosti ukoliko bi se provjeravalo da li korijen može biti podkoncept ili da li list ima podkoncept.
Primjer	Računalni sustav se sastoji od Programaska podrška. Programaska podrška <b>je podkoncept od</b> Računalni sustav.

Predložak2:	
Uvjet	$\exists(K_x, r, K_y) \in PZ'$
Tekst	$K_x r K_y$ . $K_x$ je nadkoncept od $K_y$ .
Napomena	Odabiru se dva proizvoljna koncepta koja moraju biti neposredno povezana. Važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta. Također, relacija ne smije biti <i>slot</i> ili <i>filler</i> . Također, koncept $K_x$ ne smije biti list, dok koncept $K_y$ ne smije biti korijen. Razlog tomu je uklanjanje očiglednosti ukoliko bi se provjeravalo da li list može biti nadkoncept ili da li korijen ima nadkoncept.
Primjer	Računalni sustav izvršava Temeljna funkcija računala. Računalni sustav <b>je nadkoncept od</b> Temeljna funkcija računala.

### Druga težinska kategorija

Predložak3:	
Uvjet	$\exists \text{Setnja}_{K_x K_y} \subset GPZ', l_{K_x K_y} > 1$
Tekst	$K_x$ i $K_y$ su posredno povezani jer postoji put $\text{Setnja}_{K_x K_y}$
Napomena	Odabiru se dva proizvoljna koncepta koja ne smiju biti neposredno povezana. Također je važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta. Šetnju $\text{Setnja}_{K_x K_y}$ prikazati kao niz konceptata odvojenih strelicama, bez relacija.
Primjer	Računalni sustav <u>i</u> Izlazna jedinica <b>su posredno povezani jer postoji put</b> Računalni sustav $\rightarrow$ Tehnička podrška $\rightarrow$ Izlazna jedinica.

Predložak4:	
Uvjet	$\exists(K_x, r, K_{y_1}), \dots, (K_x, r, K_{y_n}) \in PZ', n \geq 2$
Tekst	$K_x r: K_{y_1}, \dots, K_{y_n}$ . Ovo su podkoncepti od $K_x$ .
Napomena	Odabire se jedan proizvoljni koncept $K_x$ koji ne smije biti atribut ili vrijednost nekog drugog koncepta. Također treba imati podkoncepte $K_{y_1}, \dots, K_{y_n}$ s kojima je neposredno povezan zadanom relacijom u podskupu područnog znanja koje se provjerava koji ne smije biti atribut ili vrijednost. Ako je $n=1$ , onda napravi tvrdnju prema Predlošku 7.
Primjer	Računalni sustav se sastoji od Programaska podrška, Tehnička podrška. <b>Ovo su podkoncepti od</b> Računalni sustav.

Predložak5:	
Uvjet	$\exists(K_{y_1}, r, K_x), \dots, (K_{y_n}, r, K_x) \in PZ', n \geq 2$
Tekst	$K_{y_1}, \dots, K_{y_n} r K_x$ . Ovo su nadkoncepti od $K_x$ .
Napomena	Odabire se jedan proizvoljni koncept $K_x$ koji ne smije biti atribut ili vrijednost nekog drugog koncepta. Također treba imati nadkoncepte $K_{y_1}, \dots, K_{y_n}$ s kojima je neposredno povezan zadanom relacijom u podskupu područnog znanja koje se provjerava koji ne smije biti atribut ili vrijednost. Ako je $n=1$ , onda napravi tvrdnju prema Predlošku 6.
Primjer	Programski jezik ima podvrstu Programski jezik niske razine, Programski jezik visoke razine. <b>Ovo su nadkoncepti od</b> Programski jezik.

Predložak6:	
Uvjet	$\exists(K_y, r, K_x) \in PZ'$
Tekst	$K_y r K_x$ . $K_x$ ima nadkoncept $K_y$ .
Napomena	Odabiru se dva proizvoljna koncepta koja moraju biti neposredno povezana. Važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta. Također, relacija ne smije biti <i>slot</i> ili <i>filler</i> . Također, koncept $K_x$ ne smije biti korijen, dok koncept $K_y$ ne smije biti list. Razlog tomu je uklanjanje očiglednosti ukoliko bi se provjeravalo da li korijen može biti podkoncept ili da li list ima podkoncept.
Primjer	Računalni sustav izvršava Temeljna funkcija računala. Temeljna funkcija računala <b>ima nadkoncept</b> Računalni sustav.

Predložak7:	
Uvjet	$\exists (K_x, r, K_y) \in PZ'$
Tekst	$K_x r K_y$ . $K_x$ ima podkoncept $K_y$ .
Napomena	Odabiru se dva proizvoljna koncepta koja moraju biti neposredno povezana. Važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta. Također, relacija ne smije biti <i>slot</i> ili <i>filler</i> . Također, koncept $K_x$ ne smije biti list, dok koncept $K_y$ ne smije biti korijen. Razlog tomu je uklanjanje očiglednosti ukoliko bi se provjeravalo da li list može biti nadkoncept ili da li korijen ima nadkoncept.
Primjer	Logička operacija ima primjerak Konjunkcija. Logička operacija <b>ima podkoncept</b> Konjunkcija.

### Treća težinska kategorija

Predložak8:	
Uvjet	$K_x \in V' \setminus (V_A \cup V_V)$ , $K_i \in NadK_x \cup PodK_x$
Tekst	$K_x$ je neposredno povezan s $nK_x + pK_x$ koncepta. Ti koncepti su: $K_1, \dots, K_n$ .
Napomena	Odabire se jedan proizvoljni koncept koji ne smije biti atribut ili vrijednost nekog drugog koncepta. Navesti sve podkoncepte i nadkoncepte iz $NadK_x \cup PodK_x$ kao niz koncepta odvojenih zarezom.
Primjer	Računalni sustav <b>je neposredno povezan s 3 koncepta. Ti koncepti su:</b> Temeljna funkcija računala, Programska podrška, Tehnička podrška.

Predložak9:	
Uvjet	$K_x \in V' \setminus (V_A \cup V_V)$ , $K_i \in PodK_x$ , $\exists (K_x, r, K_{y_1}), \dots, (K_x, r, K_{y_{n_1}}), \dots, (K_x, r, K_{y_{1m}}), \dots, (K_x, r, K_{y_{nm}}) \in PZ'$
Tekst	$K_x$ ima $pK_x$ podkoncepta. $K_x r_1 K_{y_1}, \dots, K_{y_{n_1}}, \dots, K_x r_m K_{y_{1m}}, \dots, K_{y_{nm}}$ .
Napomena	Odabire se jedan proizvoljni koncept koji ne smije biti atribut ili vrijednost nekog drugog koncepta. Navesti sve podkoncepte iz $PodK_x$ kao niz koncepta odvojenih zarezom.
Primjer	Računalni sustav <b>ima 3 podkoncepta</b> . Računalni sustav izvršava Temeljna funkcija računala, Računalni sustav se sastoji od Programska podrška, Tehnička podrška.

Predložak10:	
Uvjet	$K_x \in V' \setminus (V_A \cup V_V)$ , $K_i \in NadK_x$ , $\exists (K_{y_1}, r_1, K_x), \dots, (K_{y_{n_1}}, r_1, K_x), \dots, (K_{y_m}, r_m, K_x), \dots, (K_{y_{nm}}, r_m, K_x) \in PZ'$
Tekst	$K_x$ ima $nK_x$ nadkoncepta. $K_{y_1}, \dots, K_{y_{n_1}} r_1 K_x, \dots, K_{y_{1m}}, \dots, K_{y_{nm}} r_m K_x$ .
Napomena	Odabire se jedan proizvoljni koncept koji ne smije biti atribut ili vrijednost nekog drugog koncepta. Navesti sve nadkoncepte iz $NadK_x$ kao niz koncepta odvojenih zarezom.
Primjer	Centralna jedinica <b>ima 2 nadkoncepta</b> . Tehnička podrška ima podvrstu Centralna jedinica. Model računalnog sustava se sastoji od Centralna jedinica.

### Četvrta težinska kategorija

Predložak11:	
Uvjet	$K_j \in V' \setminus (V_A \cup V_V)$ , $\exists (K_j, r_j, K_{j+1}) \in PZ'$ , $j > 1$
Tekst	Koncepti su neposredno povezani sljedećim redoslijedom: $K_1, r_1, K_2, r_2, \dots, r_n, K_{n+1}$
Napomena	Odabiru se proizvoljni koncepti koji moraju činiti povezani niz koncepta. Također je važno da ni jedan od njih nije atribut ili vrijednost nekog drugog koncepta.
Primjer	<b>Koncepti su neposredno povezani sljedećim redoslijedom:</b> Računalni sustav izvršava Temeljna funkcija računala. Temeljna funkcija računala ima primjerak Unos podataka. Temeljna funkcija računala ima primjerak Prikazivanje podataka. Temeljna funkcija računala ima primjerak Obrada podataka.

Predložak12:	
Uvjet	$\exists (K_x, slot, K_{slot}), (K_{slot}, filler, K_{filler}), (K_x, filler, K_{filler}) \in PZ'$
Tekst	$K_{slot}$ od $K_x$ je $K_{filler}$ .
Napomena	Odabiru se jedan proizvoljni koncept $K_x$ , jedan njegov proizvoljni atribut $K_{slot}$ i njegova vrijednost $K_{filler}$ .
Primjer	Kapacitet <b>od</b> Disketa <b>je</b> 1.44MB

#### 4.2.4.2 Algoritmi za prezentiranje elemenata računalom oblikovanog nastavnog sadržaja

Na temelju predložaka za tvrdnje i razine elementa računalom oblikovanog nastavnog sadržaja kojom određujemo količinu znanja koju će učenik učiti u jednoj iteraciji učenja i poučavanja, definiramo algoritme za prezentiranje elemenata računalom oblikovanog nastavnog sadržaja za svaki od stereotipova. Svaki algoritam je popraćen primjerom iz područnog znanja „Računalo kao sustav“.

##### Stereotip *novak*

Ako učenik pripada stereotipu *novak*, onda će mu se prezentirati NS prema sljedećem algoritmu:

<b>Algoritam 21. Novak – prezentiranje elemenata računalom oblikovanog nastavnog sadržaja</b>
<p>1. Ponavlja se za svaki element iz NS:</p> <p>1.1. Ako je element nastavna jedinica NJ, onda za svaku relaciju iz NJ koju učenik ne poznaje, tj. <math>X_A(K_x K_y) = -1,0</math>, generiraj tvrdnje prema slučajno odabranim predlošcima iz prve težinske kategorije (za svaku relaciju mora se generirati jedna tvrdnja).</p> <p>1.2. Ako je element test, onda pokreni test. Test se generira nad konceptima i relacijama iz nastavnih jedinica koje mu prethode, a nalaze se iza nekog drugog testa (ako postoji).</p> <p style="padding-left: 40px;">1.2.1. Na temelju rezultata kviza napravi reviziju stereotipa.</p>

<b>Primjer 29.</b>
<p>Primjer prezentiranja jednog elementa računalom oblikovanog nastavnog sadržaja (<math>NJ_{Računalni\ sustav}</math>) učeniku čiji je stereotip <i>novak</i>:</p> <p>Računalni sustav izvršava Temeljna funkcija računala. Računalni sustav <u>je nadkoncept od</u> Temeljna funkcija računala.</p> <p>Računalni sustav se sastoji od Programska podrška. Programska podrška <u>je podkoncept od</u> Računalni sustav.</p> <p>Računalni sustav se sastoji od Tehnička podrška. Tehnička podrška <u>je podkoncept od</u> Računalni sustav.</p> <p>Detaljan prikaz računalom oblikovanog nastavnog sadržaja za ovaj stereotip se može vidjeti u Prilogu 8.4.2.1.</p>

##### Stereotip *početnik*

Ako učenik pripada stereotipu *početnik*, onda će mu se prezentirati NS prema sljedećem algoritmu:

<b>Algoritam 22. Početnik – prezentiranje elemenata računalom oblikovanog nastavnog sadržaja</b>
<p>1. Ponavlja se za svaki element iz NS:</p> <p>1.1. Ako je element nastavna jedinica NJ, onda za svaki koncept iz NJ, generiraj tvrdnje prema predlošcima 4 ili 5 iz druge težinske kategorije nad konceptima iz te NJ. Tvrdnje prema predlošcima 6 ili 7 se mogu generirati (jedan koncept mora biti iz te NJ) samo ako prezentiraju relaciju koju učenik ne poznaje, tj. <math>X_A(K_x K_y) = -1,0</math> i ako ista relacija nije bila obuhvaćena s tvrdnjom 4 ili 5.</p> <p>1.2. Za sve NJ koje se nalaze između dva testa i sve njihove koncepte, generiraj tvrdnje prema predlošku 3 (za svaka dva koncepta za koja je to moguće). Koncepte treba prvo sortirati pretraživanjem po dubini i redom ispisivati tvrdnje za putanje koje imaju duljinu najmanje 1 i ako postoji barem jedna relacija u potanji koja nije prezentirana.</p> <p>1.3. Ako je element test, onda pokreni test. Test se generira nad konceptima i relacijama iz nastavnih jedinica koje mu prethode, a nalaze se iza nekog drugog testa (ako postoji).</p> <p style="padding-left: 40px;">1.3.1. Na temelju rezultata kviza napravi reviziju stereotipa.</p>

**Primjer 30.**

Primjer prezentiranja dva elementa računalom oblikovanog nastavnog sadržaja ( $NJ_{\text{Računalni sustav}}$  i  $NJ_{\text{Tehnička podrška}}$ ) učeniku čiji je stereotip *početnik*:

Računalni sustav se sastoji od Programska podrška, Tehnička podrška. Ovo su podkoncepti od Računalni sustav.

Računalni sustav izvršava Temeljna funkcija računala. Temeljna funkcija računala ima nadkoncept Računalni sustav.

Tehnička podrška ima podvrstu Centralna jedinica, Izlazna jedinica, Memorija, Ulazna jedinica, Uređaj za komunikaciju. Ovo su podkoncepti od Tehnička podrška.

Računalni sustav i Ulazna jedinica su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→Ulazna jedinica.

Računalni sustav i Centralna jedinica su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→Centralna jedinica.

Računalni sustav i Uređaj za komunikaciju su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→Uređaj za komunikaciju.

Računalni sustav i Memorija su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→Memorija.

Računalni sustav i Izlazna jedinica su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→Izlazna jedinica.

Detaljan prikaz računalom oblikovanog nastavnog sadržaja za ovaj stereotip se može vidjeti u Prilogu 8.4.2.1.

**Stereotip osrednji**

Ako učenik pripada stereotipu *osrednji*, onda će mu se prezentirati NS prema sljedećem algoritmu:

**Algoritam 23. Osrednji – prezentiranje elemenata računalom oblikovanog nastavnog sadržaja**

1. Ponavljač za svaki element iz NS:

1.4. Ako je element nastavna tema ili jedinica, onda za svaki koncept, generiraj tvrdnje prema predlošcima 4 ili 5 iz druge težinske kategorije nad konceptima iz te NJ. Tvrdnje prema predlošcima 6 ili 7 se mogu generirati (jedan koncept mora biti iz NJ ili NT) samo ako prezentiraju relaciju koju učenik ne poznaje, tj.  $X_A(K_x K_y) = -1, 0, 1$  i ako ista relacija nije bila obuhvaćena s tvrdnjom 4 ili 5.

1.1. Za sve nastavne teme i jedinice koje se nalaze između dva testa i sve njihove koncepte, generiraj tvrdnje prema predlošku 3 (za svaka dva koncepta za koja je to moguće). Koncepte treba prvo sortirati pretraživanjem po dubini i redom ispisivati tvrdnje za putanje koje imaju duljinu najmanje 1 i ako postoji barem jedna relacija u potanji koja nije prezentirana.

1.2. Ako je element test, onda pokreni test. Test se generira nad konceptima i relacijama iz nastavnih jedinica koje mu prethode, a nalaze se iza nekog drugog testa (ako postoji).

1.2.1. Na temelju rezultata kviza napravi reviziju stereotipa.

**Primjer 31.**

Primjer prezentiranja nekoliko elementa računalom oblikovanog nastavnog sadržaja (nisu navedene sve tvrdnje zbog jednostavnosti) učeniku čiji je stereotip *osrednji*:

Računalni sustav se sastoji od Programska podrška, Tehnička podrška. Ovo su podkoncepti od Računalni sustav.

Računalni sustav izvršava Temeljna funkcija računala. Temeljna funkcija računala ima nadkoncept Računalni sustav.

Tehnička podrška ima podvrstu Centralna jedinica, Izlazna jedinica, Memorija, Ulazna jedinica, Uređaj za komunikaciju. Ovo su podkoncepti od Tehnička podrška.

Centralna jedinica se sastoji od Masovna memorija, Računalo. Ovo su podkoncepti od Centralna jedinica.

Masovna memorija ima podvrstu Disketa, Kompaktni disk, Tvrdi disk. Ovo su podkoncepti od Masovna memorija.

Računalo se sastoji od Centralna procesorska jedinica, Radna memorija. Ovo su podkoncepti od Računalo.



Računalni sustav i Računalo su posredno povezani jer postoji put Računalni sustav → Tehnička podrška → Centralna jedinica → Računalo.  
 Računalni sustav i Masovna memorija su posredno povezani jer postoji put Računalni sustav → Tehnička podrška → Centralna jedinica → Masovna memorija.  
 Računalni sustav i Disketa su posredno povezani jer postoji put Računalni sustav → Tehnička podrška → Centralna jedinica → Masovna memorija → Disketa.

Tehnička podrška i Računalo su posredno povezani jer postoji put Tehnička podrška → Centralna jedinica → Računalo.  
 Tehnička podrška i Masovna memorija su posredno povezani jer postoji put Tehnička podrška → Centralna jedinica → Masovna memorija.  
 Tehnička podrška i Disketa su posredno povezani jer postoji put Tehnička podrška → Centralna jedinica → Masovna memorija → Disketa.

Centralna jedinica i Disketa su posredno povezani jer postoji put Centralna jedinica → Masovna memorija → Disketa.

Detaljan prikaz računalom oblikovanog nastavnog sadržaja za ovaj stereotip se može vidjeti u Prilogu 8.4.2.1.

## Stereotip napredni

Ako učenik pripada stereotipu *napredni*, onda će mu se prezentirati NS prema sljedećem algoritmu:

### Algoritam 24. Napredni – prezentiranje elemenata računalom oblikovanog nastavnog sadržaja

1. Ponavljaj za svaki element iz NS:

- 1.1. Ako je element nastavna tema ili jedinica, onda za svaki koncept (osim listova), prvo generiraj tvrdnje prema predlošku 8 iz treće težinske kategorije, osim ako učenik poznaje svaku relaciju prema nadkonceptima i podkonceptima na razini 3, tj.  $X_A(K_x K_y) = 3$ . Zatim generiraj tvrdnje prema predlošku 9 iz treće težinske kategorije, osim ako učenik poznaje svaku relaciju prema podkonceptima na razini 3, tj.  $X_A(K_x K_y) = 3$ . Na kraju generiraj tvrdnje prema predlošku 10 iz treće težinske kategorije, osim ako učenik poznaje svaku relaciju prema nadkonceptima na razini 3, tj.  $X_A(K_x K_y) = 3$ .
- 1.2. Ako je element test, onda pokreni test. Test se generira nad konceptima i relacijama iz nastavnih jedinica koje mu prethode, a nalaze se iza nekog drugog testa (ako postoji).

1.2.1. Na temelju rezultata kviza napravi reviziju stereotipa.

### Primjer 32.

Primjer prezentiranja dva elementa računalom oblikovanog nastavnog sadržaja učeniku čiji je stereotip *napredni*:

Računalni sustav je neposredno povezan s 3 konceptata. Ti koncepti su: Temeljna funkcija računala, Programska podrška, Tehnička podrška.

Računalni sustav ima 3 podkonceptata. Računalni sustav izvršava Temeljna funkcija računala, Računalni sustav se sastoji od Programska podrška, Tehnička podrška.

Računalni sustav ima 0 nadkonceptata.

Temeljna funkcija računala je neposredno povezan s 4 konceptata. Ti koncepti su: Obrada podataka, Prikazivanje podataka, Računalni sustav, Unos podataka.

Temeljna funkcija računala ima 3 podkonceptata. Temeljna funkcija računala ima primjerak Obrada podataka, Prikazivanje podataka, Unos podataka.

Temeljna funkcija računala ima 1 nadkonceptata. Računalni sustav izvršava Temeljna funkcija računala

Programska podrška je neposredno povezan s 3 konceptata. Ti koncepti su: Aplikacijska programska podrška, Sistemska programska podrška, Računalni sustav.

Programska podrška ima 2 podkonceptata. Programska podrška ima podvrstu Aplikacijska programska podrška, Sistemska programska podrška.

Programska podrška ima 1 nadkonceptata. Računalni sustav izvršava Programska podrška

Tehnička podrška je neposredno povezan s 6 konceptata. Ti koncepti su: Centralna jedinica, Izlazna jedinica, Memorija, Ulazna jedinica, Računalni sustav, Uređaj za komunikaciju.

Tehnička podrška ima 5 podkonceptata. Tehnička podrška ima podvrstu Centralna jedinica, Izlazna jedinica, Memorija, Ulazna jedinica, Uređaj za komunikaciju.

Tehnička podrška ima 1 nadkonceptata. Računalni sustav izvršava Tehnička podrška

Detaljan prikaz računalom oblikovanog nastavnog sadržaja za ovaj stereotip se može vidjeti u Prilogu 8.4.2.1.

## Stereotip *stručnjak*

Ako učenik pripada stereotipu *stručnjak*, onda će mu se prezentirati NS prema sljedećem algoritmu:

<b>Algoritam 25. Stručnjak – prezentiranje elemenata računalom oblikovanog nastavnog sadržaja</b>
<p>1. Ponavljač za svaki element iz NS:</p> <p>1.1. Ako je element nastavna cjelina, tema ili jedinica, onda za svaki koncept pronađi putanju dulju od 1 od korijena cjeline, te generiraj tvrdnju prema predlošku 11 iz četvrte težinske kategorije, osim ako svaku relaciju iz te putanje učenik poznaje na razini 4, tj. <math>X_A(K_x, K_y) = 4</math>.</p> <p>1.2. Ako element sadrži atribut i vrijednost atributa, generiraj tvrdnju prema predlošku 12 iz četvrte težinske kategorije.</p> <p>1.3. Ako je element test, onda pokreni test. Test se generira nad konceptima i relacijama iz nastavnih jedinica koje mu prethode, a nalaze se iza nekog drugog testa (ako postoji).</p> <p>1.3.1. Na temelju rezultata kviza napravi reviziju stereotipa.</p>

<b>Primjer 33.</b>
<p>Primjer prezentiranja dijela računalom oblikovanog nastavnog sadržaja (nisu navedene sve tvrdnje zbog jednostavnosti) učeniku čiji je stereotip <i>stručnjak</i>:</p> <p><b>Koncepti su neposredno povezani sljedećim redoslijedom:</b></p> <p>Računalni sustav izvršava Temeljna funkcija računala ima primjerak Unos podataka          Temeljna funkcija računala ima primjerak Prikazivanje podataka          Temeljna funkcija računala ima primjerak Obrada podataka          Obrada podataka se sastoji od Pohrana podataka          Obrada podataka se sastoji od Upravljanje podataka          Računalni sustav se sastoji od Programska podrška ima podvrstu Aplikacijska programska podrška          Programska podrška ima podvrstu Sistemska programska podrška          Sistemska programska podrška ima podvrstu Uslužni programi          Sistemska programska podrška ima podvrstu Operacijski sustav          Operacijski sustav ima primjerak Unix          Operacijski sustav ima primjerak DOS          Operacijski sustav ima primjerak Windows</p> <p>Detaljan prikaz računalom oblikovanog nastavnog sadržaja za ovaj stereotip se može vidjeti u Prilogu 8.4.2.1.</p>

### 4.2.5 Dinamičko generiranje, odabir, nizanje i prezentiranje računalom oblikovanog nastavnog sadržaja

Sljedeći algoritam pokazuje kako se odvija cjelokupni proces učenja i poučavanja u sustavu koji podržava automatsko i dinamičko generiranje, odabir, nizanje i prezentiranje računalom oblikovanog nastavnog sadržaja temeljeno na rezultatima testa, a prilagođeno stereotipu koji se pridruži učeniku upravo prema rezultatima tog istog testa.

Cijeli proces učenja i poučavanja je ciklus kojeg čini:

**učenje i poučavanje prilagođeno stereotipu → testiranje → određivanje stereotipa**

Ulaz u cijeli proces učenja i poučavanja, odnosno u njegov prvi ciklus, je stereotip određen na temelju ulaznog testa. Svaki ciklus završava određivanjem novog stereotipa učenika (ili eventualno zadržavanjem već pridijeljenog stereotipa iz prošlog ciklusa).

Proces učenja i poučavanja se, dakle, sastoji od ciklusa koji su opisani sljedećim algoritmom:

<b>Algoritam 26. Ciklusi u procesu učenja i poučavanja</b>
<ol style="list-style-type: none"> <li>1. Nakon pridruživanja stereotipa učeniku slijedi generiranje i prezentiranje elemenata računalom oblikovanog nastavnog sadržaja prilagođeno određenom stereotipu.</li> <li>2. Ako je određeni stereotip <i>novak</i>, onda: <ol style="list-style-type: none"> <li>2.1. Dodavanje i nizanje elemenata NS - Algoritam 11 <ol style="list-style-type: none"> <li>2.1.1. Ako u NS nema elemenata, onda idi na točku 10.</li> </ol> </li> <li>2.2. Prezentiranje elemenata NS - Algoritam 21 <ol style="list-style-type: none"> <li>2.2.1. Ako je element test, onda idi na točku 7.</li> </ol> </li> </ol> </li> <li>3. Ako je određeni stereotip <i>početnik</i>, onda: <ol style="list-style-type: none"> <li>3.1. Dodavanje i nizanje elemenata NS - Algoritam 13 <ol style="list-style-type: none"> <li>3.1.1. Ako u NS nema elemenata, onda idi na točku 10.</li> </ol> </li> <li>3.2. Prezentiranje elemenata NS - Algoritam 22 <ol style="list-style-type: none"> <li>3.2.1. Ako je element test, onda idi na točku 7.</li> </ol> </li> </ol> </li> <li>4. Ako je određeni stereotip <i>osrednji</i>, onda: <ol style="list-style-type: none"> <li>4.1. Dodavanje i nizanje elemenata NS - Algoritam 15 <ol style="list-style-type: none"> <li>4.1.1. Ako u NS nema elemenata, onda idi na točku 10.</li> </ol> </li> <li>4.2. Prezentiranje elemenata NS - Algoritam 23 <ol style="list-style-type: none"> <li>4.2.1. Ako je element test, onda idi na točku 7.</li> </ol> </li> </ol> </li> <li>5. Ako je određeni stereotip <i>napredni</i>, onda: <ol style="list-style-type: none"> <li>5.1. Dodavanje i nizanje elemenata NS - Algoritam 17 <ol style="list-style-type: none"> <li>5.1.1. Ako u NS nema elemenata, onda idi na točku 10.</li> </ol> </li> <li>5.2. Prezentiranje elemenata NS - Algoritam 24 <ol style="list-style-type: none"> <li>5.2.1. Ako je element test, onda idi na točku 7.</li> </ol> </li> </ol> </li> <li>6. Ako je određeni stereotip <i>stručnjak</i>, onda: <ol style="list-style-type: none"> <li>6.1. Dodavanje i nizanje elemenata NS - Algoritam 19 <ol style="list-style-type: none"> <li>6.1.1. Ako u NS nema elemenata, onda idi na točku 10.</li> </ol> </li> <li>6.2. Prezentiranje elemenata NS - Algoritam 25 <ol style="list-style-type: none"> <li>6.2.1. Ako je element test, onda idi na točku 7.</li> </ol> </li> </ol> </li> <li>7. Učenik se testira. Test se generira nad konceptima i relacijama iz elemenata NS koje mu prethode, a nalaze se iza nekog drugog testa (ako postoji). Pitanja u testu generiraju se prema trenutnom učenikovom stereotipu (osim kod ulaznog testa). Rezultati testa određuju učenikov novi stereotip (koji može ostati i nepromijenjen).</li> <li>8. Ako nakon testa za svaku uređenu trojku <math>(K_x, r, K_y) \in PZ, K_x, K_y \in V \setminus V_A \cup V_V</math>, vrijedi <math>X_A(K_x K_y) \neq -1, 0</math>, onda učeniku ponuditi izbor: nastaviti ili završiti proces učenja i poučavanja. Ako je učenik odabrao nastavak, onda idi na točku 9, inače idi na točku 10.</li> <li>9. Iz NS izbriši sve elemente na kojima se učenik upravo učio i poučavao. Idi na točku 1.</li> <li>10. Završi s procesom učenja i poučavanja.</li> </ol>

U svakom ciklusu učenik se uči i poučava samo na jednom dijelu područnog znanja, koje mu je u tom ciklusu prezentirano na način koji je prilagođen onom stereotipu koji je rezultat prethodnog ciklusa. Tada se pristupa testiranju tog znanja koje dovodi mijenjanja. Proces učenja i poučavanja završava kada učenik pokaže poznavanje svih koncepata barem na razini 1. Izlaz iz cijelog procesa učenja i poučavanja predstavlja određivanje učenikova *završnog stereotipa*.

Primijetimo da učenik ne može završiti proces učenja i poučavanja kao novak. Ukoliko učenik želi završiti proces učenja i poučavanja kao bolji stereotip, onda ponovno ulazi u proces učenja i poučavanja s tim da ne treba pisati ulazni test, već mu je početni stereotip onaj koji je bio izlaz u prethodnom procesu učenja i poučavanja.

### 4.3 Modeliranje učenika

Jedan od glavnih problema modeliranja učenika u sustavima koji koriste stereotipove je određivanje stereotipa koji najbolje opisuje učenika te u čemu se individualni učenik razlikuje od svog stereotipa. U našem pristupu razlikovanje učenika od pridijeljenog mu stereotipa opisujemo pomoću Bayesovog modela učenika koji omogućava da sustav učeniku, u svakom ciklusu učenja i poučavanja, prezentira samo one tvrdnje koje sadrže ono znanje koje učenik vjerojatno ne posjeduje.

Na temelju provedenog testa i vrijednosti težinske funkcije  $X_V$  na skupu vrhova grafa područnog znanja (Definicija 2.15), definiramo Bayesovu mrežu predikcija poznavanja koncepata područnog znanja.

**Definicija 4.11** *Bayesova mreža predikcija poznavanja koncepata područnog znanja PZ* je usmjereni aciklički graf BMP. Vrhovi su slučajne varijable  $K_x$  (odgovaraju konceptima područnog znanja) koje mogu poprimiti vrijednosti T (usvojen) i F (neusvojen), a usmjereni bridovi između slučajnih varijabli pokazuju povezanost između istih (odgovaraju bridovima između koncepata).

Za upotpunjenje Bayesove mreže strukturi je potrebno dodati:

- „a priori“ vjerojatnosti  $P(K_x)$  za vrhove bez roditelja (korijeni), te
- uvjetne vjerojatnosti  $P(K_x | K_y \in \text{Nad}K_x, K_y = T \vee K_y = F)$  za sve čvorove s roditeljima za sve moguće kombinacije ishoda njihovih roditelja

Dakle,  $P(K_x | K_y \in \text{Nad}K_x, K_y = T \vee K_y = F)$  je funkcija vjerojatnosti da učenik ne poznaje nijedan koncept koji odgovara roditeljima slučajne varijable iako poznaje koncept koji odgovara slučajnoj varijabli (*slučajna pogreška* - eng. unlucky slip) ili da poznaje sve koncepte koji odgovaraju roditeljima slučajne varijable iako ne poznaje koncept koji odgovara slučajnoj varijabli (*slučajni pogodak* – eng. lucky guess). U literaturi se ovim slučajevima pridružuju vjerojatnosti 0,1 i 0,9 (Mayo, 2001) - Tablica 4.6.

Tablica 4.6. Uvjetne vjerojatnosti

	$P(K_x = T   K_y \in \text{Nad}K_x, K_y = ?)$	$P(K_x = F   K_y \in \text{Nad}K_x, K_y = ?)$
$K_y = T$	0,9	0,1 – slučajna pogreška
$K_y = F$	0,1 – slučajni pogodak	0,9

#### Određivanje „a priori“ vjerojatnosti

Neka je  $K_x$  jedan korijen Bayesove mreže BPM. „A priori“ vjerojatnost  $P(K_x) = P(K_x = T) = X_V(K_x)$  određuje vjerojatnost poznavanja koncepta  $K_x$ , nakon provedene provjere znanja učenika, odnosno vjerojatnost  $P(K_x = F) = 1 - X_V(K_x)$  određuje vjerojatnost nepoznavanja koncepta  $K_x$ .

### Određivanje uvjetnih vjerojatnosti

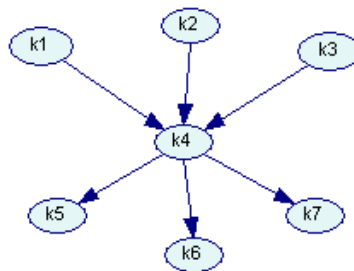
Neka je  $K_x$  slučajna varijabla iz BMP koja nije korijen koja odgovara konceptu  $K_x$ . Neka je  $NadK_x$  skup roditelja slučajne varijable  $K_x$  koji odgovaraju nadkonceptima od  $K_x$ , a  $nK_x$  broj roditelja slučajne varijable  $K_x$ . Analogno, neka je  $PodK_x$  skup djece slučajne varijable  $K_x$  koji odgovaraju podkonceptima od  $K_x$ , a  $pK_x$  broj djece slučajne varijable  $K_x$ .

Tablica uvjetne vjerojatnosti slučajne varijable  $K_x$  ima  $2^{nK_x}$  redaka. Svaki redak sadrži jednu od  $2^{nK_x}$  kombinacija vrijednosti T i F, odnosno u svakom retku se nalazi  $p$  vrijednosti T. Svaka T vrijednost ima „težinu“  $\frac{0,9}{nK_x}$ . Pojedinačne težine vrijednosti T nam omogućavaju pridjeljivanje „težine“  $p * \frac{0,9}{nK_x}$  retku tablice. Upravo ova „težina“ retka odgovara *uvjetnoj vjerojatnosti* slučajne varijable  $K_x$ , tj.  $P(K_x = T | K_y \in NadK_x, K_y = T \vee K_y = F)$ .

Objasnimo određivanje uvjetnih vjerojatnosti na sljedećem primjeru:

#### Primjer 34.

Neka je dana sljedeća Bayesova mreža:



Korijenski vrhovi k1, k2 i k3 imaju sljedeće "a priori vjerojatnosti":

Vrijednost	$P(k1)$	$P(k2)$	$P(k3)$
T	0,5	0,5	0,5
F	0,5	0,5	0,5

Pošto slučajna varijabla k4 ima 3 roditelja, svaka T vrijednost u tablici uvjetnih vjerojatnosti ima "težinu"  $\frac{0,9}{3} = 0,3$ . Dakle, slučajna varijabla k4 ima sljedeću tablicu uvjetnih vjerojatnosti:

k1	k2	k3	$P(k4 = T   k1 = ?, k2 = ?, k3 = ?)$	$P(k4 = F   k1 = ?, k2 = ?, k3 = ?)$
T	T	T	0,9	0,1 – slučajna pogreška
T	T	F	0,6	0,3
T	F	T	0,6	0,3
T	F	F	0,3	0,6
F	T	T	0,6	0,3
F	T	F	0,3	0,6
F	F	T	0,3	0,3
F	F	F	0,1 – slučajni pogodak	0,9

Pošto slučajne varijable k5, k6 i k7 imaju samo 1 roditelja, svaka T vrijednost u tablici uvjetnih vjerojatnosti ima "težinu"  $\frac{0,9}{1} = 0,9$ . Dakle, slučajne varijable k5, k6 i k7 imaju iste tablice uvjetnih vjerojatnosti:

k4	$P(k5 = T   k4 = ?)$	$P(k5 = F   k4 = ?)$
T	0,9	0,1 – slučajna pogreška
F	0,1 – slučajni pogodak	0,9

### Zaključivanje u Bayesovoj mreži

Na temelju dokaza o vrijednostima varijabli i iz postavljenih vjerojatnosti mogu se izračunati „a priori“ vjerojatnosti za sve varijable koje nisu korijeni.

Za računanje „a priori“ vjerojatnosti na temelju uvjetnih vjerojatnosti koristimo sljedeću formulu:

$$P(K_x) = \sum_{v \in \{T, F\}} P(K_x | K_1 = v, \dots, K_{n_{K_x}} = v) P(K_1 = v, \dots, K_{n_{K_x}} = v) \quad (4.1)$$

Dakle, za izračunavanje „a priori“ vjerojatnosti potrebno je izračunati spojne vjerojatnosti (formula (2.1)) za sve kombinacije vrijednosti roditelja slučajne varijable  $K_x$ , dok su uvjetne vjerojatnosti već određene na temelju broja roditelja (formula (4.2)):

$$\begin{aligned} P(K_1, K_2, \dots, K_{n_{K_x}}) &= P(K_1 | K_2, \dots, K_{n_{K_x}}) P(K_2, \dots, K_{n_{K_x}}) \\ &= P(K_1 | K_2, \dots, K_{n_{K_x}}) P(K_1 | K_3, \dots, K_{n_{K_x}}) P(K_3, \dots, K_{n_{K_x}}) \\ &= P(K_1 | K_2, \dots, K_{n_{K_x}}) P(K_1 | K_3, \dots, K_{n_{K_x}}) \dots P(K_{n_{K_x}-1} | K_{n_{K_x}}) P(K_{n_{K_x}}) \end{aligned} \quad (4.2)$$

Objasnimo određivanje „a priori“ i uvjetnih vjerojatnosti na sljedećem primjeru:

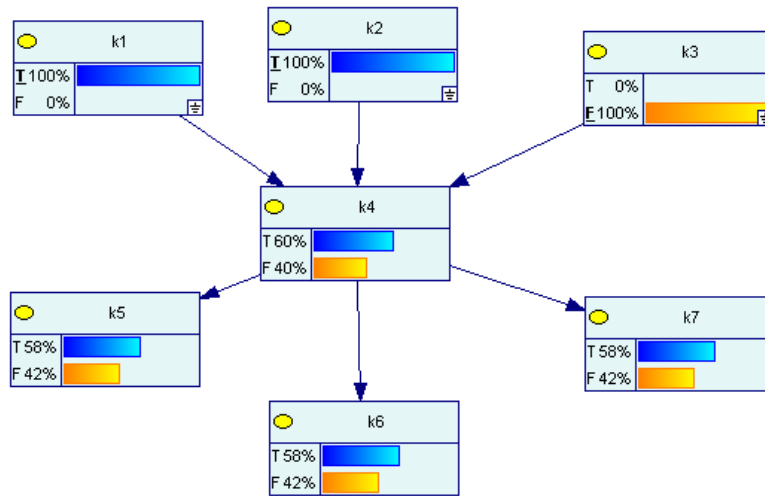
**Primjer 35.**

Ako postavimo dokaze za vrijednosti slučajnim varijablama  $k_1, k_2$  i  $k_3$  na T, odnosno vrijedi  $P(k_1) = P(k_2) = P(k_3) = 1$ .  
Tada je  $P(k_4) = 0,9$ .  
Dobivamo sjedeće vjerojatnosti u Bayesovoj mreži:

The diagram shows a Bayesian network with the following structure and values:

- Node k1:** T 100%, F 0%
- Node k2:** T 100%, F 0%
- Node k3:** T 100%, F 0%
- Node k4:** T 90%, F 10% (parents: k1, k2, k3)
- Node k5:** T 82%, F 18% (parent: k4)
- Node k6:** T 82%, F 18% (parent: k4)
- Node k7:** T 82%, F 18% (parent: k4)

Ako postavimo dokaze za vrijednosti slučajnim varijablama k1 i k2 na T, a k3 na F, dobivamo sjedeće vjerojatnosti u Bayesovoj mreži:



Algoritam za ažuriranje modela učenika koje se realizira nakon učenikova odgovora na pojedinačno pitanje:

#### Algoritam 27. Algoritam za ažuriranje modela učenika

Za svaki koncept  $K_x$

vjerojatnost da učenik poznaje koncept  $K_x$  je  $P(K_x = T | K_{y_1}, \dots, K_{y_n})$ ,  
gdje su  $K_{y_i}$  podkoncepti od  $K_x$

Model učenika koristimo prilikom prezentiranja elemenata računalom oblikovanog nastavnog sadržaja za učenje, kada učeniku želimo prezentirati, na prilagođeni način, samo one koncepte koje on vjerojatno ne poznaje (granica vjerojatnosti koja se uzima u ovom slučaju je 0,8). Pošto se prezentiranje elemenata računalom oblikovanog nastavnog sadržaja, odnosno samih koncepata i relacija, vrši pomoću predložaka tvrdnji za prezentiranje koje imaju određene uvjete, potrebno je te uvjete dodatno „otežati“ i vjerojatnostima.

Predložci tvrdnji koji za uvjet imaju postojanje relacije, zahtijevaju izračunavanje spojne vjerojatnosti  $P(K_x, K_y)$  koja govori o vjerojatnosti poznavanja oba koncepta  $K_x$  i  $K_y$ . Prema formuli (2.1), za izračunavanje ove vjerojatnosti trebamo znati uvjetnu vjerojatnost  $P(K_y | K_x)$  ili  $P(K_x | K_y)$  i odgovarajuću „a priori“ vjerojatnost  $P(K_x)$  ili  $P(K_y)$ .

Za računanje složenih spojnih vjerojatnosti koristimo formulu (2.2).

#### Primjer 36.

Na primjer, neka su  $K_1$  i  $K_2$  koncepti područnog znanja. Pretpostavimo da je  $P(K_1 = T) = 0,20$  i  $P(K_2 = T | K_1 = T) = 0,70$ . Vjerojatnost ovakvih spojenih događaja je definirana sa  $P(K_1, K_2) = P(K_1) * P(K_2 | K_1)$ , što se može prikazati  $P(K_2 | K_1) = \frac{P(K_1, K_2)}{P(K_1)}$ .

Dakle, vjerojatnost da učenik poznaje oba koncepta  $K_1$  i  $K_2$  je jednaka  $P(K_1, K_2) = P(K_1) * P(K_2 | K_1) = 0,20 * 0,70 = 0,14$ .

Vrijedi  $P(K_1 = F) = 1 - 0,20 = 0,80$  i  $P(K_2 = T | K_1 = F) = 1 - 0,70 = 0,30$

Probabilističko zaključivanje se može primijeniti i kod zaključivanja o usvojenosti nekog koncepta ovisno o učenikovom odgovoru na konkretno pitanje. Neka je  $K_x$  slučajna varijabla koja može poprimiti vrijednosti usvojen (T) i neusvojen (F), a  $Q_x$  pitanje koje može biti odgovoreno točno (T) ili netočno (F).  $P(K_x)$  je „a priori“ uvjerenje o usvojenosti koncepta, a  $P(K_x|Q_x)$  je uvjetno uvjerenje u usvojenost koncepta ovisno o odgovoru na pitanje. Na temelju Bayesovog teorema (formula (2.3)) dobivamo sljedeće:

$$P(K_x|Q_x) = \frac{P(Q_x|K_x)P(K_x)}{P(Q_x)}$$

$$= \frac{P(Q_x|K_x)P(K_x)}{P(Q_x|K_x = T)P(K_x = T) + P(Q_x|K_x = F)P(K_x = F)}$$
(4.3)

Vidimo da nam je za izračunavanje ove formule potrebno poznavati vrijednosti od  $P(Q_x|K_x)$ , što je vjerojatnost odgovaranja na pitanje s obzirom na usvojenost koncepta. Dakle,  $P(Q_x|K_x)$  je funkcija vjerojatnosti da će učenik netočno odgovoriti na pitanje iako poznaje koncept (slučajna pogreška) ili će točno odgovoriti na pitanje iako koncept ne poznaje (slučajni pogodak).

Heuristike koje se mogu koristiti za ažuriranje modela učenika na temelju točnosti odgovora na pitanja:

1. Ako je pitanje koje se odnosi na koncept  $K_x$  točno odgovoreno, onda se „a priori“ vjerojatnost  $P(K_x = T)$  povećava za 10%
2. Ako je pitanje koje se odnosi na koncept  $K_x$  netočno odgovoreno, onda se „a priori“ vjerojatnost  $P(K_x = T)$  smanjuje za 20%

U nastavku ćemo pokazati kako smo inicijalizirali Bayesovu mrežu na temelju opisanih propozicija. U realizaciji primjera korišten je sustav GeNIe (Graphical Network Interface, <http://genie.sis.pitt.edu/>), grafičko sučelje za SMILE (Structural Modeling, Inference and Learning Engine), prenosni Bayesov stroj za zaključivanje koji je razvijen od strane Decision Systems Laboratory.

Riječ je o Bayesovoj mreži od 65 čvorova (veća komponenta područnog znanja „Računalo kao sustav“) i sve varijable su logičke (2 stanja T i F) – Prilog 8.5. Važno je naglasiti da mreža ima 4 korijenska čvora: *Računalni sustav*, *Model računalnog sustava*, *Programski jezik* i *Logički sklop*, te su njihove „a priori“ vjerojatnosti postavljene na 0,5 za T, te 0,5 za F. Ove vrijednosti vrijede samo za inicijalizaciju modela učenika. Nakon svakog testa mijenjaju se „a priori“ vjerojatnosti korijenski čvorova prema vrijednosti funkcije  $X_V$ , odnosno  $P(K_x) = P(K_x = T) = X_V(K_x)$ .

Promotrimo, na primjer čvor *Programska podrška*. Budući da *Programska podrška* ima jednoga roditelja, vrijedi da svaka T vrijednost u tablici uvjetnih vjerojatnosti iznosi



$0,9/1=0,9$ . F vrijednost dobijemo kao komplement, i ona iznosi 0,1. Dakle, uvjetna vjerojatnost čvora *Programska podrška* je dana sa:

Racunalni sustav	Istina	Laz
Istina	0.9	0.1
Laz	0.1	0.9

Slika 4.1. Uvjetna vjerojatnost čvora *Programska podrška*

Na primjer, analizirajmo definiranje uvjetnih vjerojatnosti čvora *Masovna memorija* koji ima dva roditelja. U ovom slučaju svaka T u tablici uvjetnih vjerojatnosti iznosi  $0,9/2=0,45$ . Dakle, uvjetna vjerojatnost čvora *Masovna memorija* je dana sa:

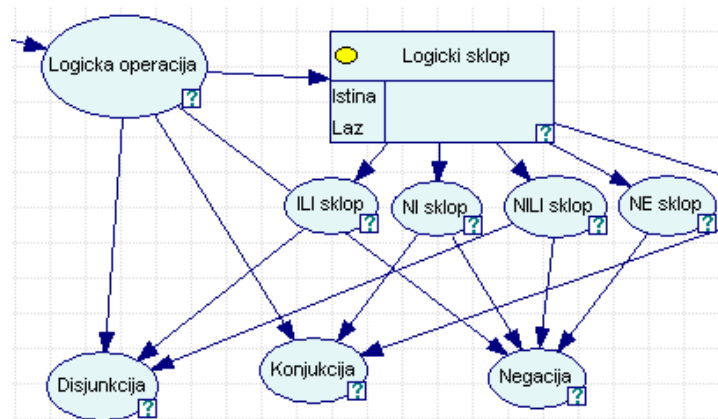
Centralna jedini..	Istina		Laz	
Memorija	Istina	Laz	Istina	Laz
Istina	0.9	0.45	0.45	0.1
Laz	0.1	0.55	0.55	0.9

Slika 4.2. Uvjetna vjerojatnost čvora *Masovna memorija*

Čvor *Negacija* ima 4 roditelja (Slika 4.4). Svaka T u tablici vjerojatnosti iznosi  $0,9/4=0,225$ , pa je uvjetna vjerojatnost čvora *Negacija* nešto složenija i dana je sa:

.logicka opera..	Istina								Laz							
	Istina		Laz		Istina		Laz		Istina		Laz		Istina		Laz	
NI sklop																
NILI sklop																
NE sklop																
Istina	0.9	0.675	0.675	0.45	0.675	0.45	0.45	0.225	0.675	0.45	0.45	0.225	0.45	0.225	0.225	0.1
Laz	0.1	0.325	0.325	0.55	0.325	0.55	0.55	0.775	0.325	0.55	0.55	0.775	0.55	0.775	0.775	0.9

Slika 4.3. Uvjetna vjerojatnost čvora *Negacija*



Slika 4.4. Isječak iz Bayesove mreže za čvor *Negacija*

Opisani model sustava automatsko i dinamičko generiranje računalom oblikovanog nastavnog sadržaja je implementirano u prototipnoj verziji koja je opisana u sljedećem poglavlju. U prototipnoj verziji nije do kraja implementirano probabilističko zaključivanje.

U ovom poglavlju detaljno je opisan model sustava za automatsko i dinamičko generiranje prilagodljivog računalom oblikovanog nastavnog sadržaja, čija je zamisao opisana u drugom poglavlju. Važnost našeg pristupa je u prilagođavanju sadržaja znanju učenika, dok se prilagođavanje u sustavima e-učenja, u većini slučajeva, realizira kroz prizmu prilagođavanja sučelja prema različitim stilovima učenja. Detaljno je prikazana struktura područnog znanja temeljenog na ontologiji, te matematički model područnog znanja zasnovan na teoriji grafova. Spomenuta teorija je omogućila definiranje pojmova kao što su šetnja i povezani niz koncepata, komponente i cjeline područnog znanja, centralni vrh, te definiranje algoritma za traženje najdužeg puta u cjelini. Navedeno je omogućilo ne-semantičko matematičko određivanje reprezentacije područnog znanja. U modulu učitelja detaljno je opisano generiranje elemenata računalom oblikovanog nastavnog sadržaja za učenje na razinama nastavne cjeline, teme i jedinice. Opisan je način automatskog generiranja pitanja prema predlošcima koji su u skladu s Bloomovom taksonomijom znanja, te način ocjenjivanja poznavanja povezanosti koncepata na temelju odgovora na pitanja. Zatim je detaljno prikazan način odabira i nizanja elemenata računalom oblikovanog nastavnog sadržaja prilagođen stereotipu učenika, kroz algoritme za novaka, početnika, osrednjeg, naprednog i stručnjaka. Slijedi opis pristupa pristup prezentiranju elemenata računalom oblikovanog nastavnog sadržaja prilagođen stereotipu učenika korištenjem predložaka za tvrdnje koje su u skladu s Bloomovom taksonomijom znanja, kroz algoritme za novaka, početnika, osrednjeg, naprednog i stručnjaka. Naposljetku je opisan cijeli proces u kojem je isprepletano generiranje, odabir, nizanje i prezentiranje prilagodljivog računalom oblikovanog nastavnog sadržaja kroz cikluse u kojima se izmjenjuju učenje i poučavanje prilagođeno stereotipu, testiranje znanja i određivanje novog stereotipa. U odjeljku o modeliranju učenika je objašnjen način na koji se opisuje razlika između individualnog učenika i stereotipa kojem učenik pripada. Zato je definirana Bayesova mreža predikcija i postupak inicijalizacije Bayesovog modela učenika određivanjem „a priori“ vjerojatnosti na temelju vrijednosti funkcije  $X_v$ , te definiranjem tablica uvjetnih vjerojatnosti na temelju broja roditelja koje neki koncept ima. Opisuje se način zaključivanja u Bayesovoj mreži koje se provodi nakon svakog testiranja znanja.

U sljedećem poglavlju opisan će se oblikovanje i implementacija prototipne verzije sustava AC-ware Tutor. Također će biti govora i o vrednovanju primjene prototipne verzije sustava s aspekta zadovoljstva korištenja.

## 5 Prikaz arhitekture i vrednovanje prototipa sustava

„Treba prilagoditi govor publici.“

poslovica

Zamisao i razvoj modela sustava AC-ware Tutor su obuhvatili oblikovanje i implementaciju prototipa kao i primjerenu metodologiju vrednovanja implementiranog prototipa. S tim u vezi, u ovom poglavlju se prikazuju arhitektura prototipa implementiranog sustava AC-ware Tutor i struktura vrednovanja provedenog s odgovarajućim skupinama studenata.

Prototip se opisuje pomoću *Unified Modelling Language* (UML) jezika. Sudionici i funkcionalnosti sustava su prikazane dijagramom slučajeva korištenja, a realizacija funkcionalnosti je prikazana dijagramima aktivnosti.

Arhitektura prototipa je na najvišoj razini opisana dijagramom postavljanja (čvorovi tehničke podrške) i dijagramom komponenti (programske komponente), a zatim se struktura temeljnih komponenti prikazuje preko klasa i objekata. Pojedine funkcionalnosti modela automatskog i dinamičkog generiranja i prilagodljivog odabira, nizanja i prezentiranja računalom oblikovanog nastavnog sadržaja implementirane u sustavu AC-ware Tutor testirane su na područnom znanju "Računalo kao sustav" (detaljno prikazano u Prilogu 8.2).

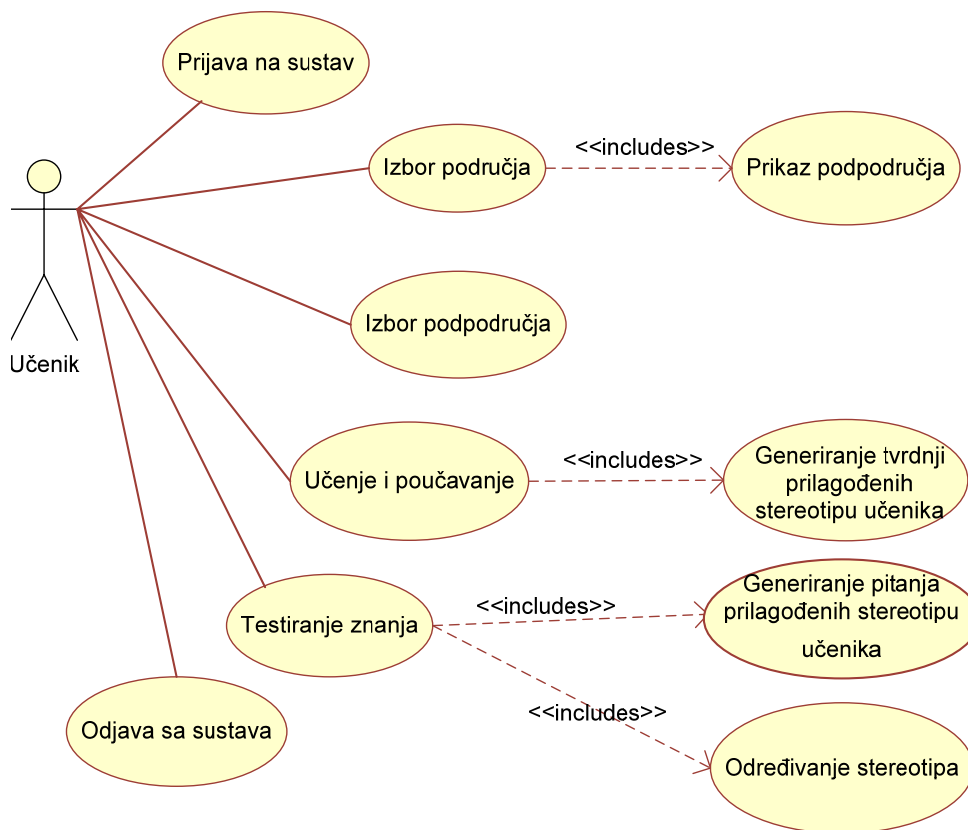
Implementirana prototipna verzija je podvrgnuta eksperimentu kojim se istražuje osjećaj "zadovoljstva" u radu s AC-ware Tutor. Za provedbu eksperimenta odabrano je akcijsko istraživanje ((Kemmis, McTaggart, & Program, 1988), (Cohen, Manion, & Morrison, 2007)). U akcijskoj fazi akcijskog istraživanja (McNiff & Whitehead, 2002) ispitanici su koristili prototip AC-ware Tutor. Promatranje u istraživanju je odrađeno metodom ankete. Tijek istraživanja, okruženje i dobiveni rezultati su opisani u podpoglavlju 5.2.

### 5.1 Arhitektura prototipa sustava

Arhitektura prototipa je u ovom podpoglavlju pojašnjena pomoću UML dijagrama. Dijagramima se opisuje statička struktura prototipa, odnosno struktura komponenata i klasa. Za modeliranje relacija baza podataka koristi se E-R dijagram.

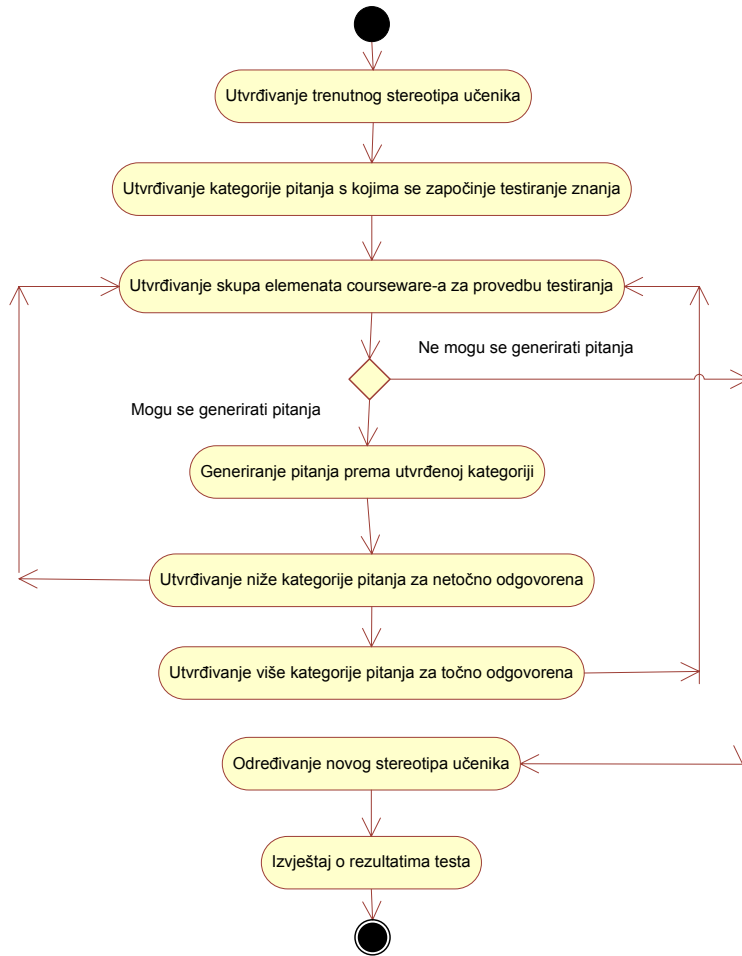
### 5.1.1 Osnovni UML dijagrami

Struktura svakog sustava se najbolje vidi kroz opis sudionika i funkcionalnosti koje koriste sudionici u sustavu. Jedini sudionik ovog sustava je učenik. Učenik se može prijaviti na sustav, izabrati područje i podpodručje, učiti se i poučavati, te testirati svoje znanje (Slika 5.1).



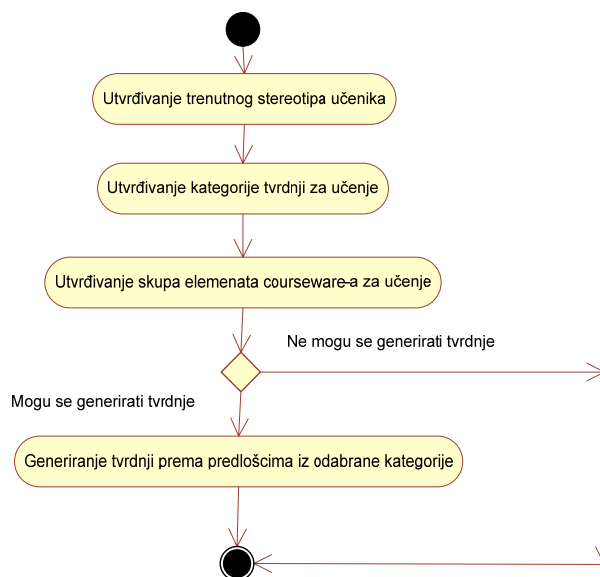
Slika 5.1. Dijagram slučajeva korištenja

Prva funkcionalnost koja se detaljnije opisuje je postupak testiranja znanja (Slika 5.2). Prije početka testa utvrđuje se postojeći stereotip učenika, ako učeniku nije dodijeljen stereotip, tada se definira kao Osrednji. Za svaki stereotip se započinje od odgovarajuće težinske kategorije pitanja, koja je uvijek za razinu viša od utvrđene razine, osim u slučaju stručnjaka gdje se počinje sa najvišom kategorijom pitanja. Nakon utvrđivanja stereotipa i početne težinske kategorije, pitanja se dinamički generiraju. Na sva ponuđena pitanja učenik mora ponuditi odgovor, dok ne ponudi odgovor ne može dobiti slijedeću grupu pitanja ili rezultate. U slučaju mogućnosti generiranja novih pitanja, ponovo se generira grupa pitanja. U slučaju nemogućnosti stvaranja novih pitanja, test se završava, učeniku se dodjeljuje stereotip i prikazuje izvještaj sa testa.



Slika 5.2. Testiranje znanja

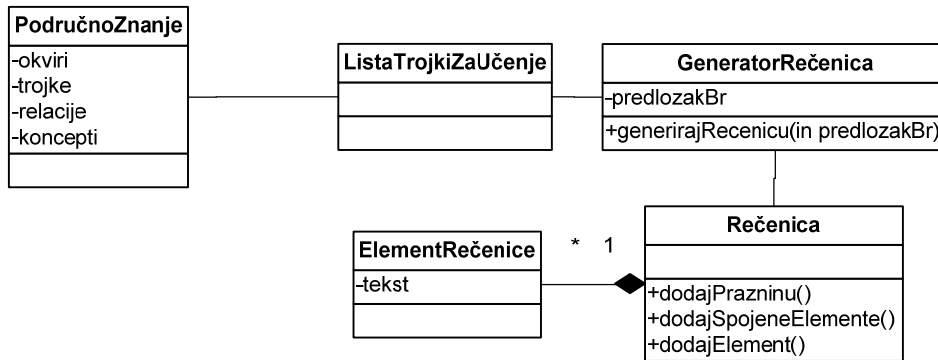
Još ćemo detaljnije prikazati proces generiranja tvrdnji prilagođenih stereotipu učenika (Slika 5.3).



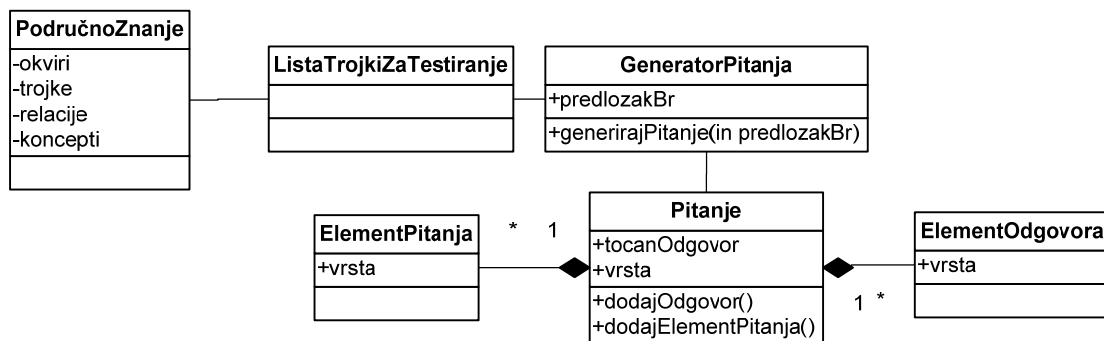
Slika 5.3. Generiranje tvrdnji za učenje

Dijagrami klasa predstavljaju vizualnu specifikaciju objekata koji postoje u sustavu i odnose između njih. Njima se može specificirati konceptualni i implementacijski dijelovi sustava.

Klase ovog sustava prikazujemo kroz dva dijagrama klasa: dijagram klasa za funkcionalnost učenje (Slika 5.4) i dijagram klasa za funkcionalnost testiranje (Slika 5.5).



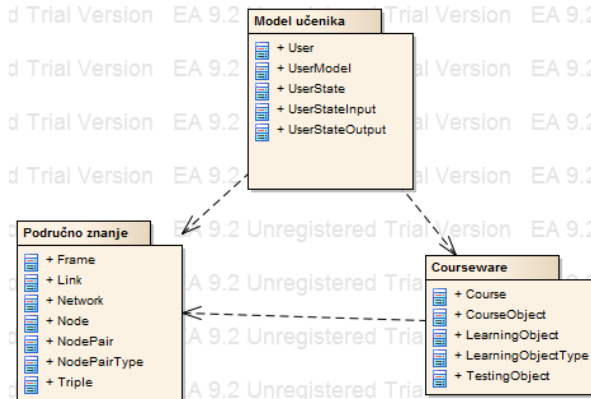
Slika 5.4. Dijagram klasa za učenje u modulu Učitelja



Slika 5.5. Dijagram klasa za testiranje u modulu Učitelja

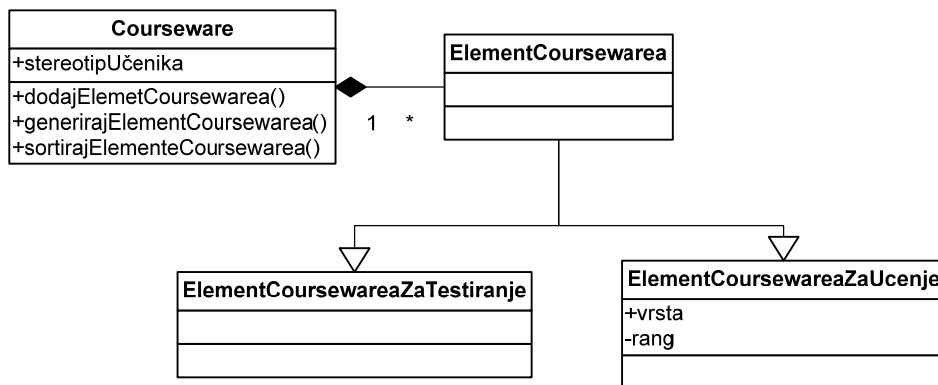
## 5.1.2 Organizacija skupova podataka

Modeli skupova podataka, su implementirani i organizirani po paketima (Slika 5.6). Objekti računalom oblikovanog nastavnog sadržaja se referenciraju na koncepte područnog znanja, a klase modela učenika iz paketa MU se povezuju s klasom objekta *Courseware* paketa NS i klasom koncepta iz paketa PZ.

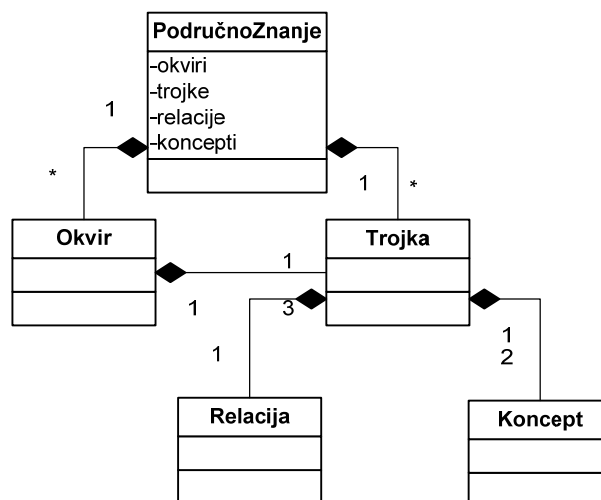


Slika 5.6. Paketi

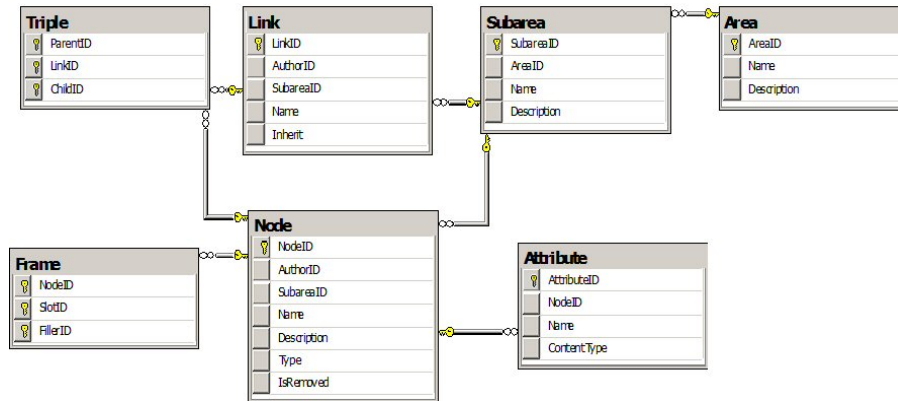
Svaki paket sadrži klase koje modeliraju određeni skup podataka. Paket *Courseware* je detaljnije opisan dijagramom klasa (Slika 5.7). Paketi *Model učenika* i *Područno znanje* su popraćeni dijagramom klasa (Slika 5.8, Slika 5.10) i relacijskom strukturom tablica baze podataka (Slika 5.9, Slika 5.11).



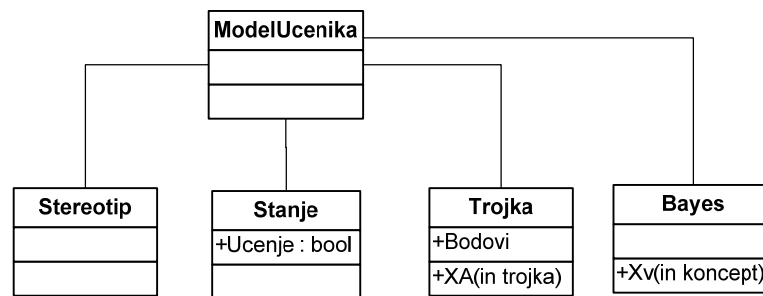
Slika 5.7. Klase modula Učitelja



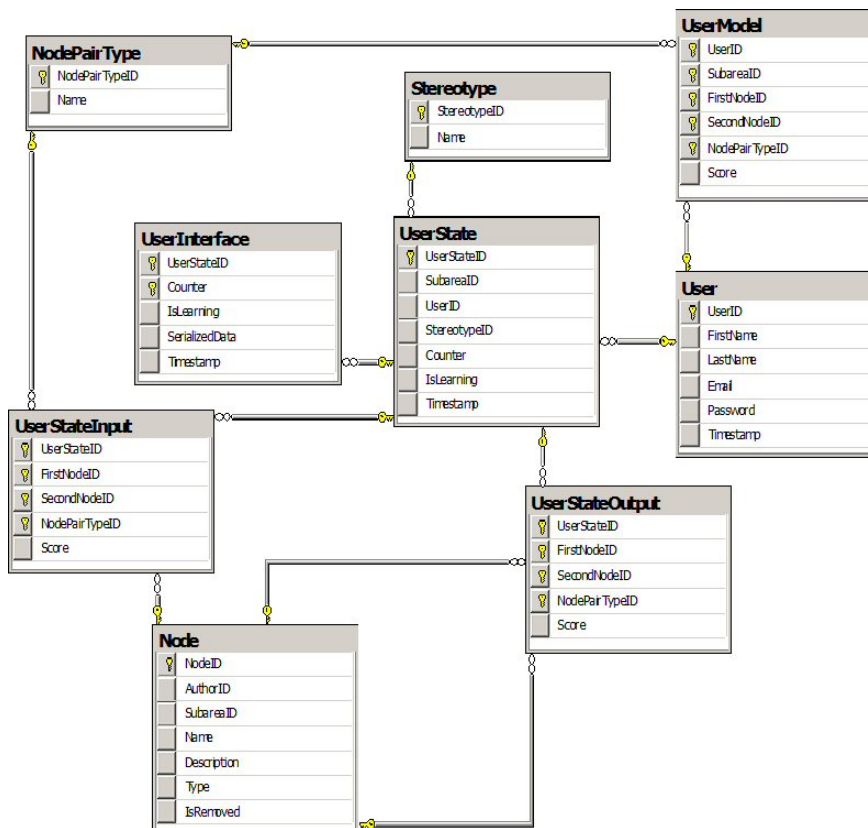
Slika 5.8. Klase modula Stručnjaka



Slika 5.9. Relacijska struktura za bazu podataka PodručnoZnanje



Slika 5.10. Klase modula Učenika

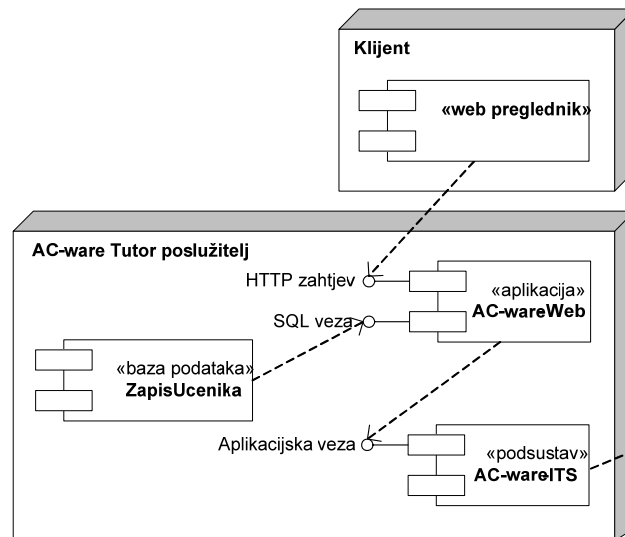


Slika 5.11. Relacijska struktura za bazu podataka ZapisUčenika



### 5.1.3 Komponente sustava

Tehnička podrška na kojoj se zasniva prototip sustava sastoji se od poslužiteljskog računala i klijentskog računala. Komunikacija među računalima se zasniva na HTTP zahtjevima. Klijentsko računalo pomoću Web preglednika pristupa uslugama sustava. Organizacija tehničke podrške i povezanost njihovih komponenti su prikazani na Slika 5.12.



Slika 5.12. Dijagram komponenti prototipa sustava

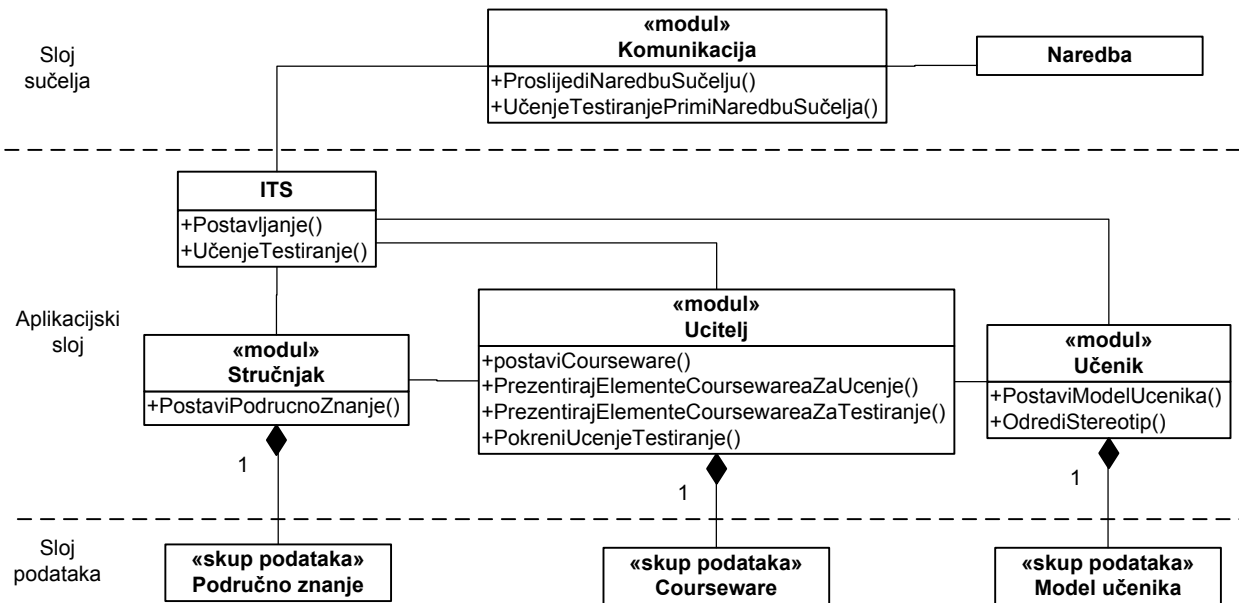
Baza podataka, podsustav i aplikacija su stereotipovi komponentata postavljenih na poslužitelju.

*AC-ware-ITS* je podsustav bez klasa koje implementiraju sučelje prema korisniku. U *AC-ware-Web* aplikaciji je implementirano korisničko sučelje. Ujedno, preko ove aplikacije se u relacijskoj bazi podataka *ZapisUcenika* trajno pohranjuju podaci o prijavljenim učenicima, povijest njihovog učenja i testiranja i modeli učenika (detaljno opisano u prethodnom potpoglavlju).

Podsustavi i aplikacija prototipa sustava su građeni u trorednoj arhitekturi. Njihova arhitektura je opisana dijagramima klasa.

#### 5.1.3.1 Podsustav AC-ware-ITS

*AC-ware-ITS* u aplikacijskom sloju sadrži module stručnjaka, učitelja i učenika. Sloj podataka sadrži klase kojima se opisuju skupovi podataka, kao što su područno znanje, računalom oblikovani nastavni sadržaj i model učenika. U sloju sučelja se nalazi komunikacijski modul koji preko naredbi (klasa *Naredba*) se ostvaruje veza s aplikacijom *AC-ware-Web* (Slika 5.13).



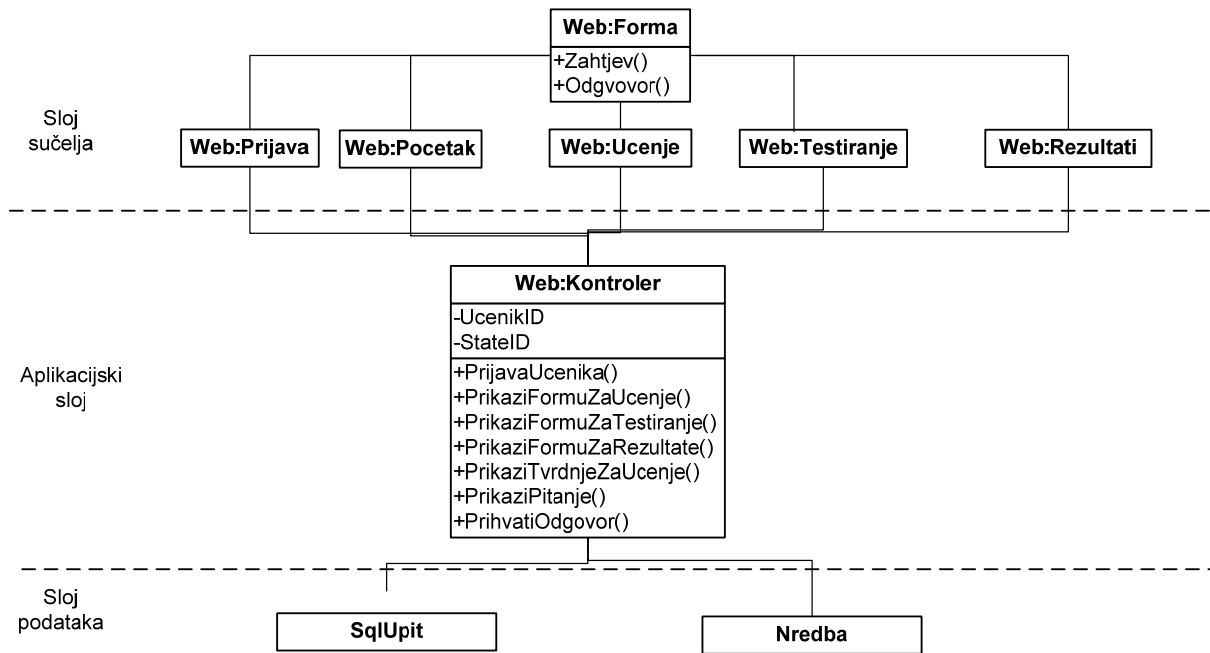
Slika 5.13. Dijagram klasa podsustava AC-ware-ITS

Inicijalizacijom *AC-ware-ITS-a* se prvo stvara primjerak klase *ITS*, a zatim primjerci klase koji predstavljaju module inteligentnog tutorskog sustava. Također započinje faza postavljanja, u kojoj se poziva metoda *Postavljanje()* klase *ITS* kojom se postavlja područno znanje (klasa *PodrucnoZnanje*), postavlja računalom oblikovani nastavni sadržaj (klasa *Courseware*) i postavlja model učenika (klasa *ModelUčenika*). Navedeni skupovi podataka su organizirani po paketima čiji su odnosi opisani u poglavlju 5.1.2.

Faza učenja i testiranja započinje pokretanjem metode *UčenjeTestiranje()* klase *ITS*, koja zatim poziva metodu *PokreniUčenjeTestiranje()* modula učitelja. Ova metoda, ovisno o tome koji je element računalom oblikovanog nastavnog sadržaja na redu, pokreće metodu *PrezentirajElementeCoursewareaZaUčenje()* (klasa *GeneratorRecenica*) ili *PrezentirajElementeCoursewareaZaTestiranje()* (klasa *GeneratorPitanja*). Metoda *OdrediStereotip()* modula učenika na temelju modela učenika računa vrijednosti funkcije  $X_V()$ , na temelju koje se određuje kojem stereotipu učenik pripada.

### 5.1.3.2 Aplikacija AC-ware-Web i baza podataka ZapisUčenika

Korisničko sučelje prema učeniku je ostvareno *AC-ware-Web* aplikacijom. Sloj sučelja sadrži klase kojima su implementirane dinamičke Web stranice. Dinamička Web stranica funkcionira na principu zahtjeva i odgovora. Za interpretaciju zahtjeva i vraćanje odgovora je odgovorna kontrolna klasa u aplikacijskom sloju. Dijagram klasa aplikacije *AC-ware-Web* je prikazan na slici.



Slika 5.14. Dijagram klasa aplikacije AC-ware-Web

Sloj sučelja sadrži pet klasa kojima se implementiraju pojedine dinamičke Web stranice. U Tablica 5.1 je opisan sadržaj dinamičkih Web stranica.

Tablica 5.1. Dinamičke Web stranice aplikacije AC-ware-Web

Klasa	Sadržaj	Izgled stranice
Prijava	unos korisničkog imena, prijava korisnika	Slika 5.15
Početak	odabir područja I podpodručja	Slika 5.16
Ucenje	prikaz prilagođenih tvrdnji za odabrani koncept,	Slika 5.17
Testiranje	prikaz prilagođenog testa znanja, unos odgovora	Slika 5.18
RezultatTesta	prikaz rezultata testiranja putem dijaloga	Slika 5.19

Temeljna klasa u sloju sučelja je *Forma* koja sadrži osnovne metode za ostvarivanje HTTP zahtjeva i HTTP odgovora prema Web poslužitelju.

## AC-ware Tutor

Korisničko ime: Maja  
 Lozinka: ●  
 Zapamti me.

Prototipna verzija (c) 2011

Slika 5.15. Prijava u sustav

## AC-ware Tutor

(Osrednji) Maja Matas Odjava

Prije testa morate izabrati područje i podpodručje iz kojeg pristupate testu.

Područje:   
 Podpodručje:

Za područje **Računarstvo** i podpodručje **Računalo kao sustav** vi pripadate stereotipu: **Osrednji**

Nakon izbora područja i podpodručja možete započeti sa kvizom. Svako pitanje morate predati da bi dobili slijedeće pitanje. Pitanja i odgovori se generiraju dinamički pa broj pitanja nije unaprijed određen.

Nakon završetka testa dobit ćete izvještaj sa pitanjima i odgovorima kao i vaš stereotip. U testu se ispituju koncepti koji su **neposredno** povezani.

Kada ste spremni kliknite na .

Prototipna verzija (c) 2011

Slika 5.16. Odabir područja i podpodručja i pozdravna poruka

## AC-ware Tutor

(Novak) Maja Matas Odjava

Kategorija: 1  
 Računalni sustav izvršava Temeljna funkcija računala  
 Temeljna funkcija računala je popdkoncept od Računalni sustav

Kategorija: 1  
 Programska podrška ima podvrstu Aplikacijska programska podrška  
 Programska podrška je nadkoncept od Aplikacijska programska podrška

Kategorija: 1  
 Tehnička podrška ima podvrstu Centralna jedinica  
 Centralna jedinica je popdkoncept od Tehnička podrška

Prototipna verzija (c) 2011

Slika 5.17. Učenje i poučavanje

**AC-ware Tutor** (Osrednji) Maja Matas [Odjava](#)

1. (3) Poveži zadane koncepte

Tehnička podrška

Centralna procesorska jedinica

Centralna jedinica

Računalo

Računalni sustav

---

2. (3) Poveži zadane koncepte

Upravljačka jedinica

Logička operacija

Instrukcija

Logička operacija

Operacija

Slika 5.18. Dio pitanja iz testa

**AC-ware Tutor** (Novak) Maja Matas [Odjava](#)

Bodovi : 5/15 (33%)

Na testu ste pokazali da je vaše znanje nepromijenjeno - i dalje ste Novak.

[Učenje](#)

1. Da li je Računalni sustav nadkoncept od Obrada podataka ? Ne

Da

Ne

Ne znam

---

2. Da li su Temeljna funkcija računala i Obrada podataka neposredno povezani ? Da

Da

Ne

Ne znam

Slika 5.19. Povratna informacija o točnosti odgovora na testu

Za izmjenu dinamičkih Web stranica i za prikazivanje sadržaja unutar svake stranice je odgovorna *Kontroler* klasa. Ova klasa prima naredbe od modula učitelja *AC-ware-ITS* podsustava i na osnovu njih vrši izmjenu i prikaz sadržaja stranica.

### 5.1.4 Implementacija i postavljanje prototipa sustava

Prototip sustava je razvijen u skladu s modelom opisanom u poglavlju 4. Realizacija prototipne verzije sustava je ostvarena Microsoft-ovom .NET 3.5 tehnologijom korištenjem koje su implementirane komponente, njihova sučelja i skupovi podataka. Izvorni kôd je pisan u objektno-orientiranom programskom jeziku C#, a za formiranje SQL upita, strukturiranje i pretraživanje XML datoteka i memorijskih kolekcija, korišten je funkcionalni programski jezik LINQ (Language INtegrated Query) ((Marguerie, Eichert, & Wooley, 2008), (Rattz, 2007)). LINQ omogućava pisanje upita u istom jeziku u kojem je i ostatak koda, provjeru ispravnosti upita već prilikom prevođenja, a ne tek prilikom izvođenja programa, te pisanje upita nad različitim izvorima podataka u istom jeziku. LINQ upiti (nad bazama podataka) se prevode u SQL upite i kao takvi šalju bazi podataka. Sintaksa LINQ upita i struktura podataka je temeljena na C# jeziku. Pomoću LINQ jezika je poprilično pojednostavljena sintaksa algoritama, a korištenjem anonimnih i generičkih tipova podataka je ujedno olakšano i oblikovanje struktura podataka.

Prototip sustava uključuje relacijske baze podataka implementirane korištenjem Microsoft SQL 2008 platforme. Veza između aplikacijskih komponenti sustava i Microsoft SQL 2008 poslužitelja je ostvarena integriranom komponentom .NET platforme. Web usluga podsustava je također temeljena na integriranim komponentama .NET platforme. Korisničko sučelje, temeljeno na dinamičkim Web stranicama, je ostvareno ASP.NET tehnologijom, a korišteni Web poslužitelj je IIS (Internet Information Services). Kako bi se izbjeglo osvježavanje cijele Web stranice, upotrijebila se AJAX (Asynchronous Javascript + Xml) funkcionalnost implementirana kao proširenje ASP.NET tehnologije (Evjen, Gibbs, Wahlin, & Reed, 2011).

Web aplikacija, Web usluga i relacijske baze podataka prototipne verzije sustava su postavljene na Microsoft Windows 2003 Server poslužitelju.

### 5.1.5 Način korištenja prototipa

Svaki učenik prije početka rada na sustavu je unio korisničko ime i zaporku radi legalizacije rada na sustavu (Slika 5.15).

Nakon prijave, prikazuje se pozdravna poruka i opcije koje ima na raspolaganju. Također, učenik dobiva kratko uputstvo u kojem su opisani osnovni pojmovi koji će se spominjati tijekom procesa učenja i poučavanja u sustavu AC-ware Tutor. Ako se učenik prvi put prijavio, onda mu je jedina opcija ulazni test (Slika 5.16). Ulaznim testom se određuje početni stereotip.

Nakon ulaznog testa (Slika 5.18) slijedi individualizirano učenje prema računalom oblikovanom nastavnom sadržaju oblikovanom u skladu s početnim stereotipom svakog pojedinog učenika. Tijekom učenja učeniku se prezentiraju tvrdnje koje su prilagođene njegovom stereotipu i trenutnom znanju koje je pohranjeno u modelu učenika. Svaka faza učenja završava testiranjem poznavanja dijela područnog znanja koje je bilo prikazano na temelju pridijeljenog stereotipa.

Na Slika 5.17 su prikazane tvrdnje koje je morao naučiti učenik koji je nakon ulaznog testa dobio stereotip Novak, u prvom ciklusu učenja. Tvrdnje su generirane na temelju nastavnih jedinica Računalo kao sustav, Tehnička podrška, Programska podrška i Temeljna funkcija računala koje čine podgraf prikazan na Slika 5.20.



Slika 5.20. Podskup područnog znanja za učenje

Pri dnu forme za učenje prikazuje se opcija za pokretanje testa kojim se provjerava poznavanje upravo prezentiranog dijela područnog znanja. Provjeravanje znanja se vrši pitanjima koja su prilagođena trenutnom stereotipu učenika (Slika 5.21). Nad konceptima i relacijama iz pitanja koje je učenik točno riješio generiraju se pitanja iz veće težinske kategorije (Slika 5.22).

AC-ware Tutor
(Novak) Maja Matas [Odjava](#)

---

1. (1) Da li je Računalski sustav nadkoncept od Obrada podataka ?

Da

Ne

Ne znam

---

2. (1) Da li su Temeljna funkcija računala i Obrada podataka neposredno povezani ?

Da

Ne

Ne znam

---

3. (1) Da li je Prikazivanje podataka podkoncept od Temeljna funkcija računala ?

Da

Ne

Ne znam

---

Slika 5.21. Prvi ciklus provjeravanja znanja za stereotip novak s pitanjima iz prve težinske kategorije

AC-ware Tutor
(Novak) Maja Matas [Odjava](#)

1. (2) Da li su Sistemska programska podrška i Programska podrška posredno povezani?

Da

Ne

Ne znam

---

2. (2) Da li su Unos podataka i Temeljna funkcija računala posredno povezani?

Da

Ne

Ne znam

---

Rezultati

Prototipna verzija (c) 2011

Slika 5.22. Drugi ciklus provjeravanja znanja za stereotip novak s pitanjima iz druge težinske kategorije

Na kraju testiranja učenik dobiva povratnu informaciju o točnosti odgovora na postavljena pitanja, broj ostvarenih bodova i postignuti stereotip (Slika 5.19).

Ukoliko sustav ustanovi da je učenik usvojio sve relacije iz područnog znanja na razini koja zadovoljava određeni stereotip, onda se učeniku ponudi opcija završavanja procesa učenja i poučavanja s ostvarenim stereotipom ili nastavak učenja i poučavanja s ciljem postizanja boljeg stereotipa (Slika 5.23).

AC-ware Tutor
(Ekspert) Maja Matas [Odjava](#)

Bodovi : 21/122 (17%)

Na testu ste pokazali da je vaše znanje nepromijenjeno - i dalje ste Ekspert.

Tijekom proteklog procesa učenja i poučavanja, pokazali ste poznavanje područja Računalo kao sustav na razini Ekspert.

Ako ste zadovoljni s tom razinom znanja, možete odabrati opciju KRAJ ili možete nastaviti s procesom učenja i poučavanja i postići razinu STRUČNJAKA.

Učenje Kraj

1. Poredaj:

<input type="text" value="Logički sklop"/>	Logički sklop
<input type="text" value="NI sklop"/>	NI sklop
<input type="text" value="Konjukcija"/>	Konjukcija

Slika 5.23. Mogućnost završavanja s procesom učenja i poučavanja na kao stereotip koji nije stručnjaka

U bilo kojem trenutku učenik se može odjaviti sa sustava. Rezultati i povijest učenja i testiranja su zapisani u relacijskoj bazi podataka ZapisUcenika.

Sljedeće poglavlje opisuje eksperimentalne skupine, okruženje za provedbu eksperimenata, kao i rezultate dobivene provedenim prototipnim testiranjem sustava AC-ware Tutor.



## 5.2 Vrednovanje prototipa sustava

Prototipna testiranja su obavljena u akcijskom istraživanju tijekom završne faze izrade doktorske disertacije s temeljnim ciljem istraživanja i vrednovanja "zadovoljstva" studenata u radu sa sustavom. Osim toga, željelo se kvalitativno prikazati rezultat modeliranja učenika iskazan postignutim završnim stereotipom nakon provedenog testiranja. U funkciji tako postavljenog cilja oblikovan je i obrađen anketni upitnik koje su ispunjavali studenti sudionici prototipnih testiranja. Prototipna testiranja su provedena na dvije skupine sa studentima na Prirodoslovno matematičkom fakultetu u Splitu i na Fakultetu prirodoslovno-matematičkih i odgojnih znanosti Sveučilišta u Mostaru. Posebno se prikazuje tijek prototipnih testiranja i analiziraju postignuti rezultati po skupinama studenata.

Studenti su u okviru od jednog nastavnog sata upoznati sa zamisli e-učenja i sustava e-učenja i posebice procesom učenja, poučavanja i testiranja znanja pomoću sustava. U drugom nastavnom satu studenti su se poučavali i testirali svoje znanje. Prototip sustava je testiran nad područnim znanjem "Računalo kao sustav". Područno znanje je obuhvatilo osnovne koncepte i raščlambu znanja računalo kao sustav i uključuje 88 koncepata, 9 relacija, 91 uređenu trojku koncept-relacija-koncept, te 9 okvira. Popis svih koncepata područnog znanja se nalazi u Prilogu 8.2.1.

Na kraju prototipnih testiranja studenti su ispunjavali anketni upitnik o „zadovoljstvu“ u radu sa sustavom (Prilog 8.7). Upitnik je proveden korištenjem online sustava za provedbu anketnih upitnika ([www.inovacije.eu](http://www.inovacije.eu)) (<http://inovacije.eu/ankete/index.php?sid=95768&lang=hr>).

Pitanja anketnog upitnika su podijeljena u tri dijela. U prvom dijelu ispitivano je "zadovoljstvo" u korištenju AC-ware Tutor-a za vrijeme učenja i testiranja. Drugim dijelom je ispitan odnos učenja i testiranja znanja uz pomoć AC-ware Tutor-a i tradicionalne nastave. U trećem dijelu je ocijenjen AC-ware Tutor.

### 5.2.1 Prvo prototipno testiranje

Prototipno testiranje je obavljeno sa četrnaest studenata prve godine preddiplomskog studija studijske grupe Matematika na kolegiju *Uvod u računarstvo* na Prirodoslovno matematičkom fakultetu Sveučilišta u Splitu. Kolegij se odvijao u zimskom semestru akademske 2011/2012. godine. Prototipno testiranje je provedeno u okviru redovite nastave u terminu vježbi i u trajanju od dva nastavna sata na dan 13.12.2011., kao i anketni upitnik o "zadovoljstvu" u radu sa sustavom.

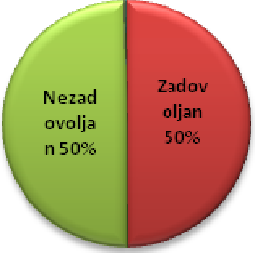
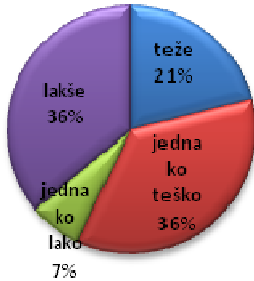
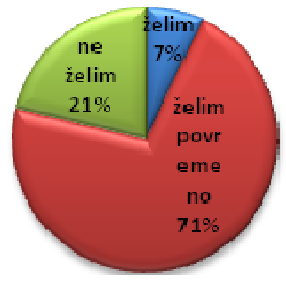
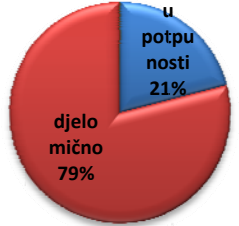
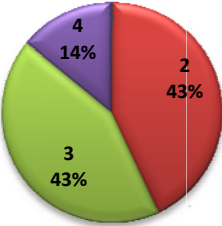
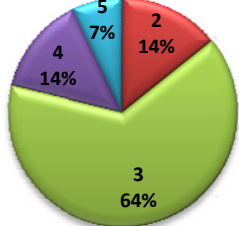
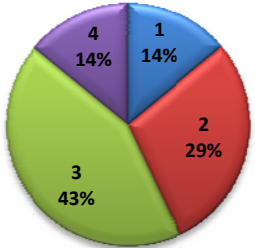
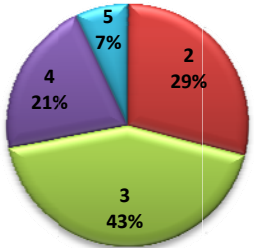
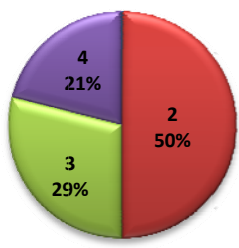
Studenti iz ove skupine nisu imali nikakvo predznanje o sustavima e-učenja, ali su zato imali jako dobro predznanje iz područja „Računalo kao sustav“, te su kao takvi mogli dobro

ocijeniti kvalitetu sustava AC-ware Tutor s obzirom na zadovoljstvo njegovim korištenjem u svakodnevnom procesu učenja i poučavanja.

### 5.2.1.1 Analiza anketnog upitnika

U *prvom dijelu* anketnog upitnika je ispitano zadovoljstvo prilikom učenja i testiranja uz pomoć AC-ware Tutor-a, a rezultati su prikazani u Tablica 5.2.

Tablica 5.2. Pitanja i odgovori prvog dijela anketnog upitnika

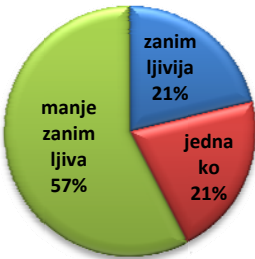
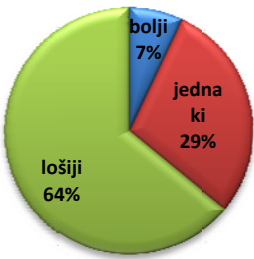
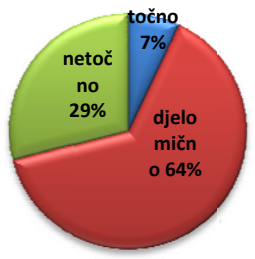
Pitanje 1	Pitanje 2	Pitanje 3
<p>Kako biste izrazili zadovoljstvo/nezadovoljstvo sustavom?</p> 	<p>Kako je vrijeme odmicalo, učenje računarstva uz pomoć AC-ware Tutor-a mi je bilo:</p> 	<p>Želite li nastaviti učiti računarstvo uz pomoć AC-ware Tutor-a?</p> 
Pitanje 4	Pitanje 5	Pitanje 6
<p>Nastavni sadržaj računarstva obrađen uz pomoć AC-ware Tutor-a razumio/la sam:</p> 	<p>Ocijenite koliko vam sustav AC-ware Tutor olakšava učenje.</p> 	<p>Ocijenite koliko je sustav AC-ware Tutor jednostavan za korištenje.</p> 
Pitanje 7	Pitanje 8	Pitanje 9
<p>Ocijenite zanimljivost načina tumačenja gradiva u sustavu AC-ware Tutor.</p> 	<p>Ocijenite razumljivost nastavnih sadržaja u sustavu AC-ware Tutor.</p> 	<p>Ocijenite razumljivost pitanja u testovima u sustavu AC-ware Tutor.</p> 

Studenti su podjednako zadovoljni, odnosno nezadovoljni sustavom AC-ware Tutor. Kako je vrijeme odmicalo, učenje je većini studenata bilo jednako teško ili lako, a trećini njih jednostavnije nego na početku. Neznatnom broju studenata je učenje bilo teže. Skoro svi studenti su izjavili kako žele povremeno nastaviti učiti uz pomoć AC-ware Tutor-a. Većina studenata je sadržaje u AC-ware Tutor-u razumjela djelomično, dok je ostatak razumio u potpunosti. Niti jedan student nije izjavio nerazumijevanje. Otprilike 40% studenata smatra da im sustav AC-ware Tutor niti ne olakšava niti otežava učenje, mali dio studenata misli da im olakšava, dok 40% misli da im otežava učenje. Oko dvije trećine studenata misli da ovaj sustav nije niti jednostavan niti težak za korištenje. Otprilike 40% studenata smatra da način tumačenja u sustavu AC-ware Tutor nije niti zanimljiv niti dosadan, dok oko 30% misli da je način tumačenja gradiva dosadan, odnosno zanimljiv. Isti udjeli studenata se odnose i na razumljivost nastavnih sadržaja. Polovina studenata misli da su pitanja u testovima nerazumljiva, dok ih 30% smatra da su niti razumljiva niti nerazumljiva, a 20% ih smatra razumljivima.

Na osnovi ovog dijela upitnika može se zaključiti kako su studenti uglavnom razumjeli računalom oblikovani nastavni sadržaj obrađen uz pomoć AC-ware Tutor-a i općenito bi željeli i dalje koristiti AC-ware Tutor u nastavi.

U *drugom djelu* je uspoređeno učenje na klasičnoj nastavi i uz pomoć AC-ware Tutor-a. Tablica 5.3 prikazuje anketna pitanja i postotke odgovora.

Tablica 5.3. Pitanja i odgovori drugog dijela anketnog upitnika

Pitanje 10	Pitanje 11	Pitanje 12
Nastava računarstva uz pomoć AC-ware Tutor-a je u odnosu na klasičnu nastavu.	Rezultati u učenju računarstva uz pomoć AC-ware Tutor-a su u odnosu na rezultate koje sam postizao/la ranije.	Korištenjem AC-ware Tutor-a u nastavi mogla bi se ostvariti kvalitetnija nastava.
		

Polovina studenata smatra kako je učenje uz pomoć AC-ware Tutor-a u odnosu na klasičnu nastavu zanimljivije ili jednako zanimljivo. Rezultati u učenju računarstva uz pomoć AC-ware Tutora su lošiji za trećinu studenata, dok su za trećinu jednaki ili bolji. Ipak, većina studenata smatra da bi se se uvođenjem AC-ware Tutor-a korisno nadopunila klasična nastava.

Trećim dijelom anketnog upitnika je ocijenjen AC-ware Tutor, te je analizirano općenito mišljenje o sustavu. Tablica 5.4 prikazuje ocjenu AC-ware Tutor-a.

Tablica 5.4. Pitanje i odgovor trećeg dijela anketnog upitnika

Pitanje 13	Pitanje 14	Pitanje 15																														
Ocijenite koliko sustav AC-ware Tutor zadovoljava vaše potrebe.	Ocijenite preglednost sustava AC-ware Tutor.	Biste li preporučili AC-ware Tutor svojim prijateljima i suradnicima?																														
<table border="1"> <tr><th>Kategorija</th><th>Postotak</th></tr> <tr><td>zadovoljavanje</td><td>80%</td></tr> <tr><td>manje od</td><td>21%</td></tr> <tr><td>ne zadovoljavanje</td><td>7%</td></tr> </table>	Kategorija	Postotak	zadovoljavanje	80%	manje od	21%	ne zadovoljavanje	7%	<table border="1"> <tr><th>Kategorija</th><th>Postotak</th></tr> <tr><td>ovisno o području</td><td>50%</td></tr> <tr><td>nepregledan</td><td>21%</td></tr> <tr><td>vrlo pregledan</td><td>14%</td></tr> <tr><td>pregledan</td><td>7%</td></tr> <tr><td>vrlo nepregledan</td><td>7%</td></tr> </table>	Kategorija	Postotak	ovisno o području	50%	nepregledan	21%	vrlo pregledan	14%	pregledan	7%	vrlo nepregledan	7%	<table border="1"> <tr><th>Kategorija</th><th>Postotak</th></tr> <tr><td>vjerojatno da</td><td>36%</td></tr> <tr><td>ne znam</td><td>29%</td></tr> <tr><td>vjerojatno ne</td><td>29%</td></tr> <tr><td>svakako ne</td><td>7%</td></tr> </table>	Kategorija	Postotak	vjerojatno da	36%	ne znam	29%	vjerojatno ne	29%	svakako ne	7%
Kategorija	Postotak																															
zadovoljavanje	80%																															
manje od	21%																															
ne zadovoljavanje	7%																															
Kategorija	Postotak																															
ovisno o području	50%																															
nepregledan	21%																															
vrlo pregledan	14%																															
pregledan	7%																															
vrlo nepregledan	7%																															
Kategorija	Postotak																															
vjerojatno da	36%																															
ne znam	29%																															
vjerojatno ne	29%																															
svakako ne	7%																															
Pitanje 16	Pitanje 17	Pitanje 18																														
Ocijenite kvalitetu AC-ware Tutor sustava.	Ocijenite funkcionalnost AC-ware Tutor sustava.	Ocijenite jasnoću AC-ware Tutor sustava.																														
<table border="1"> <tr><th>Kategorija</th><th>Postotak</th></tr> <tr><td>3</td><td>71%</td></tr> <tr><td>2</td><td>14%</td></tr> <tr><td>4</td><td>7%</td></tr> <tr><td>1</td><td>1%</td></tr> </table>	Kategorija	Postotak	3	71%	2	14%	4	7%	1	1%	<table border="1"> <tr><th>Kategorija</th><th>Postotak</th></tr> <tr><td>3</td><td>71%</td></tr> <tr><td>4</td><td>14%</td></tr> <tr><td>1</td><td>7%</td></tr> <tr><td>2</td><td>7%</td></tr> </table>	Kategorija	Postotak	3	71%	4	14%	1	7%	2	7%	<table border="1"> <tr><th>Kategorija</th><th>Postotak</th></tr> <tr><td>3</td><td>50%</td></tr> <tr><td>2</td><td>36%</td></tr> <tr><td>4</td><td>7%</td></tr> <tr><td>1</td><td>7%</td></tr> </table>	Kategorija	Postotak	3	50%	2	36%	4	7%	1	7%
Kategorija	Postotak																															
3	71%																															
2	14%																															
4	7%																															
1	1%																															
Kategorija	Postotak																															
3	71%																															
4	14%																															
1	7%																															
2	7%																															
Kategorija	Postotak																															
3	50%																															
2	36%																															
4	7%																															
1	7%																															

Polovina studenata smatra da sustav AC-ware Tutor zadovoljava manje od 50% njihovih potreba, dok 40% smatra da zadovoljava više od 50% njihovih potreba. Samo mali dio studenata smatra da njihove potrebe nisu zadovoljene. Polovina studenata smatra da je sustav pregledan i nepregledan, ovisno o području, trećina smatra da je pregledan, dok samo mali dio misli da je nepregledan. Oko 40% studenata bi preporučilo ovaj sustav svojim prijateljima i suradnicima. Oko 80% studenata smatra da je kvaliteta sustava AC-ware Tutor prosječna ili bolja od prosjeka. Isti je udio studenata koji smatra da je funkcionalnost ovog sustava prosječna ili bolja od prosjeka. Jasnoću sustava AC-ware Tutor je polovina studenata ocijenila kao prosječnu, dok 40% studenata smatra da je jasnoća sustava ispod prosjeka. Samo mali dio studenata smatra da je jasnoća bolja od prosjeka.

Odgovaranjem na pitanja 19, 20 i 21 studenti su navodili što im se sviđa kod sustava AC-ware Tutor, te koje su njegove prednosti i nedostaci. Oko 40% studenata je izjavilo da im se sustav sviđa.

Kao prednosti sustava AC-ware Tutor, studenti su naveli sljedeće: omogućava samostalno učenje i sistematizaciju sadržaja, stalno provjeravanje znanja, različitost pitanja u svakom testu i za svakog studenta, učenje iz rezultata testa, ponavljanje nenaučenih sadržaja, sustavno prelaženje s laganijih sadržaja na teže.

Kao nedostatke sustava AC-ware Tutor, studenti navode: nerazumljivost pitanja i rečenica za učenje, previše teksta, bez slika i realističnijih i sistematičnijih prikaza, nezanimljivo prezentiranje sadržaja, neprimjerenost za usvajanje novih sadržaja.

## 5.2.2 Drugo prototipno testiranje

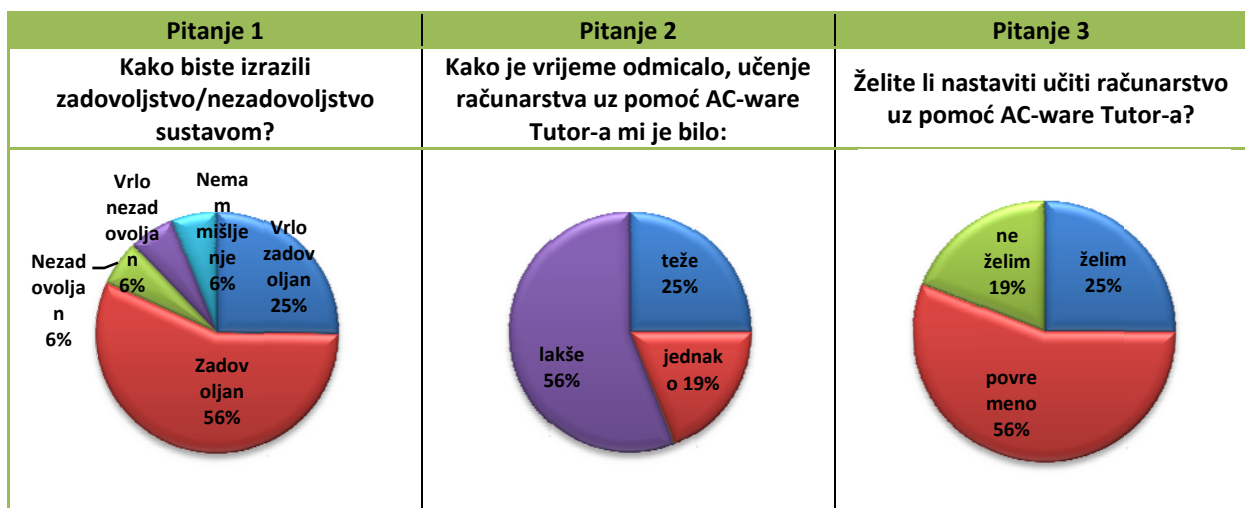
Drugo prototipno testiranje je obavljeno sa šesnaest studenata prve godine diplomskog studija studijske grupe informatike na kolegiju *Sustavi e-učenja* na Prirodoslovno matematičkom fakultetu Sveučilišta u Splitu i na Fakultetu prirodoslovno-matematičkih i odgojnih znanosti Sveučilišta u Mostaru. Kolegij se odvijao u zimskom semestru akademske 2011/2012. godine. Prototipno testiranje je provedeno u okviru redovite nastave u terminu vježbi i u trajanju od dva nastavna sata na dan 13.12.2011., kao i anketni upitnik o "zadovoljstvu" u radu sa sustavom.

Studenti iz ove skupine su upravo učili o različitim sustavima e-učenja, te su kao takvi mogli dobro ocijeniti kvalitetu sustava AC-ware Tutor s obzirom na zadovoljstvo njegovim funkcionalnostima kojima je realiziran proces učenja i poučavanja.

### 5.2.2.1 Analiza anketnog upitnika

U *prvom dijelu* anketnog upitnika je ispitano zadovoljstvo prilikom učenja i testiranja uz pomoć AC-ware Tutor-a, a rezultati su prikazani u Tablica 5.5.

Tablica 5.5. Pitanja i odgovori prvog dijela anketnog upitnika



Pitanje 4	Pitanje 5	Pitanje 6																																		
<p><b>Nastavni sadržaj računarstva obrađen uz pomoć AC-ware Tutor-a razumio/la sam:</b></p> <table border="1"> <caption>Data for Pitanje 4</caption> <thead> <tr> <th>Rating</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>44%</td> </tr> <tr> <td>5</td> <td>56%</td> </tr> <tr> <td>1</td> <td>6%</td> </tr> </tbody> </table>	Rating	Percentage	3	44%	5	56%	1	6%	<p><b>Ocijenite koliko vam sustav AC-ware Tutor olakšava učenje.</b></p> <table border="1"> <caption>Data for Pitanje 5</caption> <thead> <tr> <th>Rating</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>44%</td> </tr> <tr> <td>4</td> <td>31%</td> </tr> <tr> <td>5</td> <td>13%</td> </tr> <tr> <td>1</td> <td>6%</td> </tr> <tr> <td>2</td> <td>6%</td> </tr> </tbody> </table>	Rating	Percentage	3	44%	4	31%	5	13%	1	6%	2	6%	<p><b>Ocijenite koliko je sustav AC-ware Tutor jednostavan za korištenje.</b></p> <table border="1"> <caption>Data for Pitanje 6</caption> <thead> <tr> <th>Rating</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>44%</td> </tr> <tr> <td>4</td> <td>31%</td> </tr> <tr> <td>5</td> <td>13%</td> </tr> <tr> <td>1</td> <td>6%</td> </tr> <tr> <td>2</td> <td>6%</td> </tr> </tbody> </table>	Rating	Percentage	3	44%	4	31%	5	13%	1	6%	2	6%		
Rating	Percentage																																			
3	44%																																			
5	56%																																			
1	6%																																			
Rating	Percentage																																			
3	44%																																			
4	31%																																			
5	13%																																			
1	6%																																			
2	6%																																			
Rating	Percentage																																			
3	44%																																			
4	31%																																			
5	13%																																			
1	6%																																			
2	6%																																			
Pitanje 7	Pitanje 8	Pitanje 9																																		
<p><b>Ocijenite zanimljivost načina tumačenja gradiva u sustavu AC-ware Tutor.</b></p> <table border="1"> <caption>Data for Pitanje 7</caption> <thead> <tr> <th>Rating</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>38%</td> </tr> <tr> <td>5</td> <td>31%</td> </tr> <tr> <td>4</td> <td>19%</td> </tr> <tr> <td>1</td> <td>6%</td> </tr> <tr> <td>2</td> <td>6%</td> </tr> </tbody> </table>	Rating	Percentage	3	38%	5	31%	4	19%	1	6%	2	6%	<p><b>Ocijenite razumljivost nastavnih sadržaja u sustavu AC-ware Tutor.</b></p> <table border="1"> <caption>Data for Pitanje 8</caption> <thead> <tr> <th>Rating</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>4</td> <td>50%</td> </tr> <tr> <td>3</td> <td>31%</td> </tr> <tr> <td>5</td> <td>13%</td> </tr> <tr> <td>1</td> <td>6%</td> </tr> </tbody> </table>	Rating	Percentage	4	50%	3	31%	5	13%	1	6%	<p><b>Ocijenite razumljivost pitanja u testovima u sustavu AC-ware Tutor.</b></p> <table border="1"> <caption>Data for Pitanje 9</caption> <thead> <tr> <th>Rating</th> <th>Percentage</th> </tr> </thead> <tbody> <tr> <td>3</td> <td>38%</td> </tr> <tr> <td>4</td> <td>19%</td> </tr> <tr> <td>5</td> <td>19%</td> </tr> <tr> <td>2</td> <td>19%</td> </tr> <tr> <td>1</td> <td>6%</td> </tr> </tbody> </table>	Rating	Percentage	3	38%	4	19%	5	19%	2	19%	1	6%
Rating	Percentage																																			
3	38%																																			
5	31%																																			
4	19%																																			
1	6%																																			
2	6%																																			
Rating	Percentage																																			
4	50%																																			
3	31%																																			
5	13%																																			
1	6%																																			
Rating	Percentage																																			
3	38%																																			
4	19%																																			
5	19%																																			
2	19%																																			
1	6%																																			

Većina studenata je zadovoljna ili vrlo zadovoljna sustavom AC-ware Tutor. Samo mali broj studenata je iskazao nezadovoljstvo sustavom. Kako je vrijeme odmicalo, učenje je većini studenata bilo jednako teško ili lakše, a četvrtini njih teže nego na početku. Skoro svi studenti su izjavili kako žele povremeno ili stalno nastaviti učiti uz pomoć AC-ware Tutor-a. Oko 60% je sadržaje u AC-ware Tutor-u razumjela djelomično, oko 35% ih je ostatak razumio u potpunosti, a samo mali dio ih nije razumio. Oko 40% studenata smatra da im sustav AC-ware Tutor olakšava, isti udio misli da im niti olakšava niti otežava, dok samo mali dio misli da im otežava učenje. Polovina studenata smatra da je sustav AC-ware Tutor jednostavan za korištenje, trećina studenata misli da ovaj sustav nije niti jednostavan niti težak za korištenje, dok samo mali dio misli da je težak za korištenje. Polovina studenata misli da je način tumačenja gradiva u sustavu AC-ware Tutor zanimljiv, dok trećina studenata smatra da način tumačenja nije niti zanimljiv niti dosadan. Slični udjeli studenata se odnose i na razumljivost nastavnih sadržaja. Oko 40% studenata misli da su pitanja u testovima razumljiva, isti udio smatra da su niti razumljiva niti nerazumljiva, dok mali dio misli da su nerazumljiva.

Na osnovi ovog dijela upitnika može se zaključiti kako su studenti iz ove grupe uglavnom razumjeli računalom oblikovani nastavni sadržaj obrađen uz pomoć AC-ware Tutor-a i općenito bi željeli i dalje koristiti AC-ware Tutor u nastavi.

U *drugom djelu* je uspoređeno učenje na klasičnoj nastavi i uz pomoć AC-ware Tutor-a. Tablica 5.6 prikazuje anketna pitanja i postotke odgovora.

Tablica 5.6. Pitanja i odgovori drugog dijela anketnog upitnika

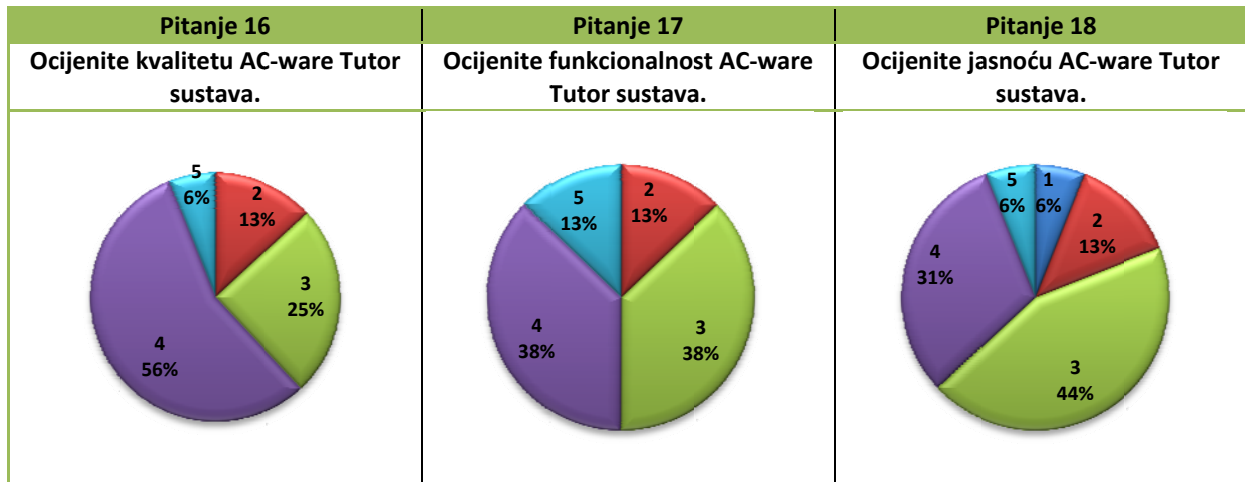
Pitanje 10	Pitanje 11	Pitanje 12																								
Nastava računarstva uz pomoć AC-ware Tutor-a je u odnosu na klasičnu nastavu:	Rezultati u učenju računarstva uz pomoć AC-ware Tutor-a su u odnosu na rezultate koje sam postizao/la ranije:	Korištenjem AC-ware Tutor-a u nastavi mogla bi se ostvariti kvalitetnija nastava.																								
<table border="1"> <tr><th>Kategorija</th><th>Postotak</th></tr> <tr><td>zanimljivija</td><td>56%</td></tr> <tr><td>jednako</td><td>31%</td></tr> <tr><td>manje zanimljiva</td><td>13%</td></tr> </table>	Kategorija	Postotak	zanimljivija	56%	jednako	31%	manje zanimljiva	13%	<table border="1"> <tr><th>Kategorija</th><th>Postotak</th></tr> <tr><td>bolji</td><td>38%</td></tr> <tr><td>jednaki</td><td>38%</td></tr> <tr><td>lošiji</td><td>25%</td></tr> </table>	Kategorija	Postotak	bolji	38%	jednaki	38%	lošiji	25%	<table border="1"> <tr><th>Kategorija</th><th>Postotak</th></tr> <tr><td>djelomično točno</td><td>63%</td></tr> <tr><td>točno</td><td>31%</td></tr> <tr><td>netočno</td><td>6%</td></tr> </table>	Kategorija	Postotak	djelomično točno	63%	točno	31%	netočno	6%
Kategorija	Postotak																									
zanimljivija	56%																									
jednako	31%																									
manje zanimljiva	13%																									
Kategorija	Postotak																									
bolji	38%																									
jednaki	38%																									
lošiji	25%																									
Kategorija	Postotak																									
djelomično točno	63%																									
točno	31%																									
netočno	6%																									

Većina studenata smatra kako je učenje uz pomoć AC-ware Tutor-a u odnosu na klasičnu nastavu zanimljivije ili jednako zanimljivo. Rezultati u učenju računarstva uz pomoć AC-ware Tutora su lošiji za četvrtinu studenata, dok su za tri četvrtine jednaki ili bolji. Većina studenata smatra da bi se se uvođenjem AC-ware Tutor-a korisno nadopunila klasična nastava.

*Trećim dijelom* anketnog upitnika je ocijenjen AC-ware Tutor, te je analizirano općenito mišljenje o sustavu. Tablica 5.7 prikazuje ocjenu AC-ware Tutor-a.

Tablica 5.7. Pitanje i odgovor trećeg dijela anketnog upitnika

Pitanje 13	Pitanje 14	Pitanje 15																																		
Ocijenite koliko sustav AC-ware Tutor zadovoljava vaše potrebe:	Ocijenite preglednost sustava AC-ware Tutor:	Biste li preporučili AC-ware Tutor svojim prijateljima i suradnicima?																																		
<table border="1"> <tr><th>Kategorija</th><th>Postotak</th></tr> <tr><td>zadovoljava</td><td>80%</td></tr> <tr><td>zadovoljava</td><td>44%</td></tr> <tr><td>ne zadovoljava</td><td>13%</td></tr> <tr><td>potpuno zadovoljava</td><td>6%</td></tr> <tr><td>manje od 50%</td><td>6%</td></tr> </table>	Kategorija	Postotak	zadovoljava	80%	zadovoljava	44%	ne zadovoljava	13%	potpuno zadovoljava	6%	manje od 50%	6%	<table border="1"> <tr><th>Kategorija</th><th>Postotak</th></tr> <tr><td>pregledan</td><td>38%</td></tr> <tr><td>vrlo pregledan</td><td>25%</td></tr> <tr><td>ovisno o području</td><td>31%</td></tr> <tr><td>nepregledan</td><td>6%</td></tr> </table>	Kategorija	Postotak	pregledan	38%	vrlo pregledan	25%	ovisno o području	31%	nepregledan	6%	<table border="1"> <tr><th>Kategorija</th><th>Postotak</th></tr> <tr><td>vjerojatno da</td><td>38%</td></tr> <tr><td>svakako da</td><td>25%</td></tr> <tr><td>ne znam</td><td>25%</td></tr> <tr><td>vjerojatno ne</td><td>6%</td></tr> <tr><td>svakako ne</td><td>6%</td></tr> </table>	Kategorija	Postotak	vjerojatno da	38%	svakako da	25%	ne znam	25%	vjerojatno ne	6%	svakako ne	6%
Kategorija	Postotak																																			
zadovoljava	80%																																			
zadovoljava	44%																																			
ne zadovoljava	13%																																			
potpuno zadovoljava	6%																																			
manje od 50%	6%																																			
Kategorija	Postotak																																			
pregledan	38%																																			
vrlo pregledan	25%																																			
ovisno o području	31%																																			
nepregledan	6%																																			
Kategorija	Postotak																																			
vjerojatno da	38%																																			
svakako da	25%																																			
ne znam	25%																																			
vjerojatno ne	6%																																			
svakako ne	6%																																			



Tri četvrtine studenata smatra da sustav AC-ware Tutor zadovoljava 50% do 80% njihovih potreba, dok samo mali dio smatra da zadovoljava manje od 50% njihovih potreba ili da njihove potrebe nisu zadovoljene. Polovina studenata smatra da je sustav pregledan ili vrlo pregledan, jedna četvrtina da pregledan i nepregledan, ovisno o području, te isti udio misli da je nepregledan. Dvije trećine studenata bi preporučilo ovaj sustav svojim prijateljima i suradnicima. Oko 60% studenata smatra da je kvaliteta sustava AC-ware Tutor bolja od prosjeka. Polovina studenata koji smatra da je funkcionalnost ovog sustava bolja od prosjeka. Jasnoću sustava AC-ware Tutor je oko 40% studenata ocijenila kao iznadprosječnu, dok trećina studenata smatra da je jasnoća sustava prosječna. Samo mali dio studenata smatra da je jasnoća manja od prosjeka.

Odgovaranjem na pitanja 19, 20 i 21 studenti su navodili što im se sviđa kod sustava AC-ware Tutor, te koje su njegove prednosti i nedostaci. Većina studenata je izjavila da im se sustav sviđa.

Kao prednosti sustava AC-ware Tutor, studenti su naveli sljedeće: samostalno generiranje pitanja i provjeravanje odgovora, razvijanje takmičarskog duha, davanje povratne informacije, postupno otežavanje prezentiranih sadržaja, može se koristiti od kuće, inteligentan pristup kojim prilagođava sadržaje učeniku, inzistiranje na učenju dok određeni sadržaji uistinu u potpunosti nisu svladani, učenje na greškama, ne omogućava preskakanje pa ne ostaju rupe u određenim dijelovima znanja.

Kao nedostatke sustava AC-ware Tutor, studenti navode: nerazumljivost pitanja i rečenica za učenje, ponavljanje stalno istog gradiva dovodi do pada koncentracije, potrebno je naviknuti se na „tvrdu“ terminologiju koju sustav koristi jer bez toga se ne može učiti, ne sadrži grafički prikaz povezanosti koncepata.





## 6 Zaključak

„Mudrac se umije prilagoditi, kao što se voda prilagođava obliku posude.“

Ming Sin Pao Kien

Zamisao novog modela inteligentnog tutorskog sustava se temelji na tradicionalnoj arhitekturi, ali s bitnim poboljšanjem koje je povezano s prilagođavanjem procesa učenja, poučavanja i testiranja aktualnoj razini znanja učenika. S tim u vezi, prilagođavanje znanju učenika je ostvareno primjenom stereotipova, Bayesovih mreža predikcije znanja i Bloomove taksonomije znanja.

U prilagodljivim sustavima e-učenja, proces učenja, poučavanja i testiranja znanja se prilagođava različitim značajkama učenika. Pošto je učenikovo znanje glavni razlog postojanja sustava e-učenja općenito, usmjerili smo naše istraživanje prema prilagođavanju računalom oblikovani nastavni sadržaj u inteligentnim tutorskim sustavima (posebna vrsta prilagodljivih sustava e-učenja), odnosno proučavamo nizanje sadržaja i složenost prilagođavanja (ŠTO), a ne način prilagodljivog prezentiranja (KAKO), što je karakteristično za sustave prilagodljive hipermedije, kao drugog predstavnika prilagodljivih sustava e-učenja.

Mali je broj inteligentnih tutorskih sustava koji generiraju prilagodljivi računalom oblikovani nastavni sadržaj, s naglaskom na prilagođavanje sadržaja, a ne sučelja. Među tim malobrojnim inteligentnim tutorskim sustavima tražili smo one koji automatski generiraju računalom oblikovani nastavni sadržaj: stvaraju, odabiru, nižu i prezentiraju elemente računalom oblikovanog nastavnog sadržaja (i za učenje i za provjeravanje znanja). Nismo mogli pronaći nijedan sustav koji posjeduje automatsko generiranje elemenata računalom oblikovanog nastavnog sadržaja, dinamički odabir i nizanje elemenata, automatsko generiranje pitanja i ulazni test nad reprezentativnim podskupom područnog znanja, u kojima se prilagođavanje realizira prema znanju učenika, kao njegovoj najvažnijoj značajki. Upravo takav model inteligentnog tutorskog sustava kao i oblikovani prototip testiran na primjerenom uzorku je rezultat istraživanja ostvaren ovom disertacijom.

Naš pristup ispunjava sve zahtjeve koji se postavljaju pred realizaciju prilagođavanja: svaki učenik uči svojim ritmom; prilagođavanje se događa često; svaki učenik može uspješno završiti proces učenja i poučavanja; kada je nešto uspješno naučio, učenik nastavlja dalje; učenika se ne poučava ono što već zna.

Zamisao AC-ware Tutor-a uključuje neke dobre osobine inteligentnih tutorskih sustava s prilagodljivim računalom oblikovanim nastavnim sadržajem. U našem pristupu pratimo

---

novije trendove i formaliziramo **područno znanje** ontološkim pristupom (sustavi ACCT, ASM, iClass, ADOPTA). Kod **modeliranja učenika** kombiniramo model prekrivanja, Bayesov probabilistički model učenika (sustavi AST, KBS) i stereotipove (ASM), čime osiguravamo modeliranje učenika s više aspekata. Složeniji pristupi modeliranju učenika omogućavaju i složenije prilagođavanje učeniku.

Proces **generiranja računalom oblikovanog nastavnog sadržaja** promatramo kroz tri faze: generiranje, odabir i nizanje, prezentiranje. *Prva faza* u generiranju računalom oblikovanog nastavnog sadržaja je generiranje elemenata računalom oblikovanog nastavnog sadržaja, što može raditi učitelj ili se elementi mogu generirati automatski. U većini sustava elemente računalom oblikovanog nastavnog sadržaja generira učitelj, kod nekih ima naznaka automatizma, ali s još velikim utjecajem učitelja, dok je automatizacija prisutna u sustavima DCG, ASM, iClass, PAIGOS. U našem pristupu nema nikakvog utjecaja učitelja na generiranje elemenata računalom oblikovanog nastavnog sadržaja. Elementi računalom oblikovanog nastavnog sadržaja se potpuno automatizirano generiraju na temelju ontologije područnog znanja. Generiraju se elementi različite složenosti, što osigurava kasniju realizaciju prilagodljivosti prema znanju.

*Druga faza* u generiranju računalom oblikovanog nastavnog sadržaja je odabir i nizanje elemenata računalom oblikovanog nastavnog sadržaja. Taj proces se može izvršiti samo jedan put prije nego što učenik započne s procesom učenja i poučavanja (statički) ili se može ponavljati više puta tijekom procesa učenja i poučavanja (dinamički). Upravo dinamika osigurava prilagodljivost računalom oblikovanog nastavnog sadržaja, pa većina sustava ima upravo dinamički odabir i nizanje elemenata. Dinamički pristup je realiziran i u našem modelu.

*Treća faza* u generiranju računalom oblikovanog nastavnog sadržaja se odnosi na prezentiranje elemenata računalom oblikovanog nastavnog sadržaja. Prezentiranje je faza u kojoj se također realizira prilagođavanje računalom oblikovanog nastavnog sadržaja. Najveći broj sustava realizira prilagođavanje prezentiranja elemenata korištenjem tehnika prilagodljive hipermedije, kao što su prilagodljiva podrška navigaciji i prilagodljivo označavanje veza. Iako se za veliki broj sustava navodi da se prilagođavaju prema znanju učenika, to prilagođavanje ne zadire u mijenjanje samog sadržaja elementa računalom oblikovanog nastavnog sadržaja, već se odnosi na odabir i nizanje elementa. Prilagođavanje prezentiranja samog sadržaja elementa prema znanju učenika nije ni u jednom sustavu napravljeno na način da se prezentiranje sadržaj elementa mijenja prema razini znanja koju učenik ima. U našem pristupu razine znanja su definirane prema Bloomovoj taksonomiji znanja, čime smo uveli inovaciju u prilagođavanje računalom oblikovanog nastavnog sadržaja i odmakli se od prilagodljivosti koja je prisutna kod prilagodljive hipermedije i prilagođavanja prema stilu učenja. Prilagođavanje je realizirano uporabom predložaka za učenje.

Prilagođavanje računalom oblikovanog nastavnog sadržaja se može realizirati odabirom i nizanjem elemenata računalom oblikovanog nastavnog sadržaja ili prilagođavanjem

---

prezentiranja elemenata. Kod realizacije prilagođavanja promatramo najmanju granulu računalom oblikovanog nastavnog sadržaja ili područnog znanja koja se može manipulirati. Što je manji element prilagođavanja, to je samo prilagođavanje „finije“ i „osjetljivije“, te u konačnici uspješnije. Kod prilagodljivih sustava e-učenja uglavnom je **element prilagođavanja** jedinica ili stranica ili lekcija ili zadatak koja uključuje više koncepata područnog znanja. Samo je kod dva sustava element prilagođavanja koncept područnog znanja (DCG i ASM). U našem pristupu najmanji element raspoloživ za prilagođavanje je koncept znanja, dakle, „atomska čestica“ znanja i u tom smislu nedjeljiva. Pošto smo se u ovom modelu dotakli najmanjeg mogućeg elementa za prilagođavanje, smatramo da našim pristupom realiziramo maksimalnu prilagodljivost.

**Provjeravanje znanja** učenika je „vrući krumpir“ u sustavima e-učenja. Naime, u većini sustava e-učenja učitelj je taj koji mora osmisliti pitanja za provjeravanje znanja učenika. Nedostaci su u tome što je to mukotrpan posao jer treba ručno unijeti pitanje u sustav, odrediti način ocjenjivanja, navesti točne i netočne odgovore, napisati povratnu informaciju, a onda sve to ponoviti što je moguće više puta jer nije dobro da se pitanja ponavljaju (učenici nauče odgovore napamet). Kod većine prilagodljivih sustava e-učenja ili uopće ne postoji provjeravanje znanja ili pitanja stvara učitelj. U našem pristupu sustav sam generira pitanja u formi zadataka objektivnog tipa (dopunjavanje, jednostavno dosjećanje, jednostruki izbor, povezivanje, sređivanje, višestruki izbor) bez ikakvog uplitanja učitelja, sam pridružuje pitanja u test, te sam vrednuje znanje učenika. Pitanja se generiraju na temelju koncepata i relacija iz ontologije područnog znanja, odnosno nad podskupom ontologije nad kojim se učenik učio i poučavao, uz pomoć predložaka za pitanja. Težina pitanja u testu također je usklađena s razinom znanja učenika (Bloomova taksonomija).

U sustavima koji žele prilagoditi proces učenja i poučavanja učeniku, potrebno je što prije započeti s prilagođavanjem. Međutim, to nije moguće dok god se ne **inicijalizira model učenika**. Inicijalizacija se realizira nekim oblikom ulaznog testa ili upitnika. Kao i kod provjeravanja znanja, i kod ulaznog testa ili upitnika problem je u generiranju samih pitanja, ali i u određivanju podskupa područnog znanja koji bi trebao predstavljati nužno predznanje učenika. Pošto smo već kazali da se u našem pristupu generiranje pitanja i testova realizira automatski, potrebno je još samo objasniti kako odrediti podskup područnog znanja nad kojim će se generirati ulazni test. Upravo u tom segmentu smo napravili najveći iskorak definirajući **reprezentaciju područnog znanja** temeljenu na matematičkom modelu kojim smo osigurali ispravnost navedenog pristupa.

Model AC-ware Tutor-a je na najvišoj razini opisan sudionicima i funkcionalnostima sustava. Osim učenika, koji uči i testira svoje znanje, stručnjak je sudionik odgovoran za opisivanje područnog znanja u skladu s ontološkim formalizmom. Istaknut je redoslijed realizacije skupa funkcionalnosti sustava koji se razlaže na fazu oblikovanja područnog znanja, fazu postavljanja i fazu učenja i testiranja. Naravno, prva faza, u kojoj sudjeluje stručnjak, je oblikovanje područnog znanja. Ulogu učitelja preuzima sam sustav koji u fazi postavljanja

---

generira računalom oblikovani nastavni sadržaj na osnovi ontološki opisanog područnog znanja i modela učenika. Osim toga, u fazi postavljanja se inicijalizira model učenika. Konačno, AC-ware Tutor u fazi učenja i testiranja vodi proces učenja i testiranja znanja učenika.

Dinamika modela se sastoji od ponavljajućih ciklusa u kojima se izmjenjuju učenje, poučavanje i testiranje znanja. Ulaz u proces učenja i poučavanja, odnosno u njegov prvi ciklus, je početni stereotip određen na temelju ulaznog testa. Svaki ciklus završava određivanjem novog stereotipa učenika (ili eventualno zadržavanjem već pridijeljenog stereotipa iz prošlog ciklusa). U svakom ciklusu učenik se uči i poučava samo na jednom dijelu područnog znanja, koje mu je u tom ciklusu prezentirano na način koji je prilagođen onom stereotipu koji je rezultat prethodnog ciklusa. Tada se pristupa testiranju tog znanja koje dovodi do poželjnog mijenjanje stereotipa na više. Proces učenja i poučavanja završava kada učenik pokaže poznavanje svih koncepata barem na razini 1. Izlaz iz cijelog procesa učenja i poučavanja predstavlja određivanje učenikova završnog stereotipa. Ukoliko učenik želi završiti proces učenja i poučavanja kao bolji stereotip, onda ponovno ulazi u proces učenja i poučavanja s tim da ne treba pisati ulazni test, već mu je početni stereotip onaj koji je bio izlaz u prethodnom procesu učenja i poučavanja.

Realizaciju funkcionalnosti provode komponente i osnovni skupovi podataka. Komponente se razlažu na podsustave i module. Najvažniji podsustav je AC-ware ITS koji je odgovoran za realizaciju faze postavljanja i faze učenja i testiranja. Moduli podsustava ITS odgovaraju tradicionalnoj arhitekturi u kojoj se ističu modul stručnjaka, modul učitelja i modul učenika. Modul komunikacije u ovom modelu ne definiramo jer se ovakav pristup može realizirati u bilo kojem ITS-u koji ima znanje zasnovano nad ontologijom, neovisno o načinu komunikacije između učenika i sustava. Svaki od navedenih modula ITS-a se razlaže na dvije vrste komponenti. Prva vrsta komponenti se koristi u fazi postavljanja kako bi izvršilo postavljanje određenog skupa podataka u ITS-u, a druga vrsta komponenti služi tijekom realizacije faze učenja i testiranja. *Modul stručnjaka* sadrži komponentu odgovornu za preslikavanje ontološki opisanog područnog znanja u područno znanje po modelu AC-ware Tutor-a i sadrži mehanizam zaključivanja koji se koristi tijekom učenja i testiranja učenikova znanja. *Modul učitelja* u fazi postavljanja koristi komponentu za planiranje nastavnog sadržaja, a tijekom faze učenja i testiranja komponentu za vođenje učenja i testiranja. *Model učenika* inicijalno postavlja komponenta modula učenika, a dijagnostička komponenta istog modula se koristi za ažuriranje modela učenika tijekom faze učenja i testiranja.

Zamisao modela AC-ware Tutor-a je omogućila implementaciju prototipne verzije sustava. U prototipu se koristi područno znanje "Računalo kao sustav". Podsustavi AC-ware-ITS i aplikacija AC-ware-Web su realizirani Microsoft .NET tehnologijom, a relacijske baze podataka u Microsoft SQL poslužiteljskom okruženju. Korisničko sučelje za učenika je implementirano tehnologijom dinamičkih Web stranica. U prototipnoj verziji nije do kraja implementirano probabilističko zaključivanje.

---

Na prototipu AC-ware Tutor-a je provedeno akcijsko istraživanje na dvije skupine ispitanika. Osnovni cilj prototipnog testiranja je bio ustanovljavanje osjećaja "zadovoljstva" u korištenju AC-ware Tutor-a.

*Prvu* skupinu čine studenti prve godine preddiplomskog studija na Prirodoslovno matematičkom fakultetu Sveučilišta u Splitu. Studenti su se testirali na AC-ware Tutor-u nakon što su na predavanjima iz kolegija "Uvod u računarstvo" završili s učenjem cjeline „Računalo kao sustav“. Studenti su, nakon učenja i testiranja uz pomoć sustava AC-ware Tutor, popunili anketni upitnik u kojem se tražila ocjena "zadovoljstva". Interpretacijom rezultata anketnog upitnika ustanovljeno je da su studenti razumjeli nastavni sadržaj i smatraju kako bi primjena AC-ware Tutor-a utjecala na kvalitetu tradicionalne nastave.

*Drugo* prototipno testiranje provedeno je nad studentima prve godine diplomskog studija na Prirodoslovno matematičkom fakultetu Sveučilišta u Splitu, te na Fakultetu prirodoslovno-matematičkih i odgojnih znanosti Sveučilišta u Mostaru. Ovi studenti su upisani na kolegij „Sustavi e-učenja“, pa imaju odgovarajuće predznanje korisno za rad na prototipnom sustavu. Interpretacija rezultata anketnog upitnika pokazuje da su studenti razumjeli nastavni sadržaj i smatraju kako bi primjena AC-ware Tutor-a utjecala na kvalitetu tradicionalne nastave

Prototipna testiranja provedena nad dvjema skupinama studenata, osim pozitivnih ocjena, dala su zanimljive kritike na implementacijskoj razini prototipa sustava AC-ware Tutor što poticajno djeluje na nastavak istraživanja. U tom smislu, u sustav treba dodati grafičku komponentu koja bi velikom broju učenika olakšala usvajanje nastavnih sadržaja. Nadalje, pošto ontologija dopušta višejezično nazivanje koncepata i relacija, prevođenjem predložaka za generiranje rečenica i pitanja, bilo bi moguće napraviti višejezičnu verziju sustava AC-ware Tutor. Također, potrebno je u prototipnoj verziji sustava implementirati u potpunosti Bayesov model učenika i time omogućiti donošenje zaključaka o poznavanju koncepata prema odrednicama navedenim u opisu modela.

Temeljem svega iznesenog smatramo da su ovim istraživanjem postignuti sljedeći izvorni znanstveni doprinosi:

- *Oblikovan je model i implementiran prototip sustava za prilagodljivo stjecanje znanja u sustavu e-učenja.*

Model inteligentnog tutorskog sustava AC-ware Tutor-a zasniva svoja znanja na ontološki formaliziranom područnom znanju, a prilagođavanje je realizirano primjenom stereotipova i Bloomove taksonomije znanja. Implementirani prototip posjeduje automatsko generiranje elemenata računalom oblikovanog nastavnog sadržaja, dinamički odabir i nizanje elemenata računalom oblikovanog nastavnog sadržaja, automatsko generiranje pitanja i ulazni test nad reprezentativnim podskupom područnog znanja, u kojima se prilagođavanje realizira prema znanju učenika.

- 
- *Formaliziran je reprezentativni podskupa područnog znanja nad ontologijom za utvrđivanje početne razine znanja učenika*

Oblikovan je pristup za definiranje podskupa područnog znanja koji će dostojno reprezentirati cijelo područno znanje. Reprezentacija područnog znanja omogućava inicijalizaciju modela učenika, a definira se ne-sematičkim matematičkim pristupom primjenom teorije grafova.

- *Napravljena je specifikacija stereotipova učenika prema Bloomovoj taksonomiji znanja*

Stereotipovi učenika su povezani s aktualnom razinom znanja, a utemeljeni su na Bloomovoj taksonomiji znanja i strukturirani po skali znanja s nazivima: novak, početnik, osrednji, napredni i stručnjak.

- *Izvršeno je automatsko, dinamičko i prilagodljivo strukturiranje nastavnih sadržaja temeljem Bloomove taksonomije u suglasju s pedagoškom praksom u Hrvatskoj*

Odabir elemenata računalom oblikovanog nastavnog sadržaja se vrši prema razini elemenata. Razina elementa određuje količinu znanja koju će učenik učiti u jednom ciklusu učenja i poučavanja. S obzirom da smo definirali pet stereotipova učenika (novak, početnik, osrednji, napredni i stručnjak), a imamo samo tri razine elemenata (jedinica, tema, cjelina), neki od stereotipova će se učiti i poučavati na istoj razini. Nakon što se za određeni stereotip odaberu svi elementi za učenje određene razine, nizanje elemenata predstavlja dodavanje sortiranih elemenata u nastavni sadržaj i dodavanje elemenata za provjeravanje znanja prema pravilima koja ovise o stereotipu učenika. Prezentiranje elemenata se vrši s aspekta Bloomove taksonomije. Svakom stereotipu se prezentira odgovarajuća razina znanja. Način na koji će se znanje prezentirati učeniku u procesu učenja i poučavanja je definirano predlošcima za generiranje tvrdnji koje učenik mora usvojiti. Postoje četiri težinske kategorije tvrdnji za učenje i poučavanje. Svaka težinska kategorija prezentira određenu razinu znanja.

---

## 7 Literatura

1. Albert, D. (2001). *E-learning Future-The Contribution of Psychology*. Catching the Future: Women and Men in Global Psychology, Proceedings of the 59th Annual Convention, International Council of Psychologists, Winchester, England, pp. 30-53.
2. Anderson, J. R. (1988). *The expert module*. Foundations of intelligent tutoring systems, pp. 21-53.
3. Anderson, L. W., Krathwohl, D. R., & Bloom, B. S. (2004). *A taxonomy for learning, teaching, and assessing*. Longman New York.
4. Anh, N. V., Ha, N. V., & Dam, H. S. (2008). *Constructing a Bayesian Belief Network to Generate Learning Path in Adaptive Hypermedia System*. Journal of Computer Science and Cybernetics, 24(11).
5. ASTD. (2001). *A vision of E-learning for America's workforce: Report of the Commission on Technology and Adult Learning*. Commission On Technology And Adult Learning. <http://www.astd.org> (22.12.2011.)
6. Bayes, R. (1763). *An essay toward solving a problem in the doctrine of chances*. Philos. Trans. R. Soc. London, 53, pp. 370-418.
7. Beaumont, I. H. (1994). *User modelling in the interactive anatomy tutoring system ANATOM-TUTOR*. User Modelling and User-Adapted Interaction, 4(1), pp. 21-45. doi:10.1007/BF01142356
8. Beck, J., Stern, M., & Haugsjaa, E. (1996). *Applications of AI in Education*. Crossroads, 3(1), pp. 11-15.
9. Benyon, D., & Murray, D. (1993). *Adaptive systems: From intelligent tutoring to autonomous agents*. Knowledge-Based Systems, 6(4), pp. 197-219.
10. Beaumont, I., & Brusilousky, P. (1995). *Adaptive Educational Hypermedia*, pp. 93-98.
11. Bhaskar, M., Das, M. M., Chithralekha, T., & Sivasathya, S. (2010). *Genetic Algorithm Based Adaptive Learning Scheme Generation For Context Aware E-Learning*. International Journal on Computer Science and Engineering, Vol. 02, No. 04, pp. 1271-1279
12. Bloom, B. S. (1956). *Taxonomy of educational objectives*. The classification of educational goals, Handbook I Cognitive Domain. Green, New York, NY: Committee of College and University Examiners, Longmans.
13. Bloom, B. S. (1984). *The 2 Sigma Problem: The Search for Methods of Group Instruction as Effective as One-to-One Tutoring*. Educational Researcher, 13(6), pp. 4-16.



14. Bontchev, B., & Vassileva, D. (2009). *Adaptive courseware design based on learner character*. Proc. of International Conference on Interactive Computer Aided Learning (ICL2009), pp. 23-25.
15. Boyle, C., & Encarnacion, A. O. (1994). *MetaDoc: an adaptive hypertext reading system*. User Modeling and User-Adapted Interaction, 4(1), pp. 1-19.
16. Bruen, C., Conlan, O., & Wade, V. P. (2002). *Multi-model, metadata driven approach to adaptive hypermedia services for personalized eLearning*. Adaptive Hypermedia and Adaptive Web-Based Systems, Lecture Notes in Computer Science, Volume 2347/2006, pp. 100-111, DOI: 10.1007/3-540-47952-X\_12
17. Brusilovsky, P. (1998). *Adaptive educational systems on the world-wide-web: A review of available technologies*. Proceedings of Workshop WWW-Based Tutoring at 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX.
18. Brusilovsky, P., & Nijhawan, H. (2002). *A framework for adaptive e-learning based on distributed re-usable learning activities*. Proceedings of World Conference on E-Learning, E-Learn (Vol. 2002). Citeseer.
19. Brusilovsky, P., & Vassileva, J. (2003). *Course sequencing techniques for large-scale web-based education*. International Journal of Continuing Engineering Education and Life Long Learning, 13(1), pp. 75-94.
20. Brusilovsky, P., Schwarz, E., & Weber, G. (1996). *ELM-ART: An intelligent tutoring system on World Wide Web*. Intelligent Tutoring Systems, pp. 261-269. Springer.
21. Brusilovsky, Peter. (1996). *Methods and Techniques of Adaptive Hypermedia*. User Modeling and User Adapted Interaction, 6(2-3), pp. 87-129.
22. Burns, H. L., & Capps, C. G. (1988). *Foundations of intelligent tutoring systems: An introduction*. Foundations of intelligent tutoring systems (M. C. Poison, J. J. Richardson (Ed.)), pp. 1-19. Lawrence Erlbaum, London.
23. Carbonell, J. R. (1970). *AI in CAI: An artificial-intelligence approach to computer-assisted instruction*. Man-Machine Systems, IEEE Transactions on, 11(4), pp. 190-202.
24. Carchiolo, V., Longheu, A., Malgeri, M., & Mangioni, G. (2003). *Automatic generation of learning paths.*, Proceedings of the 10th IEEE International Conference on Electronics, Circuits and Systems ICECS (Vol. 3), pp. 1236-1239. IEEE.
25. Carr, B., & Goldstein, I. P. (1977). *Overlays. A theory of modeling for computer-aided instruction*, AI Lab Memo 406. Massachusetts Institute of Technology, Cambridge, Massachusetts.
26. Carrier, C. A., & Jonassen, D. H. (1988). *Adapting courseware to accommodate individual differences*. Instructional designs for microcomputer courseware, pp. 203-226.

27. Carro, R. M., Moriyón, R., Pulido, E., & Rodríguez, P. (1999). *Teaching tasks in an adaptive learning environment*. The 8th International Conference on Human-Computer Interaction, Munich, Germany. Citeseer.
28. Carro, R. M., Pulido, E., & Rodríguez, P. (1999). *TANGOW: Task-based Adaptive learner Guidance On the WWW*. Proceedings of the 8th International World Wide Web Conference. Presented at the Second Workshop on Adaptive Systems and User Modeling on the Web, Toronto, Canada.
29. Chang, M., Chang, A., Heh, J. S., & Liu, T. C. (2008). *Context-aware learning path planner*. WSEAS Transactions on Computers, 7(4), pp. 316-325.
30. Chapelle, C., & Mizuno, S. (1989). *Student's strategies with learner-controlled CALL*. Calico Journal, 7(2), pp. 25-47.
31. Charniak, E. (1991). *Bayesian Networks without tears*. AI magazine, 12(4), 50.
32. Charniak, Eugene, & McDermott, D. (1985). *Introduction to Artificial Intelligence*. Addison Wesley Publishing Company.
33. Cheng, J., & Greiner, R. (2001). *Learning bayesian belief network classifiers: Algorithms and system*. Advances in Artificial Intelligence, pp. 141-151.
34. Chi, M. T. ., Siler, S. A., Jeong, H., Yamauchi, T., & Hausmann, R. G. (2001). *Learning from human tutoring*. Cognitive Science, 25(4), pp. 471-533.
35. Chin, D. N. (1986). *User modeling in UC, the UNIX consultant*. ACM SIGCHI Bulletin - Special issue: CHI '86 Conference, Vol. 17, pp. 24-28. ACM.
36. Chin, D. N. (1989). *KNOME: Modeling what the user knows in UC*. User models in dialog systems, pp. 74-107.
37. Cohen, J. (1988). *Statistical power analysis for the behavioral sciences*. Lawrence Erlbaum Associates.
38. Cohen, L., Manion, L., & Morrison, K. (2007). *Metode istraživanja u obrazovanju, prijevod petog izdanja*. Naklada Slap.
39. Cohen, P. A., Kulik, J. A., & Kulik, C. L. . (1982). *Educational outcomes of tutoring: A meta-analysis of findings*. American educational research journal, 19(2), 237.
40. Conati, Cristina, Gertner, A. S., Vanlehn, K., & Druzdzel, M. J. (1997). *On-line student modeling for coached problem solving using Bayesian networks*. User Modeling: Proceedings of the Sixth International Conference, UM97, pp. 231--242.
41. Conati, C., Gertner, A., & Vanlehn, K. (2002). *Using Bayesian networks to manage uncertainty in student modeling*. User Modeling and User-Adapted Interaction, 12(4), pp. 371-417.

42. Conejo, R., Guzmán, E., Millán, E., Trella, M., Pérez-De-La-Cruz, J. L., & Ríos, A. (2004). *Siette: a web-based tool for adaptive testing*. International Journal of Artificial Intelligence in Education, 14(1), pp. 29-61.
43. Corno, L., & Snow, R. E. (1986). *Adapting teaching to individual differences among learners*. Handbook of research on teaching, 3, pp. 605-629.
44. Cronbach, L., & Snow, R. (1977). *Aptitudes and Instructional Methods: A Handbook for Research on Interactions*. New York, NY, USA: Irvington.
45. Dagger, D., Wade, V., & Conlan, O. (2004). *Developing Adaptive Pedagogy with the Adaptive Course Construction Toolkit (ACCT)*. Second International Workshop on Authoring of Adaptive and Adaptable Educational Hypermedia, AH (Vol. 4).
46. Dagger, D., Wade, V., & Conlan, O. (2005). *Personalisation for all: Making adaptive course composition easy*. Educational Technology & Society, 8(3), pp. 9-25.
47. de Rosis, F., De Carolis, B., & Pizzutilo, S. (1993). *User tailored hypermedia explanations*. INTERCHI, Vol. 93, pp. 169-170.
48. Dick, W., & Carey, L. (1985). *The systematic design of instruction* Glenview. IL: Scott, Foreman and Co mpany.
49. Diessel, T., Lehmann, A., & Vassileva, J. (1994). *Individualised Course Generation: a Marriage between CAL and ICAL*. Computers & Education, 22(1/2), pp. 57-64.
50. Dijkstra, S., Krammer, H. P., & Van Merriënboer, J. G. (1992). *Instructional models in computer-based learning environments*. Springer-Verlag New York, Inc. Secaucus, NJ, USA.
51. eLAN. (2004). *E-learning and Assessment*, Version 4. eLearning Adviser Network, University of Bristol. [www.bris.ac.uk/elan/documentation/relateddocuments/caa\\_v.4.doc](http://www.bris.ac.uk/elan/documentation/relateddocuments/caa_v.4.doc) (22.12.2011.)
52. Essalmi, F., Ayed, L. J. B., Jemni, M., Kinshuk, & Graf, S. (2010). *A fully personalization strategy of E-learning scenarios*. Computers in Human Behavior, 26(4), pp. 581-591. doi:16/j.chb.2009.12.010
53. Evjen, B., Gibbs, M., Wahlin, D., & Reed, D. (2011). *Professional ASP. NET 3.5 AJAX*. Wrox.
54. Felder, R. M., & Silverman, L. K. (1988). *Learning and teaching styles in engineering education*. Engineering education, 78(7), pp. 674-681.
55. Gagne, R. M., & Briggs, L. J. (1979). *Principles of instructional design*. Holt Rinehart and Winston.
56. Gamboa, H., & Fred, A. (2002). *Designing intelligent tutoring systems: a bayesian approach*. Enterprise information systems III, 1, pp. 452-458.

- 
57. García-Barrios, V. M., Mödritscher, F., & Gütl, C. (2005). *Personalisation versus Adaptation? A User-centred Model Approach and its Application*. Proceedings of the International Conference on Knowledge Management (I-KNOW), pp. 120-127. Citeseer.
  58. Gascuena, J. M., Fernandez-Caballero, A., & Gonzalez, P. (2006). *Domain ontology for personalized e-learning in educational systems*. Sixth International Conference Advanced Learning Technologies, pp. 456-458. IEEE.
  59. Gavignet, E. (1991). *Environnement de conception de système d'apprentissage: une modélisation de la connaissance pédagogique*, doktorska disertacija, de l'Université de Nancy 1.
  60. Genesereth, M. R., & Nilsson, N. J. (1987). *Logical Foundations of Artificial Intelligence* (First Edition.). Morgan Kaufmann.
  61. Gertner, A. C., Conati, C., & VanLehn, K. (1998). *Procedural help in Andes: Generating hints using a Bayesian network student model*. Proceedings of the 15th National Conference on Artificial Intelligence. Cambridge, MA: The MIT Press, pp. 106-111.
  62. Gilbert, J. E. (2000). *Arthur: An intelligent tutoring system with adaptive instruction*. doktorska disertacija, University of Cincinnati.
  63. Glaser, R., & Nitko, A. J. (1970). *Measurement in Learning and Instruction*.
  64. Grandbastien, M. (1999). *Teaching expertise is at the core of ITS research*. International journal of artificial intelligence in education, 10(3-4), pp. 335-349.
  65. Grandbastien, M., & Gavignet, E. (1994). *ECSA: An Environment to Design and Instantiate Learning Material*, Handbook of Design and Production of Multimedia and Simulation-based Learning Material (de Jong T. and L. Sarti, eds.). Kluwer Academic, Publishers, Dordrecht, The Netherlands.
  66. Grigoriadou, M., Papanikolaou, K., Kornilakis, H., & Magoulas, G. (2001). *INSPIRE: an intelligent system for personalized instruction in a remote environment*. Proceedings of Third workshop on Adaptive Hypertext and Hypermedia, Vol. 14, pp. 13-24. Citeseer.
  67. Gronlund, N. E. (1985). *Measurement and Evaluation in Teaching*, 5th ed. New York: Macmillan Publishing Company.
  68. Gross, J. L., & Yellen, J. (1998). *Graph Theory and Its Applications*, 1st ed.. CRC Press.
  69. Gruber, T. R. (1993). *A translation approach to portable ontology specifications*. Knowledge acquisition, 5(2), pp. 199-220.
  70. Grubišić, A. (2007). *Vrednovanje učinka inteligentnih sustava e-učenja*, magistarski rad. Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva.
  71. Grubišić, A. (2010). *A Meta-Analytic Estimation of a Common Effect Size from a Series of Experiments Related to an E-Learning System Effectiveness Evaluation*. Intelligent

- 
- Tutoring Systems in E-Learning Environments: Design, Implementation and Evaluation, 327.
72. HEC-PIEAS. (2009). *Handout on Objective type Questions. Workshop on Testing and Assesment*, 2009, Higher Education Commission at Pakistan Institute of Engineering & Applied Sciences (PIEAS). <http://www.pieas.edu.pk/umarfaiz/workshop/objtypebloom.doc>, (22.12.2011.)
  73. Henze, N., & Nejd, W. (1999). *Adaptivity in the KBS Hyperbook System*. 2nd Workshop on User Modeling and Adaptive Systems on the WWW.
  74. Hohl, H., Böcker, H. D., & Gunzenhäuser, R. (1996). *Hypadapter: An adaptive hypertext system for exploratory learning and programming*. *User Modeling and User-Adapted Interaction*, 6(2), pp. 131-156.
  75. Hoic-Bozic, N., & Mornar, V. (2005). *AHyCo: A Web-based adaptive hypermedia courseware system*. *Journal of Computing and Information Technology*, 13(3), 165.
  76. Honey, P., & Mumford, A. (1992). *The manual of learning styles*. Maidenhead: Peter Honey
  77. Karampiperis, P., & Sampson, D. (2004). *Adaptive Learning Object Selection in Intelligent Learning Systems*. *Journal of Interactive Learning Research*, 15(4), pp. 389-408.
  78. Karampiperis, P., & Sampson, D. (2005). *Adaptive learning resources sequencing in educational hypermedia systems*. *Educational Technology & Society*, 8(4), pp. 128-147.
  79. Kemmis, S., McTaggart, R., & Program, D. U. S. of E. O. C. (1988). *The action research planner*. Deakin University: distributed by Deakin University Press.
  80. Khan, B. H. (2001). *A framework for Web-based learning*. *Web-based training*, pp. 75-98.
  81. Kinshuk, Lin, T., & Patel, A. (2005). *Supporting the Mobility and the Adaptivity of Knowledge Objects by Cognitive Trait Model*. *Innovations in Instructional Technology*, 29.
  82. Krishnamurthy, V., Faloutsos, M., Chrobak, M., Lao, L., Cui, J. H., & Percus, A. G. (2005). *Reducing large internet topologies for faster simulations*. *NETWORKING 2005*, pp. 328-341.
  83. Kyllonen, P. C., & Shute, V. J. (1988). *Learning indicators from a taxonomy of learning skills*. *Learning and individual differences*. New York: Freeman.
  84. Lee, T. B., Hendler, J., & Lassila, O. (2001). *The semantic web*. *Scientific American*, 284(5), pp. 34-43.
  85. Lepper, M. R., Aspinwall, L., Mumme, D., & Chabay, R. W. (1990). *Self-perception and social-perception processes in tutoring: Subtle social control strategies of expert tutors*. *Self-inference processes: The Ontario symposium (Vol. 6)*, pp. 217-237.
  86. Leskovec, J., & Faloutsos, C. (2006). *Sampling from large graphs*, pp. 631-636. ACM.
-

- 
87. Lin, T. (2007). *Cognitive Trait Model for Adaptive Learning Environments*. doktorska disertacija, Massey University, Palmerston North, New Zealand.
  88. LTSC. (2011). *Learning technology standards committee website*. <http://ltsc.ieee.org/wg12/20020612-Final-LOM-Draft.html> (22.12.2011.)
  89. Macías, J., & Castells, P. (2001). *An authoring tool for building adaptive learning guidance systems on the web*. Active Media Technology, pp. 268-278.
  90. Macías, J., & Castells, P. (2002). *Interactive Design of Adaptive Courses*. Computers and Education, pp. 235-242.
  91. Marguerie, F., Eichert, S., & Wooley, J. (2008). *LINQ in Action*. Manning Publications Co.
  92. Márquez, J. M., Ortega, J. A., Gonzalez-Abril, L., & Velasco, F. (2008). *Creating adaptive learning paths using Ant Colony Optimization and Bayesian Networks*. Neural Networks, IEEE World Congress on Computational Intelligence, pp. 3834-3839. IEEE.
  93. Mayo, M. J. (2001). *Bayesian Student Modelling and Decision-theoretic Selection of Tutorial Actions in Intelligent Tutoring Systems*. A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of Doctor of Philosophy in Computer Science in the University of Canterbury. doktorska disertacija, University of Canterbury, Christchurch, New Zealand.
  94. Mayo, M., & Mitrovic, A. (2001). *Optimising ITS behaviour with Bayesian networks and decision theory*. International Journal of Artificial Intelligence in Education, 12(2), pp. 124-153.
  95. McNiff, J., & Whitehead, J. (2002). *Action research: Principles and practice*. Routledge.
  96. Melis, E., Gogvadze, G., Libbrecht, P., & Ullrich, C. (2009). *Activemath-a learning platform with semantic web features*. The Future of Learning, 159.
  97. Mendez, N. D. ., Ramirez, C. J., & Luna, J. A. (2005). *IA planning for automatic generation of customized virtual courses*. Planning, scheduling and constraint satisfaction: from theory to practice, 117, 139.
  98. Merrill, D. C., Reiser, B. J., Ranney, M., & Trafton, J. G. (1992). *Effective tutoring techniques: A comparison of human tutors and intelligent tutoring systems*. Journal of the Learning Sciences, 2(3), pp. 277-305.
  99. Merrill, M. D. (1998). *Knowledge objects*. CBT solutions, 2, pp. 1-11.
  100. Mislevy, R. J., & Gitomer, D. H. (1996). *The Role of Probability-Based Inference in an Intelligent Tutoring System*. User Modeling and User-Adapted Interaction: The Journal of Personalization Research, 5(3-4), pp. 253-282.
  101. Mödritscher, Felix. (2008). *Adaptive E-Learning Environments: Theory, Practice, and Experience*. Verlag Dr. Müller.
-

- 
102. Mohan, P., Greer, J., & McCalla, G. (2003). *Instructional planning with learning objects*. Knowledge Representation and Automated Reasoning for E-Learning Systems, pp. 52-58.
103. Murray, R., & VanLehn, K. (2000). *DT Tutor: A Decision-Theoretic Dynamic Approach for Optimal Selection of Tutorial Actions*. Intelligent Tutoring Systems, pp. 153-162. Springer.
104. Notar, C. E., Zuelke, D. C., Wilson, J. D., & Yunker, B. D. (2004). *The Table of Specifications: Insuring Accountability in Teacher Made Tests*. Journal of Instructional Psychology, 31(2), pp. 115-130.
105. O Keeffe, I., Brady, A., Conlan, O., & Wade, V. (2006). *Just-in-time generation of pedagogically sound, context sensitive personalized learning experiences*. International Journal on e-Learning, 5(1), 113.
106. Ohlsson, S. (1987). *Some principles of intelligent tutoring*. Artificial Intelligence and Education (Lawler, R.W., Yazdani, M.), Vol. 1, pp. 203–237. Ablex: Norwood, NJ.
107. Ohlsson, S. (1994). *Constraint-based student modeling*. NATO ASI SERIES F COMPUTER AND SYSTEMS SCIENCES, 125, pp. 167-167.
108. Oppermann, R., & Rasher, R. (1997). *Adaptability and adaptivity in learning systems*. Knowledge Transfer, 2, pp. 173-179.
109. Papanikolaou, K. A., Grigoriadou, M., Kornilakis, H., & Magoulas, G. D. (2003). *Personalizing the Interaction in a Web-based Educational Hypermedia System: the case of INSPIRE*. User Modeling and User-Adapted Interaction, 13(3), pp. 213-267.
110. Paramythis, A., & Loidl-Reisinger, S. (2003). *Adaptive learning environments and e-learning standards*. Proceedings of the 2nd European Conference on e-Learning, Glasgow.
111. Park, O., & Lee, J. (1996). *Adaptive instructional systems*. Handbook of research for educational communications and technology, pp. 634-664.
112. Pearl, J. (1985). *Bayesian Networks: A Model of Self-Activated: Memory for Evidential Reasoning*. Computer Science Department, University of California.
113. Pukkhem, N., Evens, M. W., & Vatanawood, W. (2006). *The Concept Path Combination Model for Supporting a Personalized Learning Path in Adaptive Educational Systems*. Proceedings of the 2006 International Conference on e-Learning, e-Business, Enterprise Information Systems, e-Government, and Outsourcing (EEE'06). Citeseer.
114. Rattz, J. C. (2007). *Pro LINQ: Language Integrated Query in C#*, Springer.
115. Reiser, R. A. (1987). *Instructional technology: A history*. Instructional technology: foundations, pp. 11-48.
-

- 
116. Reye, J. (1998). *Two-phase updating of student models based on dynamic belief networks*, pp. 274-283. Springer.
117. Riad, A. M., El-Minir, H. K., & El-Ghareeb, H. A. (2009). *Review of e-Learning Systems Convergence from Traditional Systems to Services based Adaptive and Intelligent Systems*. *Journal of Convergence Information Technology*, 4(2).
118. Rich, E. (1979). *User modeling via stereotypes*. *Cognitive Science: A Multidisciplinary Journal*, 3(4), pp. 329-354.
119. Rich, E. (1983). *Users are individuals: individualizing user models*. *International journal of man-machine studies*, 18(3), pp. 199-214.
120. Rickel, J. W. (1989). *Intelligent computer-aided instruction: A survey organized around system components*. *Systems, Man and Cybernetics, IEEE Transactions on Systems, Man, and Cybernetics*, 19(1), pp. 40-57.
121. Roberson-Scott, K. (2005). *Evaluation of Student Learning: Test Construction & Other Practical Strategies*. Office of Institutional Effectiveness & Research, Roane State Community College. Retrieved from <http://www.roanestate.edu/effectiveness/resources/TestConstructionAndEvaluation.ppt> (22.12.2012.)
122. Sanrach, C., & Grandbastien, M. (2000). *ECSAIWeb: A Web-Based Authoring System to Create Adaptive Learning Systems*. In Peter Brusilovsky, O. Stock, & C. Strapparava (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems*, Vol. 1892, pp. 214-226. Berlin, Heidelberg: Springer Berlin Heidelberg.
123. Santos, J., Anido, L., Llamas, M., & Rodríguez, J. (2002). *On the Application of the semantic Web Concepts to Adaptive E-learning*. *EurAsia-ICT 2002: Information and Communication Technology*, pp. 536-543.
124. Self, J. A. (1974). *Student models in computer-aided instruction*. *International Journal of Man-Machine Studies*, 6(2), pp. 261-276.
125. Shortliffe, E. H., Davis, R., Axline, S. G., Buchanan, B. G., Green, C. C., & Cohen, S. N. (1975). *Computer-based consultations in clinical therapeutics: explanation and rule acquisition capabilities of the MYCIN system*. *Computers and Biomedical Research*, 8(4), pp. 303-320.
126. Shute, V. J., & Psotka, J. (1996). *Intelligent Tutoring Systems: Past, Present and Future*. *Handbook of Research on Educational Communications and Technology* (Jonassen, D.). New York, NY: Macmillan.
127. Shute, V. J., & Zapata-Rivera, D. (2008). *Adaptive technologies*. *Handbook of research on educational communications and technology*, pp. 277-294.
128. Shute, V. J., Lajoie, S. P., & Gluck, K. A. (2000). *Individualized and group approaches to training*. *Training and retraining: A handbook for business, industry, government, and the military*, pp. 171-207.
-



- 
129. Shute, V., & Towle, B. (2003). *Adaptive e-learning*. *Educational Psychologist*, 38(2), pp. 105-114.
130. Sleeman, D., & Brown, J. S. (1982). *Introduction: Intelligent Tutoring Systems: An Overview*. *Intelligent Tutoring Systems* (Sleeman, D.H., Brown, J.S.), pp. 1-11. Academic Press, Burlington, MA.
131. Snow, R. E., & Swanson, J. (1992). *Instructional psychology: Aptitude, adaptation, and assessment*. *Annual Review of Psychology*, 43(1), pp. 583-626.
132. Specht, M., & Oppermann, R. (1998). *ACE-adaptive courseware environment*. *The New Review of Hypermedia and Multimedia*, 4(1), pp. 141-161.
133. Specht, M., Kravcik, M., Pesin, L., & Klemke, R. (2001). *Authoring adaptive educational hypermedia in WINDS*. *Proceedings of ABIS2001*, Dortmund, Germany, pp. 1-8.
134. Stankov, S. (1997). *Izomorfni model sustava kao osnova računalom poduprtog poučavanja načela vođenja*. doktorska disertacija, Fakultet elektrotehnike, strojarstva i brodogradnje, Sveučilište u Splitu, Split.
135. Stankov, S. (2003). *Web orijentirana inteligentna hipermedijska autorska ljuska*, TP-02/0177-01 Tehnologijski projekt MZT.
136. Stankov, S., Glavinić, V., & Grubišić, A. (2004). *What is our effect size: Evaluating the educational influence of a web-based intelligent authoring shell*, In: Nedeveschi, S., Rudas, I. J. (eds.): 8th International Conference on Intelligent Engineering Systems. Cluj-Napoca pp. 545-550.
137. Stankov, S., Grubišić, A., & Žitko, B. (2004). *E-learning paradigm & Intelligent tutoring systems*. *Annual 2004 of the Croatian Academy of Engineering*, pp. 21-31.
138. Stankov, S., Rosic, M., Zitko, B., & Grubisic, A. (2008). *TEx-Sys model for building intelligent tutoring systems*. *Computers & Education*, 51(3), pp. 1017-1036.
139. Touretzky, D. S. (1992). *Inheritance Hierarchy*. In I. Shapirio & C. Stuart (Eds.), *Artificial Intelligence – Encyclopedias*, pp. 690-701. John Willy & Sons, Inc.
140. Uhr, L. (1969). *Teaching machine programs that generate problems as a function of interaction with students*. *Proceedings of the 24th national conference*, pp. 125-134. ACM.
141. Ullrich, C. (2008). *Pedagogically founded courseware generation for web-based learning: an HTN-planning-based approach implemented in PAIGOS*. *Lecture Notes In Computer Science*; Vol. 5260, 232.
142. Urban-Lurain, M. (1996). *Intelligent tutoring systems: An historic review in the context of the development of artificial intelligence and educational psychology*. In: Technical Report, Department of Computer Science and Engineering, Michigan State University.
-

- 
143. Van Marcke, K. (1990). *A generic tutoring environment*. The European Conference on Artificial Intelligence, pp. 655-660.
144. Van Marcke, K. (1992). *A generic task model for instruction*. Instructional models for computer-based learning environments, Nato ASI Series F, Vol. 104, Berlin: Springer-Verlag.
145. Van Marcke, K. (1998). *GTE: An epistemological approach to instructional modelling*. Instructional Science, 26(3), pp. 147-191.
146. VanLehn, K., Niu, Z., Siler, S., & Gertner, A. (1998). *Student modeling from conventional test data: A Bayesian approach without priors*, In Goettle B., Halff H., Redfield C., and Shute V. (Eds.) Proc. of the 5th International Conference on Intelligent Tutoring Systems, Springer-Verlag, pp. 434-443.
147. Vassileva, J. (1992). *Dynamic CAL-courseware generation within an ITS-shell architecture*. Computer Assisted Learning, pp. 581-591.
148. Vassileva, J. (1998). *DCG+ GTE: Dynamic courseware generation with teaching expertise*. Instructional Science, 26(3), pp. 317-332.
149. Vassileva, J., & Wasson, B. (1996). *Instructional planning approaches: From tutoring towards free learning*. Proc. EuroAIED, 96, pp. 1-8.
150. Veljan, D. (1989). *Kombinatorika s teorijom grafova*. Zagreb: Školska knjiga.
151. Weber, G. (1996). *Episodic Learner Modeling*. Cognitive Science, 20(2), pp. 195-236. doi:16/S0364-0213(99)80006-8
152. Weber, G., & Moellenberg, A. (1995). *ELM-Programming-Environment: A Tutoring System for LISP Beginners*. Cognition and Computer Programming (K. F. Wender, F. Schmalhofer and H.-D. Böcker), pp. 373-408. Norwood, NJ: Ablex.
153. Weber, G., Kuhl, H. C., & Weibelzahl, S. (2002). *Developing adaptive internet based courses with the authoring system NetCoach*. Hypermedia: Openness, Structural Awareness, and Adaptivity, pp. 222-223.
154. Wenger, E. (1987). *Artificial Intelligence and Tutoring Systems*. Morgan Kaufmann Publishers, Inc., California, USA.
155. Wikipedia. (2011). *Stereotype*. <http://en.wikipedia.org/wiki/Stereotype> (22.12.2012.)
156. Wilensky, R., Arens, Y., & Chin, D. (1984). *Talking to UNIX in English: an overview of UC*. Communications of the ACM, 27(6), pp. 574-593.
157. Wilensky, R., Chin, D. N., Luria, M., Martin, J., Mayfield, J., & Wu, D. (1988). *The Berkeley UNIX consultant project*. Computational Linguistics, 14(4), pp. 35-84.
158. Wiley, D. A. (2000). *Connecting learning objects to instructional design theory: A definition, a metaphor, and a taxonomy*. The instructional use of learning objects, 2830(435), pp. 1-35.
-

159. Zhu, F., & Cao, J. (2008). *Learning activity sequencing in personalized education system*. Wuhan University Journal of Natural Sciences, 13(4), pp. 461-465.
160. Žitko, B. (2010). *Model inteligentnog tutorskog sustava zasnovan na obradi kontroliranog jezika nad ontologijom*. Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva.

---

## 8 Prilozi

S obzirom na kompleksnost područja obuhvaćenog ovom doktorskom disertacijom, veliko dio detaljnih objašnjenja, opisa i analiza se nalazi u prilogima. Prilozi ovoj doktorskoj disertaciji obuhvaćaju sljedeće:

- prikaz i detaljan opis sustava koji su analizirani za potrebe utvrđivanja stanja istraženosti iz područja generiranja courseware-a (Prilog 8.1)
- ontologija "Računalo kao sustav" prikazana u sustavu Cmap Tools COE (Prilog 8.2)
- analiza istraživanja o određivanju reprezentacije temeljem semantičke analize koncepata i relacija u područnom znanju (Prilog 8.3)
- oblikovanje i strukturiranje elemenata računalom oblikovanog nastavnog sadržaja (Prilog 8.4)
- inicijalizacija modela učenika (Prilog 8.5)
- detaljni dijagrami klasa implementiranog prototipa AC-ware Tutor (Prilog 8.6)
- anketni upitnik korišten u okviru prototipnog testiranja sustava AC-ware Tutor (Prilog 8.7)

### ***8.1 Prikaz sustava za generiranje računalom oblikovanog nastavnog sadržaja***

Svaki je sustav opisan na način da je ukratko navedena svrha sustava i njegovi autori. Zatim je navedeno da li sustav ima i kakvo je područno znanje, model učenika, računalom oblikovani nastavni sadržaj, provjeravanje znanja i prilagodljivost. Također je eksplicitno navedena značajka učenika prema kojoj se sustav prilagođava, te koji je element prilagođavanja.

#### **8.1.1 GTE 1990**

Generic Tutoring Environment (GTE) je okruženje za razvoj inteligentnog računalom oblikovanog nastavnog sadržaja temeljenog na generičkom znanju koje uključuje zadatke, metode i objekte ((Van Marcke, 1990), (Van Marcke, 1992), (Van Marcke, 1998)).

**Područno znanje** - Nema ga.

**Model učenika** - Model učenika u GTE-u se sastoji od tri dijela. Prvi dio je standardni model prekrivanja (Carr & Goldstein, 1977) koji sadrži procjene koliko dobro učenik razumije područno znanje. Učeničko znanje je prikazano u terminima tema klasificiranih sa „nije

naučena“, „prilično dobro naučena“ i „dobro naučena“. Drugi dio prati koji objekti su već dani učeniku i vode odabir ostalih objekata. Treći dio sadrži učenikove atribute: motivacija, stil učenja, kontrolu i predznanje. Učenikovi atributi se ne mijenjaju automatski, već ih određuje učenik ili učitelj.

**Računalom oblikovani nastavni sadržaj** - Obrazovni zadaci i metode kontroliraju interakcije sa učenikom u bilo kojem područnom znanju. Zadatak je skup aktivnosti koje se trebaju ispuniti tijekom procesa poučavanja, npr. „daj primjer“, „daj protuprimjer“, „daj vježbu“, „daj pregled“, itd. Zadaci su generički jer se isti mogu javiti u različitim područnim znanjima i različitim učenicima. Dakle, kontekst može biti različit, ali se isti zadatak javlja zbog sličnih razloga i koristi slične mehanizme.

Metode izvršavaju zadatke dijeleći ih u podzadatke sve do primitiva. Dok zadaci opisuju što treba napraviti, metode opisuju kako to treba napraviti. Na primjer, zadatak „pojasni koncept“ se može ostvariti metodom „objasni pomoću analogije“. Ljudski učitelji imaju na raspolaganju nekoliko metoda za ostvarenje jednog zadatka. GTE-ova fleksibilnost se očituje u mogućnosti dinamičkog odabira najprikladnije metode za svaki zadatak. Metode su također generičke, tj. neovisne o područnom znanju. Konačni rezultat dekompozicije zadataka je stablo nazvano struktura zadataka.

**Prilagodljivost** - Dinamički odabir metoda je ključan za GTE-ovo fleksibilno prilagođavanje. Kada se određeni zadatak treba izvršiti, GTE u trenutku izvođenja odabire najprikladniju od metoda koje su pridijeljene tom zadatku. Odabir vrši na temelju primjenjivosti (eng. *applicability value*) koja se izračunava korištenjem uvjeta primjenjivosti (eng. *applicability conditions*) koji opisuju koliko je određena metoda primjenjiva u određenom kontekstu.

GTE omogućava nizanje sadržaja za učenje, tj. on je fokusiran na problem „Kako poučavati?“, a ne na „Što poučavati?“. Iako GTE ima model sadržaja (eng. *content model*) koje se sastoji od tema, on je ipak statični kostur oko kojeg se gradi prezentacija. Umjesto dinamičkog odlučivanja koju temu prezentirati, generira se cijela putanja tema. Sadržaji za učenje u DCG-u su klasificirani prema pedagoškoj funkciji i prema mediju kojeg koriste (tekst, slika, animacija, video, itd.) i odgovaraju „obrazovnim primitivima“ (eng. *instructional primitives*) (Van Marcke, 1992).

**Značajka učenika prema kojoj se prilagođava** - Nije navedeno.

**Element prilagođavanja** - Tema.

**Provjeravanje znanja** - Nema ga.

### 8.1.2 ECSAI 1991

ECSAI (franc. *environnement de conception de Système d'Apprentissage*) je generator za okruženja za učenje, kojemu je u središtu inteligentno nizanje nastavnih jedinica ((Gavignet, 1991), (Grandbastien & Gavignet, 1994), (Grandbastien, 1999)).

---

**Područno znanje** - Jedinice organizirane u hijerarhijsku mrežu.

**Model učenika** - Model učenika je jednostavni model prekrivanja.

**Računalom oblikovani nastavni sadržaj** - Oblikovanje u sustavu ECSAI izvodi se u nekoliko koraka. Autor stvara ili importira skup objekata učenja iz raspoloživih dokumenata. Zatim svakoj nastavnoj jedinici u modelu pridružuje opis koji uključuje: oznaku, vrstu (prezentacija, primjer, vježba, itd.), tekstualni opis, preuvjete (završene nastavne jedinice, znanje učenika o određenim elementima područnog znanja) i postuvjete (promjene u modelu učenika nakon završavanja nastavne jedinice). Dakle, generiranje elemenata računalom oblikovanog nastavnog sadržaja nije automatizirano, već sve mora raditi učitelj.

**Prilagodljivost** - Nakon što učenik navede svoje ciljeve učenja (iz izbornika bira elemente područnog znanja koje želi naučiti i razinu na kojoj ih želi naučiti), ECSAI određuje prvu nastavnu jedinicu koju će prezentirati, prikazuje je učeniku, promijeni model učenika prema učenikovom ponašanju tijekom ove interakcije, zatim prezentira sljedeću jedinicu. Postupak se nastavlja dok se ne ostvare ciljevi ili dok ne prezentira sve nastavne jedinice.

ECSAIWeb (Sanrach & Grandbastien, 2000) je web verzija sustava ESCAI, koja omogućava oblikovanje inteligentnih tutorskih sustava kod kojih je prilagodljivost realizirana prilagodljivom prezentacijom i prilagodljivom podrškom navigaciji.

**Značajka učenika prema kojoj se prilagođava** - Znanje i ponašanje učenika.

**Element prilagođavanja** - Jedinica.

**Provjeravanje znanja** - Predznanje se provjerava upitnikom. Ostalih testova nema.

### 8.1.3 DCG 1995

Dynamic Courseware Generator (DCG) je kompromis između ITS-a i tradicionalnog CAI pristupa ((Vassileva, 1992), (Vassileva, 1995)). DCG razlikuje koncepte područnog znanja predstavljenih grafom (planiranje sadržaja) i obrazovne resurse (planiranje prezentacije).

**Područno znanje** - Jezgra DCG-a je eksplicitna reprezentacija strukture koncepata područnog znanja koja je odvojena od sadržaja za učenje i pedagoških zadataka ((Vassileva, 1992), (Diessel, Lehmann, & Vassileva, 1994)).

Struktura područnog znanja sadrži koncepte područnog znanja koje treba usvojiti. Prikazana je I/ILI grafom. Čvorovi predstavljaju elemente znanja (koncepti, teme, pravila, itd.). Ako su dva čvora A i B povezana s trećim čvorom C bridom I, to znači da se oba čvora A i B moraju usvojiti prije čvora C. Ako su povezana bridom ILI, onda čvorovi A i B predstavljaju alternative od kojih biramo samo jednu. Bridovi u grafu predstavljaju veze između koncepata različitog značenja (agregacija, generalizacija,...). Moguće je organizirati koncepte područnog znanja u skupove manjih I/ILI grafova koji predstavljaju relativno nezavisne dijelove područnog znanja, različite „pogled“ ili različite razine granulacije. Najjednostavnija struktura

---

područnog znanja se može postići korištenjem samo jedne vrste veze – preduvjet od (eng. prerequisite). Na ovaj način se postiže struktura koja omogućava nizanje sadržaja (Brusilovsky & Vassileva, 2003).

Svaki čvor i svaki brid je povezan sa skupom sadržaja za učenje (eng. teaching materials - TM) koji omogućava različite načine poučavanja koncepta (npr. upoznaj, objasni, daj primjer, vježbu ili test). Struktura područnog znanja se koristi za stvaranje plana računalom oblikovanog nastavnog sadržaja (podgraf područnog znanja) da bi se ostvario cilj (poznavanje koncepta). Ovaj plan se naziva „Plan sadržaja“, a proces se naziva „Planiranje sadržaja“. Tijekom izvršavanja računalom oblikovanog nastavnog sadržaja, odabiru se različiti sadržaji za učenje.

**Model učenika** - Model učenika u DCG-u je numeričko prekrivanje sa strukturom konceptata, tj. učenikovo poznavanje svakog koncepta je prikazano brojem unutar određenog intervala. Ako učenik ne može postići graničnu vrijednost za određeni koncept, generira mu se novi plan koji izbjegava teški koncept.

**Računalom oblikovani nastavni sadržaj** - Obrazovni resursi su HTML stranice vezane na koncepte područnog znanja. Računalom oblikovani nastavni sadržaj kojeg generira DCG izgleda kao tradicionalno strukturiran računalom oblikovani nastavni sadržaj. Međutim, ovaj računalom oblikovani nastavni sadržaj je generiran za svakog učenika posebno da bi se ostvario određeni cilj učenja. Generiranje uzima u obzir učenikovo znanje, te uvažava različitosti u načinu i ritmu stjecanja znanja.

Kada su poznati model učenika i ciljni koncepti, planer sadržaja generira putanje kroz graf konceptata područnog znanja, koje povezuju koncepte poznate učeniku s ciljevima učenja. Taj plan predstavlja nacrt za planiranje prezentacije. Planer prezentacije bira obrazovne resurse povezane s konceptima područnog znanja i definira redoslijed u kojem će oni biti prezentirani učeniku (Vassileva, 1992).

DCG koristi strukturu konceptata (predstavljenu skupom pravila) kao osnovicu za generiranje plana računalom oblikovanog nastavnog sadržaja. Na temelju ciljnog koncepta kojeg učenik želi naučiti i modela učenika koji sadrži koncepte koje učenik poznaje (model učenika se inicijalizira na temelju rezultata ulaznog testa), planer traži putanju koja povezuje koncepte koje učenik poznaje s ciljnim konceptom. Učenik vidi niz sadržaja za učenje koji se odnose na koncepte iz plana. Sadržaji za učenje sadrže prezentaciju i elemente za provjeravanje znanja kojima se ostvaruje interakcija sa studentom. Sadržaji za učenje su klasificirani prema pedagoškoj funkciji i prema mediju kojeg koriste (tekst, slika, animacija, video, itd.).

Sam planer sadržaja u DGC-u ne može odrediti kako će prezentirati odabrane sadržaje učeniku, odnosno koji pedagoški tip sadržaja za učenje odabrati ili kako nizati nekoliko sadržaja za učenje da bi poučavao učenika o nekom konceptu. U tu svrhu se koristi „Plan prezentiranja“ za svaki koncept. Ovaj proces planiranja koristi reprezentaciju zadataka za poučavanje (eng. teaching tasks) u obliku grafa, kojim se iskazuje pedagoško znanje o tome kako poučavati koncept.

---

Skup generičkih pravila za poučavanje (pravila su neovisna o područnom znanju) upravlja odabirom planova sadržaja i planova prezentiranja na temelju učenikova kognitivnog stila ili preferenci za učenje ((Vassileva, 1995), (Vassileva, 1998)).

**Prilagodljivost** - DCG prvo određuje skup koncepata koje treba poučavati, odnosno dinamički stvara plan sadržaja. Reprerentacija obrazovnih zadataka i metoda dozvoljava sustavu dinamičko planiranje načina prezentiranja sadržaja koji su povezani s određenim konceptom i to na način koji je prilagođen učeniku (koje sadržaja za učenje odabrali i kako ih nizati).

Tijekom prezentiranja sadržaja učeniku, ako učenik točno odgovori na pitanja u testu, pokazuje da je usvojio koncepte te nastavlja s učenjem i poučavanjem prema definiranom planu koji se ne mijenja. Ukoliko učenik pokaže nepoznavanje koncepata, potrebno je pre-planirati računalom oblikovani nastavni sadržaj. Pre-planiranje se prvo radi na prezentacijskom nivou, odnosno mijenja se redoslijed sadržaja za učenje ili pedagoških metoda za prezentiranje određenog koncepta. Ako učenik opet pokaže neznanje, planer sadržaja generira novi niz koncepata koji vode prema ciljnom konceptu u skladu s trenutnim učenikovim znanjem.

**Značajka učenika prema kojoj se prilagođava** - Znanje učenika, kognitivni stil, te način i ritam stjecanja znanja.

**Element prilagođavanja** - Koncept.

**Provjeravanje znanja** - Model učenika se inicijalizira na temelju rezultata ulaznog testa. U bilo kojem trenutku učenik može provjeriti svoje znanje o nekom konceptu. Vježbe i testovi su prezentirani sa skupom manjih jedinica koje sadrže unaprijed pohranjene točne odgovore, pomoć, objašnjenje, itd.

### 8.1.4 ELM-ART 1996

ELM-ART (ELM Adaptive Remote Tutor) je inteligentno okruženje za učenje LISP-a na web-u (Brusilovsky, Schwarz, & Weber, 1996). Nastao je na temeljima sustava ELM-PE (ELM programming environment) koji podržava programiranje temeljeno na primjerima, inteligentnu analizu rješenja problema, napredno testiranje i otklanjanje grešaka (Weber & Moellenberg, 1995). Oba sustava se temelje na ELM (eng. episodic learner modeling) modelu (Weber, 1996). Donošenje zaključaka o učenikovom znanju se temelji na promatranju koje je stranice učenik vidio i možda pročitao.

**Područno znanje** - Konceptualna mreža jedinica.

**Model učenika** - Za svaku stranicu koja je posjećena, odgovarajuća jedinica u modelu učenika se označava kao posjećena (model prekrivanja). Nakon uspješnog rješavanja testa ili problema, usvojeni koncepti jedinice se označavaju kao poznati, te započinje proces donošenja zaključaka o učeniku.



---

**Računalom oblikovani nastavni sadržaj** - Jedinice su hijerarhijski organizirane u lekcije, sekcije, podsekcije i krajnje stranice (eng. terminal pages). Krajnje stranice mogu sadržavati nove koncepte ili probleme koje treba riješiti. Svaka jedinica ima objekt koji sadržava praznine (eng. slot) za tekst koji će se prikazati odgovarajućom stranicom, te za informacije koje se mogu koristiti za međusobno povezivanje jedinica i koncepata.

Statične praznine sadrže informacije o preduvjetnim konceptima, vezanim konceptima, te rezultatima jedinice (usvojenim konceptima). Jedinice koje su krajnje stranice imaju prazninu za test koja sadrži opis pitanja koja učenik treba odgovoriti ili prazninu za opis problema.

Dinamičke praznine su pohranjene zajedno s modelom učenika koji se obnavlja nakon svake učenikove interakcije sa sustavom.

**Prilagodljivost** - Tijekom procesa donošenja zaključaka o učeniku, svi koncepti koji su bili preduvjeti za jedinicu se smatraju zaključenima. Informacije iz dinamičkih praznina u modelu učenika se koriste za individualno označavanje veza i optimalno vođenje učenika kroz računalom oblikovani nastavni sadržaj.

ELM-ART koristi tehniku prilagodljivog označavanja temeljenu na proširenju metafore „semafora“. Također, postoji mogućnost učenja po optimalnoj putanji koja ovisi o trenutnom znanju učenika i preduvjetima stranica u putanji. Prva sljedeća stranica u optimalnoj putanji odgovara onoj čija je veza označena kao „preporuča se“ (eng. suggested).

**Značajka učenika prema kojoj se prilagođava** - Znanje.

**Element prilagođavanja** - Jedinica.

**Provjeravanje znanja** - ELM-ART koristi četiri vrste pitanja: da/ne, prisilni odabir, višestruki odabir, esej. Za svako pitanje potrebno je navesti objašnjenje zašto je točan odgovor baš onaj koji je naveden. Učenik odgovara na pitanja dok god ne odgovori točno zadani broj (*min-problems-solved* parametar između 4 i 10 ovisno o težini testa) pitanja bez puno pogrešaka (*max-errors* parametar je 1). Tek tada sustav može koncepte iz te jedinice označiti kao usvojene. To znači da učenik mora točno odgovoriti minimalni broj pitanja, a može pogrešno odgovoriti na samo jedno pitanje.

### 8.1.5 AST 1997

Adaptive Statistics Tutor (AST) je web orijentirani računalom oblikovani nastavni sadržaj za statistiku (Specht et al., 2001).

**Područno znanje** - Nema ga.

**Model učenika** - Svaka interakcija sa sustavom ima utjecaja na model učenika. Utjecaj ovisi o sadržaju za učenje i o njegovim parametrima. Ovisno o težini i važnosti, povećava se vrijednost pouzdanosti koncepta u probabilističkom modelu učenika. Također se uzima u obzir i postotak točnih testova.

---

Na temelju važnosti i vrijednosti za opću težinu obnavlja se model učenika nakon svakog riješenog testa. Teški testovi s visokom važnošću imaju najveći utjecaj na vrijednost pouzdanosti određene jedinice. Ako učenik prijeđe graničnu vrijednost jedinice, sustav zaključuje da učenik poznaje i jedinicu i njene preduvjete.

**Računalom oblikovani nastavni sadržaj** - Bazu znanja čini konceptualna mreža s različitim vrstama jedinica koje mogu biti lekcije, sekcije, podsekcije i koncepti. Nekoliko vrsta informacija se pridružuje svakom konceptu: tekst, primjeri, demonstracije, „igrališta“ i testovi. Postoje tri razine teksta: od osnovnih informacija do vrlo detaljnog teksta s porukama stručnjama. Osim toga, svaka jedinica ima preduvjete (jedinice koje učenik mora poznavati) i posljedice (mogući utjecaj na druge jedinice). Težina testova i preuvjeta se određuje prema njihovoj važnosti za jedinicu.

**Prilagodljivost** - Učitelji koriste različite strategije za poučavanje različitih vrsta koncepata. Te različite strategije se nazivaju strategijama poučavanja i definiraju određeni redoslijed sadržaja za učenje. Sustav ima predefiniranu strategiju za svaku vrstu koncepta, ali je učitelji mogu promijeniti, ako je to potrebno. Također, postoje pravila za svaku strategiju koja omogućavaju sustavu prilagodljivi odabir strategije poučavanja ovisno o značajkama učenika i o vrsti koncepta koji se poučava.

Sustav prati koje kombinacije i redoslijed sadržaja za učenje učenik često koristi i u skladu s time mijenja strategiju poučavanja. Prilagođavanje prema stilu učenja uzima u obzir učenikov uspjeh dok se poučavao prema tom stilu učenja. Strategija kod koje učenik pokazuje poželjne rezultate na testovima se smatra uspješnim za tog učenika, te postaje predefinirana za tog učenika.

Kada učenik odabere sekciju, sustav provjerava da li posjeduje svo potrebno predznanje, te mu daje test da provjeri poznavanje koncepata za koje nema potvrdu da ih učenik poznaje. Ako učenik ne može riješiti taj test, AST mu predlaže da pvo nauči preduvjete.

Učenik u bilo kojem trenutku može zatražiti da mu sustav sam kaže koja sledeću jedinicu je najbolje učiti. AST to radi na temelju probabilističkog modela prekrivanja i preuvjeta mogućih sljedećih jedinica. Prvo izbacuje one za koje učenik nema ispunjene preduvjete, a zatim promatra pouzdanost i težinu jedinica za koje ima ispunjene preduvjete. Za sljedeću jedinicu se odabire ona koja ima ispunjene najpouzdanije najvažnije preduvjete za učenika.

U AST-u su implemetirane prilagodljiva podrška navigaciji („semafor“), prilagodljivo nizanje i prilagodljivo testiranje.

**Značajka učenika prema kojoj se prilagođava** - Znanje i stilovi učenja.

**Element prilagođavanja** - Jedinica.

**Provjeravanje znanja** - Novi učenici moraju ispuniti upitnik o svojim preferencama, stilu učenja i ciljevima. Učenici mogu naznačiti koja vrsta sadržaja i strategija poučavanja im najviše odgovara (učenje putem primjera, čitanje teksta ili učenje kroz rad). Također mogu odrediti razinu detalja u tekstu.

---

Model učenika se inicijalizira ulaznim testom.

Postoje četiri vrste testova koje se koriste: da/ne, višestruki odabir, prisilni odabir, popunjavanje praznina i esej. Svaki test ima opću vrijednost za težinu, a važnost za preovjeravani koncept se može odrediti. Svaki test je povezan s jednim ili više koncepata na temelju važnosti, te ima vrijednost za opću težinu.

### 8.1.6 ACE 1998

Adaptive *Courseware* Environment (ACE) (Specht & Oppermann, 1998) kombinira podršku prilagodljivoj navigaciji iz sustava ELM-ART (Brusilovsky, Schwarz & Weber, 1996) s planiranjem prezentacije predstavljene u DCG-u (Vassileva, 1995).

**Područno znanje** - Svaki koncept iz grafa koncepata područnog znanja je povezan s različitim vrstama materijala za učenje (tekst, primjer, demonstracija, interaktivno okruženje, test) koji opisuju različite značajke koncepta i imaju različitu ulogu u procesu učenja i poučavanja.

**Model učenika** - Model učenika u ACE-u kombinira model prekrivanja (Carr & Goldstein, 1977) i epizodni pristup (Weber, 1996). Model učenika se sastoji od tri dijela: profila, modela znanja i modela interesa. Profil učenika sadrži informacije o jeziku, medijima, postavkama sučelja, itd. te se može uvijek mijenjati. Model znanja sadrži podatke o jedinicama na kojima je učenik radio i označene su kao naučene jedinice. Naučene jedinice imaju vrijednost pouzdanosti koja ovisi o iskustvu učenika s jedinicom. Model interesa sadrži interesne klastere i dinamički postavlja hipoteze o interesima učenika. Hipoteze se postavljaju na temelju pravila koja povezuju karakteristike epizoda s određenim pretpostavkama o učenikovim interesima.

**Računalom oblikovani nastavni sadržaj** - Struktura računalom oblikovanog nastavnog sadržaja je konceptualna mreža jedinica za učenje koje mogu biti sekcije (koje mogu sadržavati druge jedinice za učenje) ili koncepti (sadrže jednu jedinicu znanja). Svaka od jedinica za učenje može imati preduvjetne jedinice (eng. prerequisite units) koje učenik mora poznavati prije nego počne učiti određenu jedinicu. Preduvjetne jedinice imaju težine određene prema njihovoj važnosti.

Model učitelja predstavlja znanje učitelja o načinu poučavanje neke jedinice. Sastoji se od dva dijela: strategija poučavanja i dijagnoze znanja. Učitelji slijede različite strategije za poučavanje različitih vrsta koncepata. Učitelj mora eksplicitno definirati putanju kroz strukturu područnog znanja (za planiranje prezentacije). Dakle, putanja za učenje se ne generira automatski uzimajući u obzir ciljeve učenja.

Na temelju modela učenika (znanje učenika), područnog znanja i modela učitelja (pravila za učenje i pedagoške specifikacije autora), prezentacijska komponenta ACE-a odabire prikladne jedinice za učenje i generira individualni hipermedijski dokument.

---

**Prilagodljivost** - Strategije poučavanja se prilagodljivo mijenjaju na temelju važnih značajki učenika. Na primjer, ako je učenik već vidio neke od materijala za učenje, oni će biti preskočeni.

Tekst u ACE-u može sadržavati ključne riječi prema konceptima koje učenik još nije naučio. Ako se tekst prezentira učeniku prvi put, onda neće sadržavati hiperveze prema povezanim ali nenaučenim konceptima. Nakon rada u sustavu, ACE će prezentirati sve ključne riječi u tekstu s hipervezama prema naučenim konceptima i prema konceptima koje je učenik spreman naučiti.

Komponenta za prilagodljivo nizanje pokušava zadržati učenika na optimalnoj putanji na temelju učenikova trenutnog znanja. Nadalje, prilagođava nizanje prema interesima koje je specificirao učenik. Optimalna putanja se generira dinamički i svaka učenikova akcija može imati utjecaj na rezultat generiranja. Na primjer, završavanje testa za određenu sekciju omogućava učeniku preskakanje svih koncepata koje ona sadržava.

ACE implementira dvije metode podrške prilagodljivoj navigaciji: prilagodljivo označavanje (eng. adaptive annotation) i inkrementalno povezivanje hiperveza (eng. incremental linking of hyperlinks). Prilagodljivo označavanje hiperveza pruža učeniku dodatne informacije o sadržaju iza hiperveza (koriste se različite boje). Označavanje je prilagođeno učeniku uzimajući u obzir učenikovo znanje i odnose između jedinica koje mora naučiti (koncepti koji su posječeni, koncepti za koje učenik nema predznanje, preporučeni koncepti, jedinice koje nisu preporučene, ali ne zahtjevaju predznanje).

**Značajka učenika prema kojoj se prilagođava** - Znanje, interesi i preference učenika.

**Element prilagođavanja** - Jedinica.

**Provjeravanje znanja** - Dijagnostička komponenta sadrži znanje o nekoliko vrsta testova (kako ih treba generirati i vrednovati). Svaki test u bazi testova može se povezati s više jedinica, a težina mu se određuje prema važnosti za te jedinice. Upravo o težini testa ovisi koliko će utjecaja učenikovi odgovori imati na model učenika.

Kada učenik započne s učenjem novog računalom oblikovanog nastavnog sadržaja, sustav započinje s dijagnostikom učenikovih preferenci, interesa i znanja o području na kojem se uči i poučava. Učenike se u uvodnom upitniku pita o njihovim preferencama vezanima za sadržaje koje će učiti te o općim interesima. Učenik može specificirati strategije poučavanja koje mu odgovaraju (primjeri, čitanje teksta, učenje kroz rad). Također mogu procjeniti svoju stručnost u područnom znanju (početnik, iskusan, stručnjak). Da bi potvrdio učenikovu samo-procjenu, ACE učeniku daje kratki test znanja koji se dinamički generira tako da se uzmu *svi preduvjetni koncepti prve razine i pridruženi im testovi* (u našem pristupu to je reprezentacija). Na temelju rezultata ovog testa se inicijalizira model učenika.

---

Kroz cijeli računalom oblikovani nastavni sadržaj znanje učenika se provjerava tako da se sustav može prilagoditi dinamičkom mijenjanju znanja, interesa i preferenci učenika.

### 8.1.7 KBS Hyperbook System 1999

KBS Hyperbook je prilagodljiva hiperknjiga za kolegij Uvod u računarstvo (Henze & Nejd, 1999).

**Područno znanje** - Koristi se graf zavisnosti kojim se modelira područno znanje. Vrhovi u tom grafu su elementi znanja (eng. knowledge items – KI), a bridovi su zavisnosti između elemenata znanja.

**Model učenika** - Model učenika je Bayesov probabilistički model učenika. Čvorovi u Bayesovoj mreži su elementi znanja. Zavisnosti između elemenata znanja su iskazane uvjetnim vjerojatnostima između elemenata znanja.

**Računalom oblikovani nastavni sadržaj** - Informacijske jedinice (eng. information units) odgovaraju semantičkim djelovima knjige i semantički su međusobno povezane. Ta semantička povezanost određuje strukturu računalom oblikovanog nastavnog sadržaja. Informacijske jedinice su indeksirane elementima znanja. Na svakoj stranici hiperknjige postoji jedan ili više glavnih elemenata znanja. Za svaki element znanja postoji jedna informacijska jedinica za koju je taj element znanja glavni. Postoje i druge informacijske jedinice gdje se javlja isti element znanja, ali za te informacijske jedinice on nije glavni.

Projektne jedinice (eng. project units) sadrže opis projekta (vježbe ili primjere riješenih problema). Projektne jedinice su indeksirane onim elementima znanja koji su učeniku potrebni da bi uspješno realizirao projekt. Za rad na jednoj projektnoj jedinici, sustav mora generirati niz odgovarajućih informacijskih jedinica (koristeći indeksiranja preko elemenata znanja).

Učenik odabire svoj cilj učenja, a sustav mu predlaže projektne jedinice na kojima treba raditi da bi ostvario svoj cilj (eng. adaptive project selection). Također, sustav može sam predložiti ciljeve učenja koji su u skladu s učenikovim znanjem (eng. adaptive goal selection), a zatim predložiti projekte, te generirati računalom oblikovani nastavni sadržaj.

**Prilagodljivost** - Iz niza se izostavljaju već poznate informacijske jedinice, a raspoložive informacijske jedinice se u računalom oblikovani nastavni sadržaj nižu prema njihovoj složenosti. Ako učenik nema predznanje potrebno za rad na nekom projektu, onda se na početak računalom oblikovanog nastavnog sadržaja dodaju i informacijske jedinice koje se odnose na nedostajuće elemente znanja. Sustav koristi prilagodljivu podršku navigaciji („semafor“) kojom prilagođava prikaz liste informacijskih jedinica u računalom oblikovanom nastavnom sadržaju.

---

**Značajka učenika prema kojoj se prilagođava** - Znanje i ciljevi učenja.

**Element prilagođavanja** - Jedinica.

**Provjeravanje znanja** - Nema. Samoocjenjivanje i vansko procjenjivanje znanja učenika od strane stručnjaka.

### 8.1.8 ATLAS 2001

ATLAS (Authoring Tool for Adaptive Software design) je grafički alat za oblikovanje interaktivnog dinamički prilagodljivog računalom oblikovanog nastavnog sadržaja ((Macías & Castells, 2001), (Macías & Castells, 2002)). ATLAS pruža razumijevanje odnosa između strukture, sadržaja, prezentacije računalom oblikovanog nastavnog sadržaja, te modela učenika. Učitelj stvara, mijenja i pridružuje zadatke, sadržaj i model učenika. ATLAS zapravo predstavlja grafičko sučelje za oblikovanje računalom oblikovanog nastavnog sadržaja, koji se zatim, kao rezultat, predaje sustavu TANGOW (Task-based Adaptive learner Guidance On the Web) u procesu izvođenja ((Carro, Pulido, & Rodríguez, 1999), (Carro, Moriyón, Pulido, & Rodríguez, 1999)).

TANGOW oblikuje računalom oblikovani nastavni sadržaj u formi hijerarhije zadataka koji reprezentiraju didaktičke jedinice koje učenici mogu usvajati. Prilikom stvaranja novog računalom oblikovanog nastavnog sadržaja, učitelj mora napraviti konceptualnu dekompoziciju na zadatke, te uspostaviti veze između zadataka uz pomoć pravila.

**Područno znanje** - Nema.

**Model učenika** - Profil učenika sadržava osobne značajke kao što su učenikova dob i materinji jezik, odnos učenika i područnog znanja, itd. Model učenika se sastoji od skupa jednostavnih i složenih atributa u obliku stabla, koje učitelj može mijenjati i koristiti za definiranje uvjeta koji će određivati strukturu računalom oblikovanog nastavnog sadržaja.

**Računalom oblikovani nastavni sadržaj** - Učitelj definira strukturu računalom oblikovanog nastavnog sadržaja na temelju modela učenikovih zadataka koji su hijerarhijski organizirani (prema TANGOW modelu). Učitelju su na raspolaganju sadržaji u formi HTML fragmenata. Također, učitelj može vidjeti koje učenikove značajke su važne za računalom oblikovani nastavni sadržaj kojeg oblikuje.

Zadatak je osnovna jedinica računalom oblikovanog nastavnog sadržaja koji ima hijerarhijsku strukturu. Definira se određivanjem vrijednosti sljedećim atributima: naziv, opis, vrsta (teorijski, praktični, primjer), složenost (nedjeljiv i složen), zahtjevi za ispunjenje. Zadaci također sadrže i popis multimedijalnih elemenata (tekst, slika, video, zvuk, animacija, itd.) koji se koriste za generiranje HTML stranica. HTML stranice se koje prezentiraju učeniku, dinamički se generiraju. Opisi komponenti koje će se pojaviti na stranicama uključeni su kao

---

vrijednosti HTML polja u odgovarajućim zadacima. Ti opisi moraju sadržavati vrstu multimedijskog elementa koji će se nalaziti na stranici, kao i njegov položaj na stranici. Složeni zadatak se može dekomponirati na temelju pravila. Svako pravilo ima naziv, te sadrži informacije o zadatku kojeg treba podijeliti u podzadatke, popis podzadataka i ključnu riječ koja opisuje nizanje podzadataka (AND, ANY, XOR, OR).

**Prilagodljivost** - Računalom oblikovani nastavni sadržaj koji se stvara uz pomoć TANGOW-a se dinamički prilagođava svakom učeniku uzimajući u obzir njegov profil i akcije, kao i različite strategije poučavanja.

**Značajka učenika prema kojoj se prilagođava** - Profil učenika i akcije učenika.

**Element prilagođavanja** - Zadatak.

**Provjeravanje znanja** - Prvi put kada učenik pristupi sustavu, mora odgovoriti na pitanja u testu vezana za osobne podatke i preference.

### 8.1.9 INSPIRE 2001

INSPIRE (INtelligent System for Personalized Instruction in a Remote Environment) je sustav prilagodljive hipermedije ((Grigoriadou, Papanikolaou, Kornilakis, & Magoulas, 2001), (Papanikolaou, Grigoriadou, Kornilakis, & Magoulas, 2003)).

**Područno znanje** - Područno znanje je prikazano mrežom koncepata. Svaki koncept ima pridružen obrazovni sadržaj koji se sastoji od modula (prezentacija teorije, pitanja, primjeri, vježbe, aktivnosti, definicije, itd.).

**Model učenika** - Model učenika ima sljedeće karakteristike: (1) to je model prekrivanja, (2) pamti informacije koje opisuju učenikovu interakciju koja predstavlja stav učenika prema učenju, (3) pamti opće podatke o učeniku: ime, zanimanje, spol, stil učenja, te (4) vidljiv je učeniku koji ga može kontrolirati.

**Računalom oblikovani nastavni sadržaj** - Cilj učenja odgovara skupu koncepata područnog znanja. Koncepti koji su ključni za ostvarivanje cilja se nazivaju temeljni koncepti (eng. outcome concepts). Da bi ostvario cilj, učenik mora učiti sve temeljne koncepte i pokazati njihovo poznavanje. Preduvjeti i vezani koncepti su pridruženi svakom temeljnom konceptu. Preduvjete se mora učiti za jedno s obrazovnim sadržajima temeljnog koncepta, a vezani koncepti su primitivni koncepti koji se koriste u obrazovnim sadržajima temeljnog koncepta.

Svaka lekcija uključuje cilj učenja, temeljne koncepte i obrazovne sadržaje koji su pridruženi temeljnim konceptima i njihovim preduvjetnim konceptima.

**Prilagodljivost** - Na temelju cilja učenja kojeg učenik odabere, sustav generira lekcije usklađene s učenikovim stilom učenja i znanjem. Učenik može utjecati na proces generiranja

---

lekcije tako što mu je dozvoljeno iskazati mišljenje o svojim karakteristikama ili o sadržaju lekcije. Sustav koristi prilagodljivu podršku navigaciji i prilagodljivu prezentaciju.

**Značajka učenika prema kojoj se prilagođava** - Znanje i stil učenja.

**Element prilagođavanja** - Lekcija.

**Provjeravanje znanja** - Pitanja koja su pridružena svakom konceptu.

### 8.1.10 WINDS 2001

WINDS (Web-based Intelligent Design and Tutoring System) (Specht et al., 2001) je autorski alat za upravljanje sadržajem koji je usko povezan s ACE (Specht & Oppermann, 1998). Ovo osigurava stvaranje individualiziranog računalom oblikovanog nastavnog sadržaja.

**Područno znanje** - Nema.

**Model učenika** - Model učenika pamti sve obrazovne događaje. Ovi događaji se nazivaju epizodama i uključuju učenikove operacije nad objektima učenja, vrednovanja njihovog rada od strane učitelja i sustava, te sve zaključke koje je sustav donio o njegovoj razini znanja. Ovakav model se naziva epizodni model (Weber, 1996).

**Računalom oblikovani nastavni sadržaj** - Sustav omogućava izgradnju nezavisnih objekata učenja i njihovo strukturiranje u predefimirani računalom oblikovani nastavni sadržaj, kao i njihovu uporabu u novim dinamički generiranim računalom oblikovanim nastavnim sadržajima.

WINDS sadrži četiri tipa objekata učenja: (1) kolegij (eng. course unit) je element na najvišoj razini koji može imati samo podjedinice; (2) jedinice za učenje (eng. learning units) predstavljaju temelje za strukturiranje računalom oblikovanog nastavnog sadržaja; (3) elementi za učenje (eng. learning elements) su osnovni komadići informacija s predlošcima za različite pedagoške svrhe; te (4) indeksni pojmovi (eng. indeks terms) su osnovni pojmovi u riječniku koji omogućavaju povezivanje heterogenih sadržaja i individualnih putanja kroz materijale za učenje.

Elementi za učenje mogu biti: (1) paragrafi koji se sastoje od blokova sadržaja i mogu imati različite pedagoške funkcije; (2) vježbe, te (3) zadaci objektivnog tipa kojima se vrednuje znanje učenika. Elementi za učenje imaju predefimirani redoslijed unutar jedinice za učenje. Elementi za učenje se sastoje od blokova sadržaja koji imaju predifinirani redoslijed i pedagošku ulogu

U WINDS-u je omogućen pregled svih kolegija i elemenata za učenje, kao i pregled strukture računalom oblikovanog nastavnog sadržaja (koji elementi za učenje čine koje jedinice za učenje).



---

**Prilagodljivost** - Sustav omogućava generiranje prilagodljivog obrazovnog hipermedijskog računalom oblikovanog nastavnog sadržaja s personaliziranim nizanjem kurikuluma i personaliziranim odabirom sadržaja na razini jedinica i stranica. Struktura računalom oblikovanog nastavnog sadržaja i model učenika omogućavaju prilagodljivu navigaciju i prezentiranje. Između objekata za učenje i elemenata za učenje postoje različite veze kao što su preduvjeti, „dio od“, „se odnosi na“. Uspoređujući model učenika s ovim relacijama, struktura računalom oblikovanog nastavnog sadržaja i navigacija se mogu dinamički prilagoditi učeniku. Na temelju trenutnog modela učenika, odnosa između objekata učenja i njihovih pedagoških uloga, WINDS prilagođava prezentiranje elemenata za učenje.

Pošto su objekti učenja međusobno povezani, mogu se koristiti tehnike prilagođavanja kao što su: direktno vođenje (eng. direct guidance), sortiranje veza (eng. link sorting), skrivanje veza (eng. link hiding), označavanje veza (eng. link annotation). Elementi za učenje se sastoje od blokova sadržaja koji imaju preddefinirani redoslijed i pedagošku ulogu, što omogućava primjenu metoda prilagođavanja kao što su: dodatno objašnjavanje (eng. additional explanation), objašnjavanje preduvjeta (eng. prerequisite explanation), usporedno objašnjavanje (eng. comparative explanation), varijacije objašnjavanja (eng. explanation variants), sortiranje.

**Značajka učenika prema kojoj se prilagođava** - Znenje, preference i stil učenja.

**Element prilagođavanja** - Element za učenje.

**Provjeravanje znanja** - Zadaci objektivnog tipa.

### 8.1.11 NetCoach 2002

NetCoach je autorski sustav za stvaranje web-orijentiranog prilagodljivog courseware-a (Weber et al., 2002). NetCoach je nastao na temeljima sustava ELM-ART (Weber & Specht, 1997).

**Područno znanje** - Područno znanje se sastoji od koncepata koji predstavljaju reprezentaciju stranica koje će biti prezentirane učeniku. Među konceptima postoje dvije vrste veza: preduvjet i zaključak. Učitelj definira koncepte koji predstavljaju preduvjete za usvajanje određenog koncepta. Pošto i preduvjetni koncepti mogu imati i svoje preduvjete, oni su posredni preduvjeti određenog koncepta. Na taj način se definiraju i zaključni koncepti određenog koncepta. To su oni koncepti kojima je odabrani koncept preduvjet. Svaki koncept područnog znanja je vezan s pitanjima. Skupovi pitanja omogućavaju vrednovanje učenikovog poznavanja nekog koncepta. Istim pitanjem može se provjeravati poznavanje više koncepata.

**Model učenika** - Model učenika je višeslojni model prekrivanja. Prvi sloj opisuje da li je učenik posjetio stranicu koja sadrži određeni koncept. Drugi sloj sadrži informaciju o pitanjima

vezanima s određenim konceptom na koja je učenik odgovarao i kakvo je znanje pokazao. Treći sloj opisuje da li se može zaključiti da učenik poznaje određeni koncept na temelju veze tog koncepta i naprednijih koncepata koje učenik poznaje. Četvrti sloj opisuje da li je učenik označio da određenik koncept već poznaje.

**Računalom oblikovani nastavni sadržaj** - Cilj učenja je skup koncepata koje učenik mora naučiti. Svi preduvjeti (posredni i neposredni) se određuju automatski i odgovarajuće stranice se predlažu. Na temelju opisa koncepata, sve stranice se generiraju individualno u skladu s modelom učenika.

**Prilagodljivost** - Računalom oblikovani nastavni sadržaj je adaptivan, interaktivan, adaptabilan i komunikativan. Sustav koristi prilagodljivo označavanje veza. Sljedeća stranica koja se preporuča učeniku za učenje se dinamički generira na temelju cilja učenja i učenikova znanja. Učenici dobiju upozorenje ako pogledaju stranicu za koju nemaju sve preduvjete (upozorenja se mogu isključiti).

**Značajka učenika prema kojoj se prilagođava** - Znanje i ciljevi učenja.

**Element prilagođavanja** - Stranica.

**Provjeravanje znanja** - Sustav omogućava prezentiranje vježbi i testova u različitim formatima: višestruki odabir, prisilni odabir, popunjavanje praznina, esej (primjer kao povratna informacija - samovrednovanje), e-mail (vrednuje učitelj). Povratna informacija se sastoji u navođenju točnog odgovora i zašto je taj odgovor točan zajedno ili odvojeno. Povratnu informaciju definira učitelj.

### 8.1.12 APeLS 2002

Adaptive Personalized e-Learning Service (APeLS) predstavlja računalom oblikovani nastavni sadržaj korištenjem opisa ili bilješki (eng. narratives) i kandidatski grupa (eng. candidate group) (Bruen et al., 2002). Ovaj pristup sadrži model sadržaja, model učenika i model opisa (eng. narrative model).

**Područno znanje** - Model opisa stvara stručnjak područnog znanja. Opis je reprezentacija stručnjakovog znanja.

**Model učenika** - Model učenika sadrži predznanje i ciljeve učenje te olakšava personaliziranu isporuku sadržaja temeljenih na učenikovom iskustvu i ciljevima.

**Računalom oblikovani nastavni sadržaj** - Opis predstavlja niz koraka kroz sadržaj u kojem se svaki korak sastoji od skupova obrazovnih resursa koji imaju isti cilj učenja (pripadaju istoj kandidatskoj grupi). Svaka kandidatska grupa sadrži objekte učenja koji ispunjavaju iste zahtjeve sadržaja. Objekti učenja u kandidatskoj grupi se razlikuju u tehničkom, nastavnom ili

---

u bilo kojem drugom pogledu koji se može prilagođavati. Planiranje prezentacije je ograničeno na biranje određene kandidatske grupe. Kandidatske grupe su unaprijed definirane i imaju različite strukture i format obrazovnih resursa. Razdvojenost sadržaja od njegovih opisa pospješuje ponovnu upotrebljivost objekata učenja. Računalom oblikovani nastavni sadržaj je organiziran u sekcije, module i jedinice.

**Prilagodljivost** - Stroj za prilagođavanje (eng. adaptive engine) usklađuje model sadržaja, model učenika i model opisa da bi u trenutku izvođenja, dinamički stvorio personalizirani računalom oblikovani nastavni sadržaj. Opisi se odnose na kandidatske grupe, a stroj za prilagođavanje odabire iz te grupe kandidata koji je najprikladniji za isporuku. Pravila prilagođavanja definiraju učitelji. Model opisa sadrži i pravila prilagođavanja. Opisi omogućavaju razdvajanje pravila koja upravljaju načinom generiranja personaliziranog računalom oblikovanog nastavnog sadržaja, od sadržaja koji će biti uključen u računalom oblikovani nastavni sadržaj.

**Značajka učenika prema kojoj se prilagođava** - Znanje i stil učenja.

**Element prilagođavanja** - Kandidatska grupa.

**Provjeravanje znanja** - Prije nego što APeLS može stvoriti personalizirani računalom oblikovani nastavni sadržaj, mora imati prikladne informacije o učeniku. Ove informacije se dobivaju kroz upitnik o predznanju i stilu učenja. Istom upitniku učenik može pristupiti u bilo kojem dijelu procesa učenja i poučavanja čime je omogućeno mijenjanje modela učenika i ponovno generiranje personaliziranog računalom oblikovanog nastavnog sadržaja.

### 8.1.13 ACCT 2004

Adaptive Course Construction Toolkit (ACCT) je skup alata za stvaranje računalom oblikovanog nastavnog sadržaja (prilagodljivog ili ne) ((Dagger, Wade, & Conlan, 2004), (Dagger, Wade & Conlan, 2005)). Sadrži alate za generiranje ontologije područnog znanja, opisa, repozitorija obrazovnih resursa, testova. ACCT eksportira računalom oblikovani nastavni sadržaj sustavu APeLS (Adaptive Personalized eLearning Service) (Bruen, Conlan & Wade, 2002) zajedno s apodručnim znanjem i opisima.

**Područno znanje** - Ontologija područnog znanja.

**Model učenika** - Nema.

**Računalom oblikovani nastavni sadržaj** - Model opisa obuhvaća semantiku pedagoške strategije koja se koristi. On opisuje logiku na kojoj leži odabir i isporuke aktivnosti za učenje i konceptata. Struktura opisa (eng. Narrative Structure) se sastoji od kolekcije opisa konceptata (eng. Narrative Concepts). Opis koncepta dozvoljava učitelju primjenu aspekata

---

pedagoške strategije na određene dijelove prilagodljivog računalom oblikovanog nastavnog sadržaja.

**Prilagodljivost** - Da bi računalom oblikovani nastavni sadržaj postao prilagodljiv, učitelj mora pridružiti opisne attribute (eng. Narrative Attributes) konceptima i aktivnostima za učenje. Opisni atributi se sastoje od karakteristike prilagođavanja (znanje, stil učenja. itd.), tehnike prilagođavanja (označavanje veza, prilagodljiva podrška navigaciji, itd.) i opisa smjernica/uporabljivosti. Učitelj određuje koja tehnika prilagođavanja će se koristiti za koje karakteristike prilagođavanja.

Odabrani obrazovni resursi čine apstraktne kandidatske grupe, koje odabrana tehnika prilagođavanja koristi prilikom usklađivanja prilagodljivog računalom oblikovanog nastavnog sadržaja. Iz kandidatske grupe se, na temelju opisnih atributa, odabire jedan obrazovni resurs.

**Značajka učenika prema kojoj se prilagođava** - Ciljevi učenja, stil učenja, znanje.

**Element prilagođavanja** - Kandidatska grupa.

**Provjeravanje znanja** - Nema.

### 8.1.14 ASM 2005

Karampiperis i Sampson su predložili metodologiju za prilagodljivo nizanje (eng. Adaptive Sequencing Methodology - ASM) u kojoj koriste statističke metode za pronalaženje najbolje putanje za učenje (Karampiperis & Sampson, 2005).

**Područno znanje** - Područno znanje je iskazano ontologijom područnog znanja. Graf putanja koncepata (eng. Concepts Path Graph) je jednostavni aciklični usmjereni graf koji reprezentira strukturu koncepata ontologije područnog znanja. Koncepti iz grafa putanje koncepata se odabiru na temelju veza između hijerarhije ciljava učenja i ontologije područnog znanja.

**Model učenika** - Model prekrivanja se koristi za reprezentiranje prostora učenikova znanja (eng. Learners Knowledge Space), a stereotipni model za reprezentiranje učenikovog stila učenja prema Honey i Mumford modelu (Honey & Mumford, 1992), te za način učenja (vizualni, tekstualni, auditorni, miješani). Za utvrđivanje učenikove razine znanja, prati se broj bodova postignut na testovima, kao i broj pokušaja rješavanja testova.

**Računalom oblikovani nastavni sadržaj** - Graf putanja učenja (eng. Learning Paths Graph) je usmjereni graf koji sadrži sve moguće putanje učenja (nizove obrazovnih resursa) koje dovode do ostvarivanja cilja. Dobiva se tako da se svaki koncept iz grafa putanja koncepata

---

zamijeni s odgovarajućim skupom obrazovnih resursa. Graf putanja učenja je također jednostava aciklički usmjereni graf.

**Prilagodljivost** - Na temelju *funkcije prikladnosti* (eng. suitability function), koja pridružuje težine bridovima u grafu putanja učenja, i algoritma za traženje najkraće putanje, bira najprikladnija putanja učenja iz grafa putanja učenja. Odabrana putanja se naziva *optimalna putanja za učenje* (eng. optimal learning path) i nasljeđuje odnose iz strukture područnog znanja i one odnose koji postoje između resursa. Funkcija prikladnosti, kroz statističke metode, određuje koliko je određeni obrazovni resurs prikladan za određenog učenika. Ona uspoređuje karakteristike objekta učenja s karakteristikama učenika, i obratno.

**Značajka učenika prema kojoj se prilagođava** - Znanje i stilovi učenja.

**Element prilagođavanja** - Koncept.

**Provjeravanje znanja** - Spominje se vrednovanje znanja, ali se ne opisuje.

### 8.1.15 iClass 2006

Nakon rada na sustavima APeLS i ACCT, Conlan i Wade su nastavili svoje istraživanje u okviru projekta Intelligent Distributed Cognitive-based Open Learning System for Schools (iClass) (O Keeffe, Brady, Conlan, & Wade, 2006). U iClass-u generiranje kolegija se inteligentno prilagođava učeniku prema njegovim kognitivnim značajkama. Za razliku od APeLS, iClass odvajava pedagoške informacije i strukturu područnog znanja.

**Područno znanje** - Ontologija područnog znanja.

**Model učenika** - Nije opisan.

**Računalom oblikovani nastavni sadržaj** - Pedagoške strategije su skupovi pravila koje određuju pristup za prezentiranje koncepata. Određuje se na temelju preferenci i učenika (način učenja) i učitelja (način poučavanja).

**Prilagodljivost** - Personalizirana putanja učenja se sastoji od koncepata, a određuje se u skladu s odabranom pedagoškom strategijom. Učitelj određuje granice prilagođavanja. Pristup je sličan onom u sustavima APeLS (Bruen, Conlan & Wade, 2002) i WINDS (Specht, Kravcik, Pesin & Klemke, 2001).

Poboljšana personalizacija se realizira primjenom različitih pedagoških strategija na isto područno znanje. To znači da se prilagodljivo odabire najprikladnija strategija poučavanja bez obzira na to što se poučava.

**Značajka učenika prema kojoj se prilagođava** - Znanje, ciljevi i preference učenika.

**Element prilagođavanja** - Objekt učenja.

---

**Provjeravanje znanja** - Nije opisano.

### 8.1.16 PAIGOS 2008

PAIGOS je model generiranja računalom oblikovanog nastavnog sadržaja koji raspolaže velikim pedagoškim znanjem, te je neovisan o teoriji učenja (Ullrich, 2008). PAIGOS generiranje računalom oblikovanog nastavnog sadržaja temelji na hijerarhijskog mrežnog planiranja zadataka (eng. hierarchical task network planning - HTN planning), koje je prvi put implementirano u okruženju za e-učenje FORMACION HUMANA (FORHUM) (Mendez, Ramirez, & Luna, 2005). ActiveMath je inteligentni sustav e-učenja koji koristi generator računalom oblikovanog nastavnog sadržaja PAIGOS (Melis, Gogvadze, Libbrecht, & Ullrich, 2009).

Kod HTN planiranja, cilj je postići djelomično ili potpuno uređenu listu aktivnosti koje se nazivaju *zadaci*. Ti zadaci se rješavaju dekomponiranjem na podzadatke sve dok se ne dođe do primitivnih zadataka koji se mogu direktno izvršiti. U FORHUM-u se obrazovni ciljevi (slično konceptima strukture područnog znanja) ostvaruju obrazovnim resursima koji se nazivaju obrazovne jedinice. Svaka obrazovna jedinica ima pridruženi stil učenja. Za svaki koncept se generira HTN metoda. Generiranje pedagoškog znanja se radi automatski nakon definiranja strukture područnog znanja. FORHUM poučava prvo potrebna predznanja, te zatim bira obrazovne resurse prema stilu učenja.

U PAIGOS-u generirani kolegiji su strukturirani i prilagođeni različitim ciljevima učenja i učenikovim sposobnostima. PAIGOS ima eksplicitnu prezentaciju pedagoškog znanja napravljenu na temelju znanja pedagoških stručnjaka i pedagoške teorije. On dozvoljava generiranje kolegija za iste koncepte, ali s različitim ciljevima učenja.

**Područno znanje** - Reference na obrazovne resurse.

**Model učenika** - Nema.

**Računalom oblikovani nastavni sadržaj** - Računalom oblikovani nastavni sadržaj se sastoji od strukturiranih nizova referenci na obrazovne resurse. Isti obrazovni resurs se može koristiti u više računalom oblikovanog nastavnog sadržaja i podupirati ostvarivanje različitih ciljeva učenja.

Generiranje računalom oblikovanog nastavnog sadržaja se može zaustaviti na razini koja određuje vrstu obrazovnog resursa, ali ne određuje točno koje obrazovne resurse uzeti u obzir. Konkretni obrazovni resursi se odabiru što je kasnije moguće, odnosno, u trenutku kada ih učenik želi vidjeti.

**Prilagodljivost** - Nije eksplicitno navedeno.

**Značajka učenika prema kojoj se prilagođava** - Nije eksplicitno navedeno.

---

**Element prilagođavanja** - Nije eksplicitno navedeno.

**Provjeravanje znanja** - Nije opisano.

### 8.1.17 ADOPTA 2009

ADOPTA (ADaptive technOlogy-enhanced Platform for eduTAInment) je okruženje za generiranje prilagodljivog računalom oblikovanog nastavnog sadržaja u sustavima e-učenja (Bontchev & Vassileva, 2009). Računalom oblikovani nastavni sadržaj se u ovom okruženju naziva priča (eng. *storyboard*).

**Područno znanje** - Sadržaj za učenje se granulira u objekte učenja koji mogu biti opisi, zadaci ili pitanja. Objekti učenja su organizirani u semantički ontološki graf, te imaju dvije vrste veza: *is\_a* za veze između opisa, te *has\_a* za veze između zadataka i pitanja i opisa. Potrebno je oblikovati što više objekata učenja različite vrste koji su prikladni za bilo koji stil učenja, čime se učiteljima omogućava generiranje različitih putanja učenja za različite stilove učenja.

**Model učenika** - Model učenika sadrži ciljeve i preference, stil učenja, te znanje.

**Računalom oblikovani nastavni sadržaj** - Učitelj smješta objekte učenja na stranice priče. Za svaki objekt učenja, učitelj definira koeficijent razine znanja koji koristi mehanizam za prilagodljivo odabiranje sadržaja na temelju učenikovog znanja koje je pokazao na zadnjoj kontrolnoj stranici.

Učitelj također mora definirati parametre stranica računalom oblikovanog nastavnog sadržaja tako da se mogu prilagoditi učenikovom stilu učenja i rezultatima provjere znanja.

**Prilagodljivost** - Prilikom isporuke računalom oblikovanog nastavnog sadržaja koristi se prilagodljiva podrška navigaciji, prilagodljivo označavanje (stil učenja) i prilagodljivi odabir sadržaja (znanje). Znanje se provjerava pomoću automatski generiranih testova na kontrolnim točkama unutar priče.

Putanja učenja se odabire na način da se uzme putanja najveće težine. Prikazuju se samo prikladni objekti učenja. Na kraju svake putanje učenja određuju se težine na prijedenoj putanji, te se odlučuje može li učenik nastaviti ili se mora vratiti na početak putanje.

**Značajka učenika prema kojoj se prilagođava** - Znanje i stil učenja.

**Element prilagođavanja** - Objekt učenja.

**Provjeravanje znanja** - Sadržaj kontrolnih stranica se automatski generira odabirom objekata učenja tipa pitanje koji se odnose na objekte učenja koji se nalaze na stranicama koje je učenik učio. Učitelj postavlja granicu za uspješno savladavanje sadržaja.

## 8.2 Područno znanje - Računalo kao sustav

Znanje o računalu kao sustavu je uzeto za područno znanje u modelu automatskog i dinamičkog generiranja i prilagodljivog odabira, nizanja i prezentiranja računalom oblikovanog nastavnog sadržaja, a služilo je i kao područno znanje u prototipu inteligentnog tutorskog sustava AC-ware Tutor nad kojim je provedeno prototipno testiranje.

### 8.2.1 Popis koncepata i relacija u ontologiji „Računalo kao sustav“

U sljedećoj tablici atribut Type označava vrstu koncepta: 0 – „obični“ koncept (76), 1 – atribut (3), 2 – vrijednost (9).

NodeID	Name	Type
1	Aplikacijska programska podrška	0
2	Aritmetička operacija	0
3	Aritmetičko-Logička jedinica	0
4	Asembler	0
5	Basic	0
6	Binarni brojevni sustav	0
7	Brojevni sustav	0
8	C	0
9	Centralna jedinica	0
10	Centralna procesorska jedinica	0
11	Dekadski brojevni sustav	0
12	Disjunkcija	0
13	Disketa	0
14	DOS	0
15	Fortran	0
16	Heksadekadski brojevni sustav	0
17	I sklop	0
18	ILI sklop	0
19	Informacija	0
20	Instrukcija	0
21	Interpretator	0
22	Izlazna jedinica	0
23	Jezični prevoditelji	0
24	Kompaktni disk	0
25	Kompilator	0
26	Konjunkcija	0
27	Logička operacija	0
28	Logički sklop	0
29	Masovna memorija	0
30	Memorija	0
31	Miš	0
32	Model računalnog sustava	0
33	Modem	0
34	Monitor	0
35	Mrežna kartica	0
36	NE sklop	0
37	Negacija	0
38	NI sklop	0
39	NILI sklop	0
40	Obrada podataka	0
41	Oduzimanje	0
42	Oktalni brojevni sustav	0

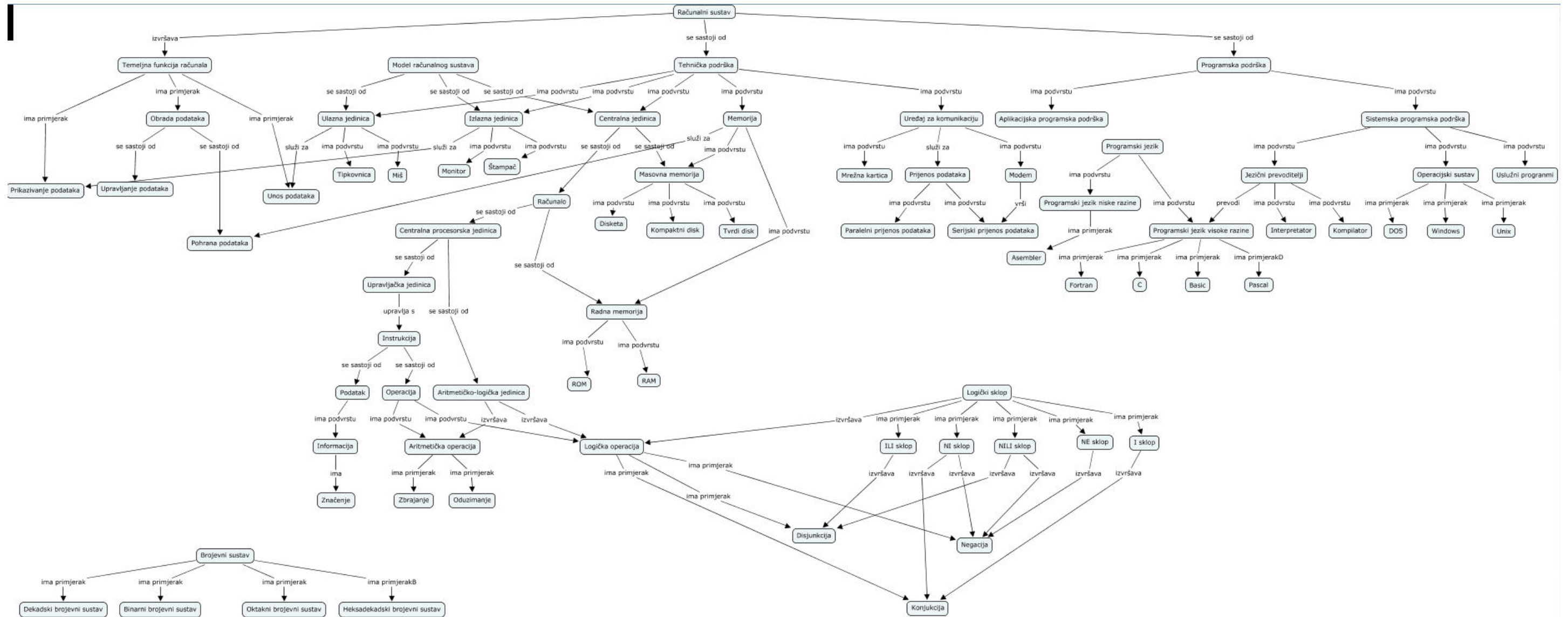


43	Operacija	0
44	Operacijski sustav	0
45	Paralelni prijenos podataka	0
46	Pascal	0
47	Podatak	0
48	Pohrana podataka	0
49	Prijenos podataka	0
50	Prikazivanje podataka	0
51	Programska podrška	0
52	Programski jezik	0
53	Programski jezik niske razine	0
54	Programski jezik visoke razine	0
55	Računalni sustav	0
56	Računalo	0
57	Radna memorija	0
58	RAM	0
59	ROM	0
60	Serijski prijenos podataka	0
61	Sistemska programska podrška	0
62	Štampač	0
63	Tehnička podrška	0
64	Temeljna funkcija računala	0
65	Tipkovnica	0
66	Tvrđi disk	0
67	Ulazna jedinica	0
68	Unix	0
69	Unos podataka	0
70	Upravljačka jedinica	0
71	Upravljanje podataka	0
72	Uređaj za komunikaciju	0
73	Uslužni programi	0
74	Windows	0
75	Zbrajanje	0
76	Značenje	0
77	Baza	1
78	Kapacitet	1
79	Znamenke	1
80	0 1	2
81	2	2
82	8	2
83	10	2
84	16	2
85	0 1 2 3 4 5 6 7	2
86	0 1 2 3 4 5 6 7 8 9	2
87	0 1 2 3 4 5 6 7 8 9 A B C D E F	2
88	1.44MB	2

U sljedećoj tablici je prikazano 9 relacija koje se koriste za povezivanje koncepata u područnom znanju „Računalo kao sustav“:

LinkID	Name
1	izvršava
2	ima primjerak
3	se sastoji od
4	ima podvrstu
5	prevodi
6	služi za
7	vrši
8	upravlja s
9	ima

### 8.2.2 Ontologija "Računalo kao sustav" prikazana u sustavu Cmap Tools COE – grafički prikaz



## 8.2.3 Ontologija "Računalo kao sustav" prikazana u sustavu Cmap Tools COE – LifeMap prikaz

U područnom znanju „Računalo kao sustav“ postoji 91 uređena trojka nastala povezivanjem koncepata relacijama. Također, postoji 9 okvira.

Računalni sustav	izvršava	Temeljna funkcija računala
Temeljna funkcija računala	ima primjerak	Unos podataka
Temeljna funkcija računala	ima primjerak	Obrada podataka
Obrada podataka	se sastoji od	Pohrana podataka
Obrada podataka	se sastoji od	Upravljanje podataka
Temeljna funkcija računala	ima primjerak	Prikazivanje podataka
Računalni sustav	se sastoji od	Programska podrška
Programska podrška	ima podvrstu	Sistemska programska podrška
Sistemska programska podrška	ima podvrstu	Operacijski sustav
Operacijski sustav	ima primjerak	Unix
Operacijski sustav	ima primjerak	DOS
Operacijski sustav	ima primjerak	Windows
Sistemska programska podrška	ima podvrstu	Jezični prevoditelji
Jezični prevoditelji	ima podvrstu	Interpretator
Jezični prevoditelji	ima podvrstu	Kompilator
Jezični prevoditelji	prevodi	Programski jezik visoke razine
Programski jezik visoke razine	ima primjerak	Basic
Programski jezik visoke razine	ima primjerak	Fortran
Programski jezik visoke razine	ima primjerak	C
Programski jezik visoke razine	ima primjerak	Pascal
Sistemska programska podrška	ima podvrstu	Uslužni program
Programska podrška	ima podvrstu	Aplikacijska programska podrška
Računalni sustav	se sastoji od	Tehnička podrška
Tehnička podrška	ima podvrstu	Memorija
Memorija	ima podvrstu	Radna memorija
Radna memorija	ima podvrstu	ROM
Radna memorija	ima podvrstu	RAM
Memorija	ima podvrstu	Masovna memorija
Masovna memorija	ima podvrstu	Tvrđi disk
Masovna memorija	ima podvrstu	Kompaktni disk
Disketa	filler	Disketa
Disketa	filler	1.44MB
Kapacitet	filler	Kapacitet
Memorija	služi za	1.44MB
Tehnička podrška	ima podvrstu	Pohrana podataka
Uređaj za komunikaciju	ima podvrstu	Uređaj za komunikaciju
Modem	vrši	Modem
Uređaj za komunikaciju	služi za	Serijski prijenos podataka
Prijenos podataka	ima podvrstu	Prijenos podataka
Prijenos podataka	ima podvrstu	Serijski prijenos podataka
Uređaj za komunikaciju	ima podvrstu	Paralelni prijenos podataka
Tehnička podrška	ima podvrstu	Mrežna kartica
Ulazna jedinica	služi za	Ulazna jedinica
Ulazna jedinica	ima podvrstu	Unos podataka
Ulazna jedinica	ima podvrstu	Tipkovnica
Tehnička podrška	ima podvrstu	Miš
Izlazna jedinica	ima podvrstu	Izlazna jedinica
Izlazna jedinica	ima podvrstu	Štampač
Izlazna jedinica	služi za	Monitor
Tehnička podrška	ima podvrstu	Prikazivanje podataka
Centralna jedinica	se sastoji od	Centralna jedinica
Računalo	se sastoji od	Računalo
Centralna procesorska jedinica	se sastoji od	Centralna procesorska jedinica
Upravljačka jedinica	upravlja s	Upravljačka jedinica
Instrukcija	se sastoji od	Instrukcija
Podatak	ima podvrstu	Podatak
		Informacija

Informacija	ima	Značenje
Instrukcija	se sastoji od	Operacija
Operacija	ima podvrstu	Aritmetička operacija
Aritmetička operacija	ima primjerak	Zbrajanje
Aritmetička operacija	ima primjerak	Oduzimanje
Operacija	ima podvrstu	Logička operacija
Logička operacija	ima primjerak	Negacija
Logička operacija	ima primjerak	Disjunkcija
Logička operacija	ima primjerak	Konjunkcija
Centralna procesorska jedinica	se sastoji od	Aritmetičko-Logička jedinica
Aritmetičko-Logička jedinica	izvršava	Aritmetička operacija
Aritmetičko-Logička jedinica	izvršava	Logička operacija
Računalo	se sastoji od	Radna memorija
Centralna jedinica	se sastoji od	Masovna memorija
Model računalnog sustava	se sastoji od	Izlazna jedinica
Model računalnog sustava	se sastoji od	Ulazna jedinica
Model računalnog sustava	se sastoji od	Centralna jedinica
Programski jezik	ima podvrstu	Programski jezik visoke razine
Programski jezik	ima podvrstu	Programski jezik niske razine
Programski jezik niske razine	ima primjerak	Asembler
Logički sklop	ima primjerak	NE sklop
NE sklop	izvršava	Negacija
Logički sklop	izvršava	Logička operacija
Logički sklop	ima primjerak	ILI sklop
ILI sklop	izvršava	Disjunkcija
Logički sklop	ima primjerak	NILI sklop
NILI sklop	izvršava	Negacija
NILI sklop	izvršava	Disjunkcija
Logički sklop	ima primjerak	I sklop
I sklop	izvršava	Konjunkcija
Logički sklop	ima primjerak	NI sklop
NI sklop	izvršava	Konjunkcija
NI sklop	izvršava	Negacija
Brojevni sustav	ima primjerak	Heksadekadski brojevni sustav
Heksadekadski brojevni sustav	slot	Znamenke
Znamenke	filler	0 1 2 3 4 5 6 7 8 9 A B C D E F
Znamenke	filler	0 1
Znamenke	filler	0 1 2 3 4 5 6 7
Znamenke	filler	0 1 2 3 4 5 6 7 8 9
Heksadekadski brojevni sustav	filler	0 1 2 3 4 5 6 7 8 9 A B C D E F
Heksadekadski brojevni sustav	filler	16
Heksadekadski brojevni sustav	slot	Baza
Baza	filler	16
Baza	filler	2
Baza	filler	8
Baza	filler	10
Brojevni sustav	ima primjerak	Binarni brojevni sustav
Binarni brojevni sustav	filler	2
Binarni brojevni sustav	filler	0 1
Binarni brojevni sustav	slot	Baza
Binarni brojevni sustav	slot	Znamenke
Brojevni sustav	ima primjerak	Dekadski brojevni sustav
Dekadski brojevni sustav	filler	10
Dekadski brojevni sustav	slot	Znamenke
Dekadski brojevni sustav	slot	Baza
Dekadski brojevni sustav	filler	0 1 2 3 4 5 6 7 8 9
Brojevni sustav	ima primjerak	Oktakni brojevni sustav
Oktakni brojevni sustav	filler	8
Oktakni brojevni sustav	slot	Baza
Oktakni brojevni sustav	slot	Znamenke
Oktakni brojevni sustav	filler	0 1 2 3 4 5 6 7

### 8.3 Semantičko određivanje reprezentacije analizom konceptata i relacija u područnom znanju

U okviru istraživanja vezanog za pronalaženje načina kako definirati reprezentaciju područnog znanja, proveli smo istraživanje na dvije razine o tome što bi bila reprezentacija područnog znanja „Računalo kao sustav“, odnosno koji koncepti i relacije upravo zbog svog značenja moraju pripadati reprezentaciji tog područnog znanja. Prva razina je obuhvatila 7 stručnjaka iz područnog znanja „Računalo kao sustav“, a druga razina je obuhvatila 6 studenata koji su slušali kolegij „Uvod u računarstvo“ (čija je jedna cjelina „Računalo kao sustav“).

Analiza reprezentacija odabranih na prvoj razini od strane stručnjaka je prikazana u Tablica 8.1:

Tablica 8.1. Rezultati semantičke analize stručnjaka

	Stručnjak 1	Stručnjak 2	Stručnjak 3	Stručnjak 4	Stručnjak 5	Stručnjak 6	Stručnjak 7	
Aplikacijska programska podrška	+			+		+		3
Aritmetička operacija						+		1
Aritmetičko-Logička jedinica	+	+	+	+	+	+	+	7
Asembler					+			1
Basic								0
Binarni brojevni sustav	+	+		+	+			4
Brojevni sustav	+		+		+			3
C								0
Centralna jedinica	+			+	+	+	+	5
Centralna procesorska jedinica	+	+	+	+	+	+	+	7
Dekadski brojevni sustav					+			1
Disjunkcija	+						+	2
Disketa								0
DOS								0
Fortran								0
Heksadekadski brojevni sustav					+			1
I sklop	+				+			2
ILI sklop					+		+	2
Informacija	+							1
Instrukcija			+		+			2
Interpretator			+					1
Izlazna jedinica	+			+		+		3
Jezični prevoditelji				+	+			2
Kompaktni disk								0
Kompilator	+				+			2
Konjunkcija	+							1
Logička operacija	+		+			+	+	4
Logički sklop	+		+		+			3
Masovna memorija				+	+	+		3
Memorija	+			+	+			3
Miš								0
Model računalnog sustava	+	+				+		3
Modem								0
Monitor								0
Mrežna kartica								0
NE sklop					+			1

Negacija								0
NI sklop					+			1
NILI sklop		+			+			2
Obrada podataka	+		+	+		+		4
Oduzimanje								0
Oktalni brojevni sustav								0
Operacija	+		+					2
Operacijski sustav	+		+	+	+			4
Paralelni prijenos podataka								0
Pascal								0
Podatak	+					+		2
Pohrana podataka				+		+		2
Prijenos podataka				+				1
Prikazivanje podataka				+		+		2
Programska podrška	+	+	+	+		+		5
Programski jezik	+		+	+	+			4
Programski jezik niske razine					+			1
Programski jezik visoke razine	+				+			2
Računalni sustav				+	+	+	+	4
Računalo				+		+	+	3
Radna memorija	+			+	+	+		4
RAM	+			+	+			3
ROM				+	+			2
Serijski prijenos podataka								0
Sistemska programska podrška	+	+		+	+	+		5
Štampač								0
Tehnička podrška	+		+	+		+	+	5
Temeljna funkcija računala	+			+		+		3
Tipkovnica								0
Tvrđi disk			+					1
Ulazna jedinica	+			+		+		3
Unix					+			1
Unos podataka				+		+		2
Upravljačka jedinica	+			+		+		3
Upravljanje podataka				+		+		2
Uređaj za komunikaciju								0
Uslužni programi								0
Windows	+				+			2
Zbrajanje								0
Značenje								0
	31	7	14	28	30	24	9	

Analiza reprezentacija odabranih na drugoj razini od strane studenata je prikazana u :

**Tablica 8.2. Rezultati semantičke analize od strane studenata**

	Student1	Student2	Student3	Student4	Student5	Student6	
Aplikacijska programska podrška	+		+	+	+	+	5
Aritmetička operacija							0
Aritmetičko-Logička jedinica	+	+		+	+	+	5
Asembler							0
Basic							0
Binarni brojevni sustav			+				1
Brojevni sustav			+				1
C							0
Centralna jedinica	+	+	+	+	+	+	6
Centralna procesorska jedinica	+	+	+	+	+	+	6

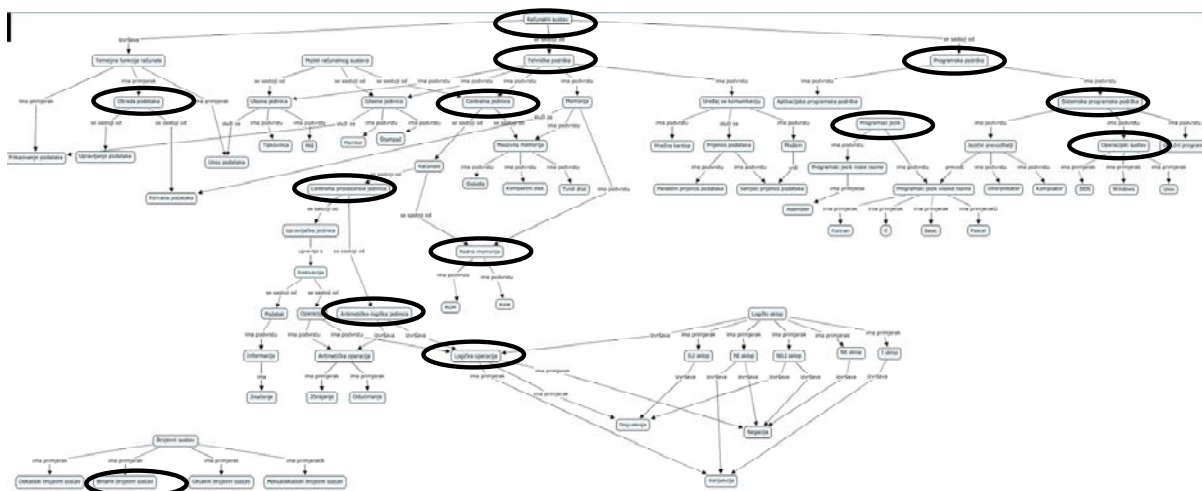
Dekadski brojevni sustav			+				1
Disjunkcija							0
Disketa							0
DOS						+	1
Fortran							0
Heksadekadski brojevni sustav			+				1
I sklop							0
II sklop							0
Informacija		+	+	+			3
Instrukcija							0
Interpretator							0
Izlazna jedinica	+	+	+	+	+	+	6
Jezični prevoditelji							0
Kompaktni disk							0
Kompilator							0
Konjunkcija							0
Logička operacija							0
Logički sklop							0
Masovna memorija	+	+	+		+		4
Memorija	+	+	+	+	+	+	6
Miš	+				+	+	3
Model računalnog sustava							0
Modem							0
Monitor	+				+	+	3
Mrežna kartica							0
NE sklop							0
Negacija							0
NI sklop							0
NII sklop							0
Obrada podataka	+	+	+		+	+	5
Oduzimanje				+			1
Oktalni brojevni sustav			+				1
Operacija							0
Operacijski sustav				+		+	2
Paralelni prijenos podataka							0
Pascal							0
Podatak		+					1
Pohrana podataka						+	1
Prijenos podataka							0
Prikazivanje podataka	+	+			+	+	4
Programska podrška	+		+	+	+	+	5
Programski jezik							0
Programski jezik niske razine							0
Programski jezik visoke razine							0
Računalni sustav	+	+		+	+	+	5
Računalo	+	+	+			+	4
Radna memorija	+	+	+	+	+	+	6
RAM				+			1
ROM				+			1
Serijski prijenos podataka							0
Sistemska programska podrška	+	+	+	+	+	+	6
Štampač	+				+	+	3
Tehnička podrška	+	+	+	+	+	+	6
Temeljna funkcija računala	+	+	+		+	+	5
Tipkovnica	+			+	+	+	4
Tvrđi disk							0
Ulazna jedinica	+	+	+	+	+	+	6
Unix							0
Unos podataka	+	+	+		+	+	5

Upravljačka jedinica	+	+		+	+	+	5
Upravljanje podataka							0
Uređaj za komunikaciju		+	+	+			3
Uslužni programi							0
Windows				+		+	2
Zbrajanje				+			1
Značenje							0
	23	20	22	22	22	26	

Kod prve razine reprezentacija uočavamo raznolikost semantičkog viđenja reprezentacije, odnosno nejedinstvenosti kako u veličinama reprezentacija (broj odabranih koncepata), tako i u sastavu tih reprezentacija (koji su koncepti odabrani). Stručnjaci su odabrali reprezentacije od 7 do 31 koncepta bez obzira na to da li su povezani ili nisu. Utvrdili smo da su se samo dva koncepta našla u svih 7 reprezentacija, četiri koncepta su se našla u njih 5, sedam koncepata u 4 reprezentacije. Dakle, na temelju semantičke analize od strane stručnjaka, ako bismo za prag uvrštavanja koncepta u reprezentaciju uzeli njegovo navođenje od strane 50% stručnjaka (ovaj prag se uzima heuristički), onda bi se reprezentacija sastojala od sljedećih 13 koncepata (Slika 8.1):

- Aritmetičko-Logička jedinica
- Binarni brojevni sustav
- Centralna jedinica
- Centralna procesorska jedinica
- Logička operacija
- Obrada podataka
- Operacijski sustav
- Programska podrška
- Programski jezik
- Računalni sustav
- Radna memorija
- Sistemska programska podrška
- Tehnička podrška



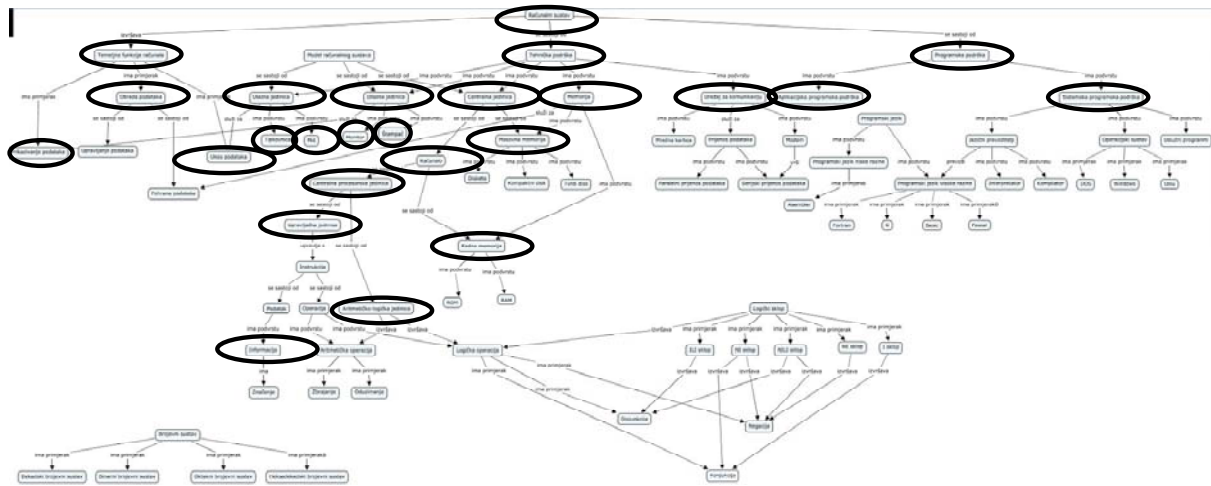


Slika 8.1. Reprezentacija za stručnjake uz prag prihvaćanja od 50%

Kod prve druge reprezentacija uočavamo veću jednobraznost u semantičkom viđenju reprezentacije, odnosno postoji manja nejedinstvenosti kako u veličinama reprezentacija (broj odabranih koncepata), tako i u sastavu tih reprezentacija (koji su koncepti odabrani). Studenti su odabrali reprezentacije od 20 do 26 koncepata uglavnom povezanih. Utvrdili smo da se osam koncepata našla u svih 6 reprezentacija, osam koncepata su se našla u njih 5, četiri koncepata u 4 reprezentacije. Dakle, na temelju semantičke analize od strane studenata, ako bismo za prag uvrštavanja koncepta u reprezentaciju uzeli njegovo navođenje od strane 50% studenata (ovaj prag se uzima heuristički), onda bi se reprezentacija sastojala od sljedećih 25 koncepata:

- Aplikacijska programska podrška
- Aritmetičko-Logička jedinica
- Centralna jedinica
- Centralna procesorska jedinica
- Informacija
- Izlazna jedinica
- Masovna memorija
- Memorija
- Miš
- Monitor
- Obrada podataka
- Prikazivanje podataka
- Programska podrška
- Računalni sustav
- Računalo
- Radna memorija
- Sistemska programska podrška
- Štampač
- Tehnička podrška

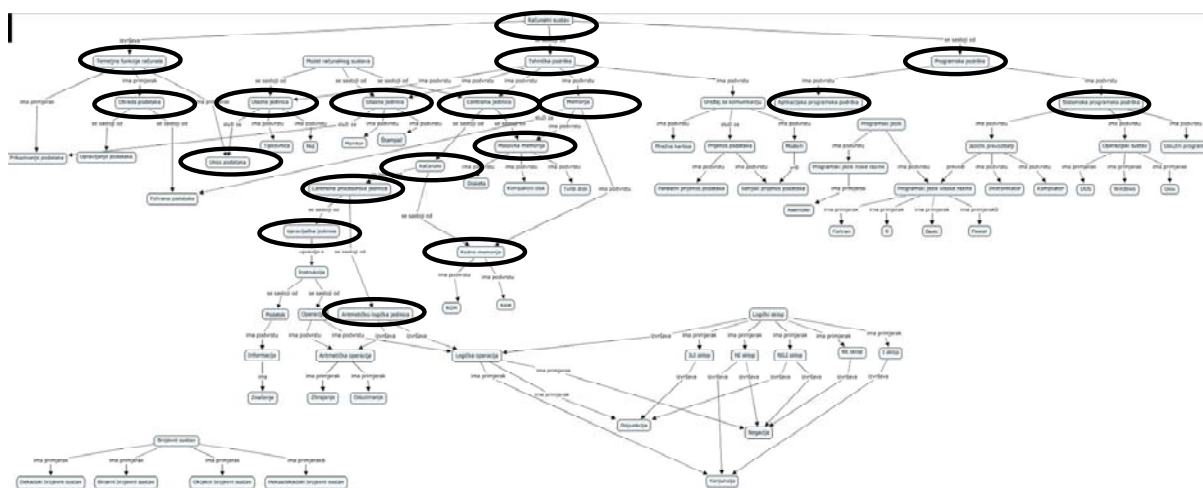
- Temeljna funkcija računala
- Tipkovnica
- Ulazna jedinica
- Unos podataka
- Upravljačka jedinica
- Uređaj za komunikaciju



Slika 8.2. Reprezentacija za studente uz prag prihvaćanja od 50%

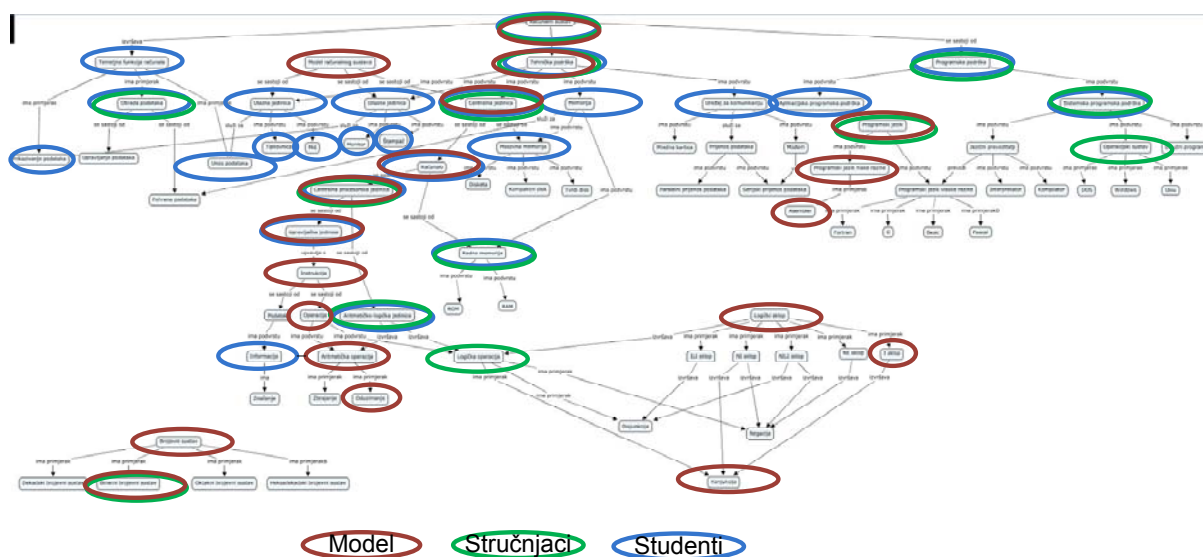
Ako bismo promatrali te dvije razine zajedno, uz prag od 50% za uvrštavanja koncepta u reprezentaciju, onda bi se reprezentacija sastojala od sljedećih 18 konceptata (Slika 8.3):

- Aplikacijska programska podrška
- Aritmetičko-Logička jedinica
- Centralna jedinica
- Centralna procesorska jedinica
- Izlazna jedinica
- Masovna memorija
- Memorija
- Obrada podataka
- Programska podrška
- Računalni sustav
- Računalo
- Radna memorija
- Sistemska programska podrška
- Tehnička podrška
- Temeljna funkcija računala
- Ulazna jedinica
- Unos podataka
- Upravljačka jedinica



Slika 8.3. Reprezentacija zajednička za stručnjake i studente uz prag prihvaćanja od 50%

Usporedbu reprezentacija koje smo definirali na temelju semantičke analize provedene od strane stručnjaka i studenata zasebno, te reprezentacije dobivene matematičkim putem prema modelu sustava opisanom u doktorskoj disertaciji možemo vidjeti na Slika 8.4:



Slika 8.4. Usporedba reprezentacija dobivenih semantičkom analizom i matematičkim pristupom

## 8.4 Oblikovanje i strukturiranje elemenata računalom oblikovanog nastavnog sadržaja

U oblikovanje i strukturiranje elemenata računalom oblikovanog nastavnog sadržaja prikazano konstruiranje samih elemenata računalom oblikovanog nastavnog sadržaja, zatim je prikazana struktura elemenata za svaki od stereotipova učenika prema znanju, te je dan primjer prezentiranja računalom oblikovanog nastavnog sadržaja u slučaju da učenik ne mijenja svoj stereotip tijekom procesa učenja i poučavanja.

### 8.4.1 Konstruiranje elemenata računalom oblikovanog nastavnog sadržaja u područnom znanju „Računalo kao sustav“

Cjelina  $C_1$  s centralnim vrhom **Računalni sustav** sadrži put duljine veće od 3, pa je ona nastavna cjelina, tj.  $NC_{Računalni\ sustav} = C_1, R_{NC_{Računalni\ sustav}} = (1,0,1)$

Cjelina  $C_2$  s centralnim vrhom **Model računalnog sustava** sadrži put duljine veće od 3, pa je ona nastavna cjelina, tj.  $NC_{Model\ računalnog\ sustava} = C_1, R_{NC_{Model\ računalnog\ sustava}} = (2,0,1)$

Cjelina  $C_3$  s centralnim vrhom **Logički sklop** ne sadrži put duljine veće od 3, ali sadrži put duljine veće od 1, pa je ona nastavna tema, tj.  $NT_{Logički\ sklop} = C_3, R_{NT_{Logički\ sklop}} = (3,0,2)$

Cjelina  $C_4$  s centralnim vrhom **Programski jezik** ne sadrži put duljine veće od 3, ali sadrži put duljine veće od 1, pa je ona nastavna tema, tj.  $NT_{Programski\ jezik} = C_3, R_{NT_{Programski\ jezik}} = (4,0,2)$

Cjelina  $C_5$  s centralnim vrhom **Brojevni sustav** ne sadrži put duljine veće od 3, niti put duljine veće od 1, pa je ona nastavna jedinica, tj.  $NJ_{Brojevni\ sustav} = C_3, R_{NT_{Brojevni\ sustav}} = (5,0,3)$

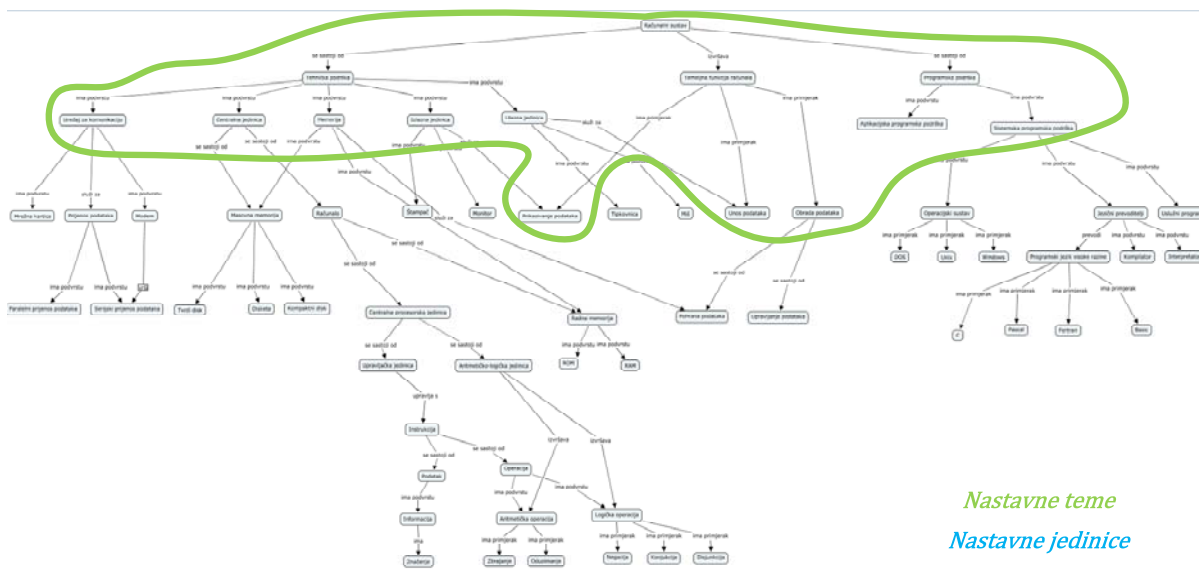
#### **Nastavna cjelina $NC_{Računalni\ sustav}$**

Definirajmo nastavne teme unutar nastavne cjeline  $NC_{Računalni\ sustav}$

Krećemo od centralnog vrha *Računalni sustav*

$$V_{NT_{Računalni\ sustav}} = \left\{ \begin{array}{l} Računalni\ sustav, Temeljna\ funkcija\ računala, Tehnička\ podrška, \\ Programska\ podrška, Prikazivanje\ podataka, Unos\ podataka, \\ Obrada\ podataka, Centralna\ jedinica, Izlazna\ jedinica, Memorija, \\ Ulazna\ jedinica, Uređaj\ za\ komunikaciju, \\ Aplikacijska\ programska\ podrška, Sistemska\ programska\ podrška \end{array} \right\},$$

$$R_{NT_{Računalni\ sustav}} = (1,0,2)$$



Promotrimo sada listove nastavne teme  $NT_{\text{Računalni sustav}}$  - Prikazivanje podataka, Unos podataka, Obrada podataka, Centralna jedinica, Izlazna jedinica, Memorija, Ulazna jedinica, Uređaj za komunikaciju, Aplikacijska programska podrška, Sistemska programska podrška

Za koncepte Prikazivanje podataka, Unos podataka i Aplikacijska programska podrška se ne može konstruirati niti nastavna tema niti nastavna jedinica.

Za koncepte Obrada podataka, Izlazna jedinica i Ulazna jedinica možemo konstruirati nastavne jedinice:

$$V_{NJ_{\text{Obrada podataka}}} = \{\text{Obrada podataka, Pohrana podataka, Upravljanje podataka}\},$$

$$R_{NJ_{\text{Obrada podataka}}} = (1,2,3)$$

$$V_{NJ_{\text{Izlazna jedinica}}} = \{\text{Izlazna jedinica, Monitor, Prikazivanje podataka, Štampač}\},$$

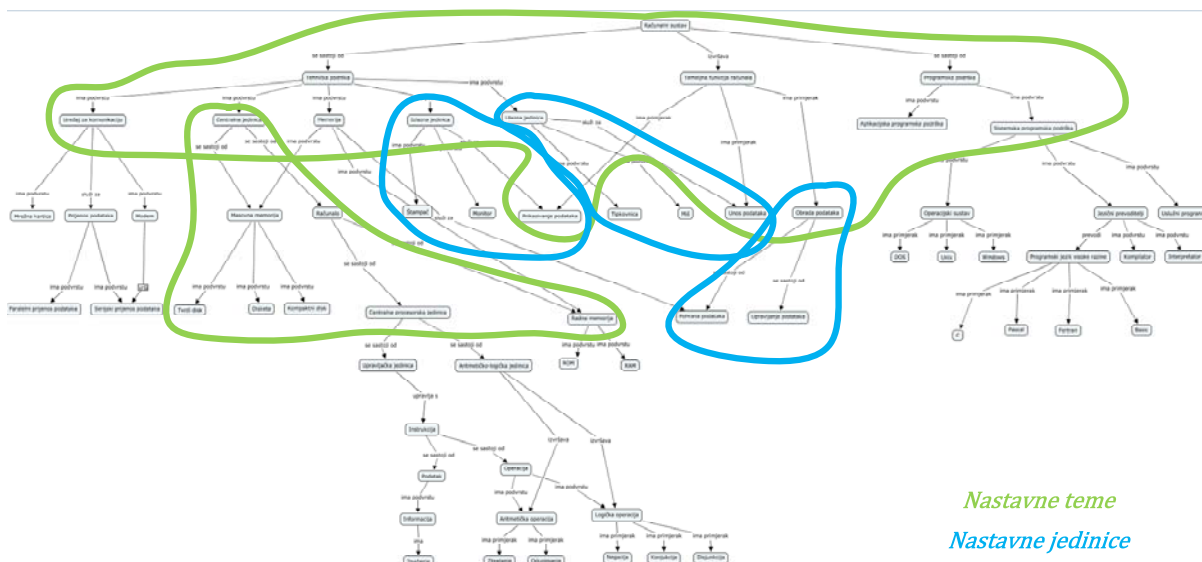
$$R_{NJ_{\text{Izlazna jedinica}}} = (1,2,3)$$

$$V_{NJ_{\text{Ulazna jedinica}}} = \{\text{Ulazna jedinica, Miš, Unos podataka, Tipkovnica}\}, R_{NJ_{\text{Ulazna jedinica}}} = (1,2,3)$$

Za koncept *Centralna jedinica* se može konstruirati nastavna tema jer postoji put u cjelini  $NC_{\text{Računalni sustav}}$  koji počinje u tom listu, a ima duljinu barem 2

$$V_{NT_{\text{Centralna jedinica}}} = \left\{ \begin{array}{l} \text{Centralna jedinica, Masovna memorija, Računalo, Tvrdi disk, Disketa,} \\ \text{Kompaktni disk, Radna memorija, Centralna procesorska jedinica} \end{array} \right\}$$

$$R_{NT_{\text{Centralna jedinica}}} = (1,2,2)$$



Nastavne teme  
Nastavne jedinice

Promotrimo sada listove nastavne teme  $NT_{\text{Centralna jedinica}}$  - Tvrdi disk, Disketa, Kompaktni disk, Radna memorija, Centralna procesorska jedinica

Za koncepte Tvrdi disk, Disketa, Kompaktni disk se ne može konstruirati niti nastavna tema niti nastavna jedinica.

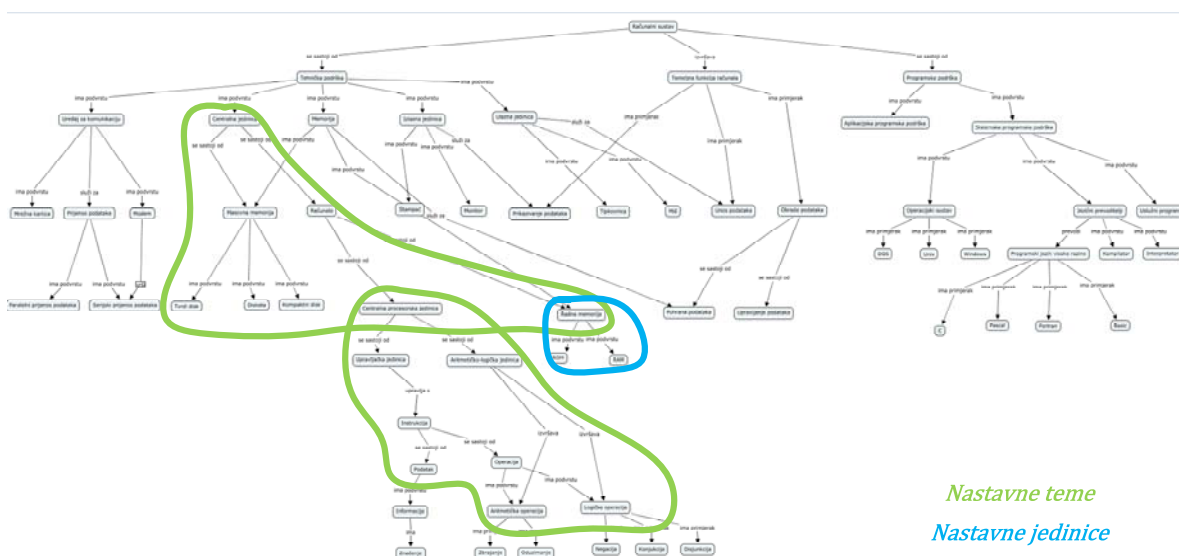
Za koncept Radna memorija možemo konstruirati nastavnu jedinicu:

$$V_{NJ_{\text{Radna memorija}}} = \{\text{Radna memorija, ROM, RAM}\}, R_{NJ_{\text{Radna memorija}}} = (1,4,3)$$

Za koncept Centralna procesorska jedinica možemo konstruirati nastavnu temu:

$$V_{NT_{\text{Centralna procesorska jedinica}}} = \left\{ \begin{array}{l} \text{Centralna procesorska jedinica, Aritmetičko – logička jedinica,} \\ \text{Upravljačka jedinica, Instrukcija, Podatak, Aritmetička operacija,} \\ \text{Logička operacija, Operacija} \end{array} \right\}$$

$$R_{NT_{\text{Centralna procesorska jedinica}}} = (1,4,2)$$



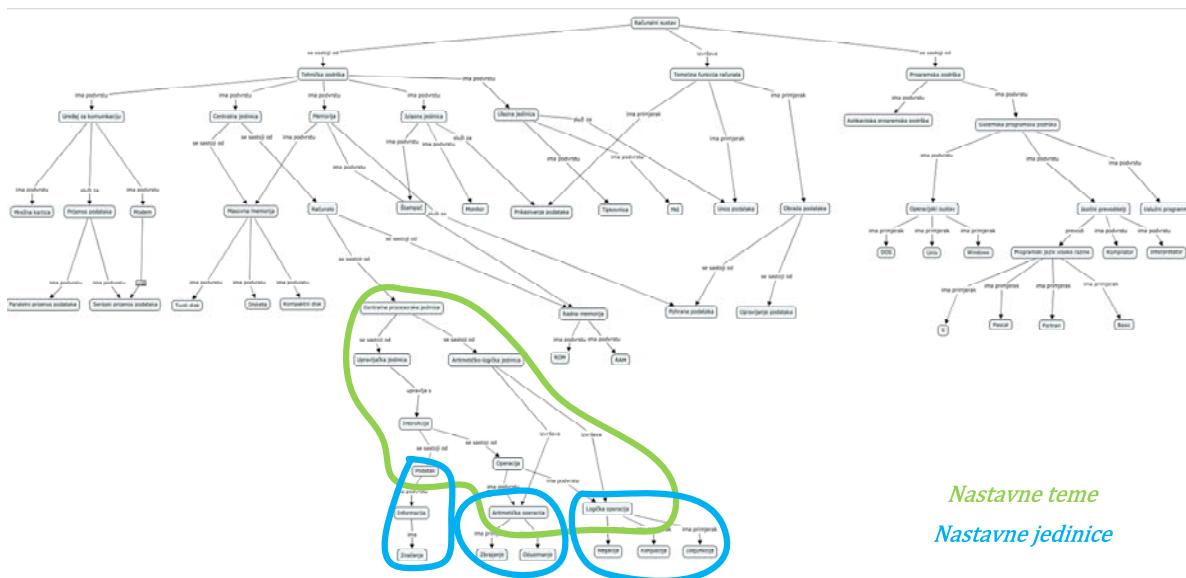
Nastavne teme  
Nastavne jedinice

Promotrimo sada listove nastavne teme  $NT_{Centralna\ procesorska\ jedinica}$  - Podatak, Aritmetička operacija, Logička operacija. Ne možemo konstruirati nastavne teme, pa konstruiramo nastavne jedinice.

$$V_{NJ_{Podatak}} = \{Podatak, Informacija, Značenje\}, R_{NJ_{Radna\ memorija}} = (1,7,3)$$

$$V_{NJ_{Aritmetička\ operacija}} = \{Aritmetička\ operacija, Zbrajanje, Oduzimanje\}, R_{NJ_{Aritmetička\ operacija}} = (1,5,3)$$

$$V_{NJ_{Logička\ operacija}} = \{Logička\ operacija, Negacija, Konjunkcija, Disjunkcija\}, R_{NJ_{Logička\ operacija}} = (1,8,3)$$

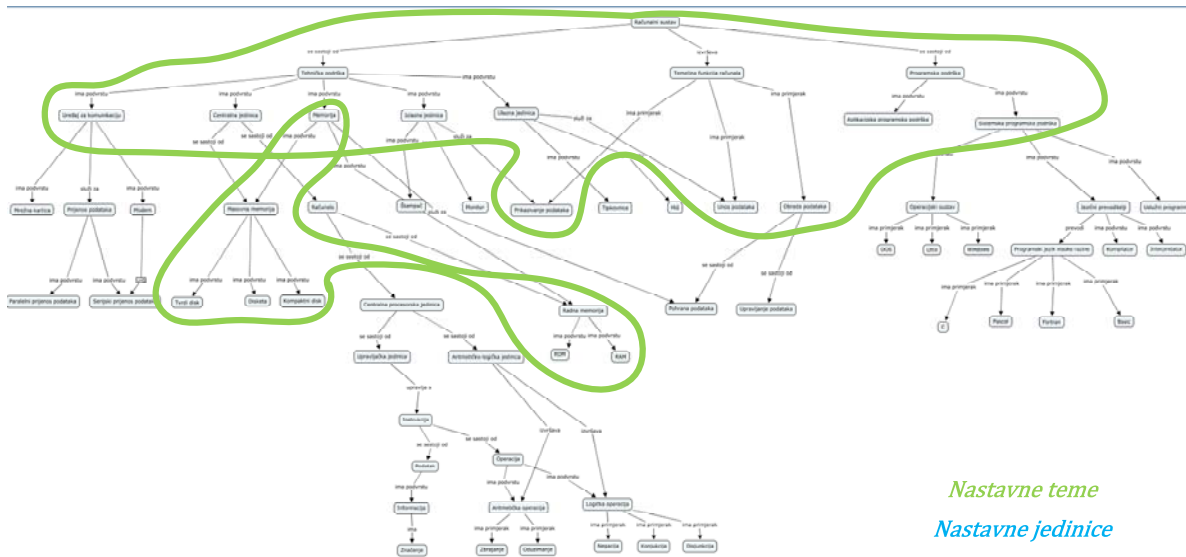


Za koncept *Memorija* se može konstruirati nastavna tema jer postoji put u cjelini  $NC_{Računski\ sustav}$  koji počinje u tom listu, a ima duljinu barem 2

$$V_{NT_{Memorija}} = \left\{ \begin{array}{l} Memorija, Masovna memorija, Radna memorija, Pohrana podataka, Tvrdi disk, \\ Disketa, Kompaktni disk, RAM, ROM \end{array} \right\}$$

$$R_{NT_{Memorija}} = (1,2,2)$$

Promotrimo sada listove nastavne teme  $NT_{Memorija}$  - Tvrdi disk, Disketa, Kompaktni disk, ROM, RAM, Pohrana podataka. Za nijedan od njih se ne može konstruirati niti nastavna tema niti nastavna jedinica.



Za koncept *Uređaj za komunikaciju* se može konstruirati nastavna tema jer postoji put u cjelini  $NC_{Računalni\ sustav}$  koji počinje u tom listu, a ima duljinu barem 2

$$V_{NT}^{Uređaj\ za\ komunikaciju} = \left\{ \begin{array}{l} Uređaj\ za\ komunikaciju, Mrežna\ kartica, Modem, Prijenos\ podataka, \\ Paralelni\ prijenos\ podataka, Serijski\ prijenos\ podataka \end{array} \right\}$$

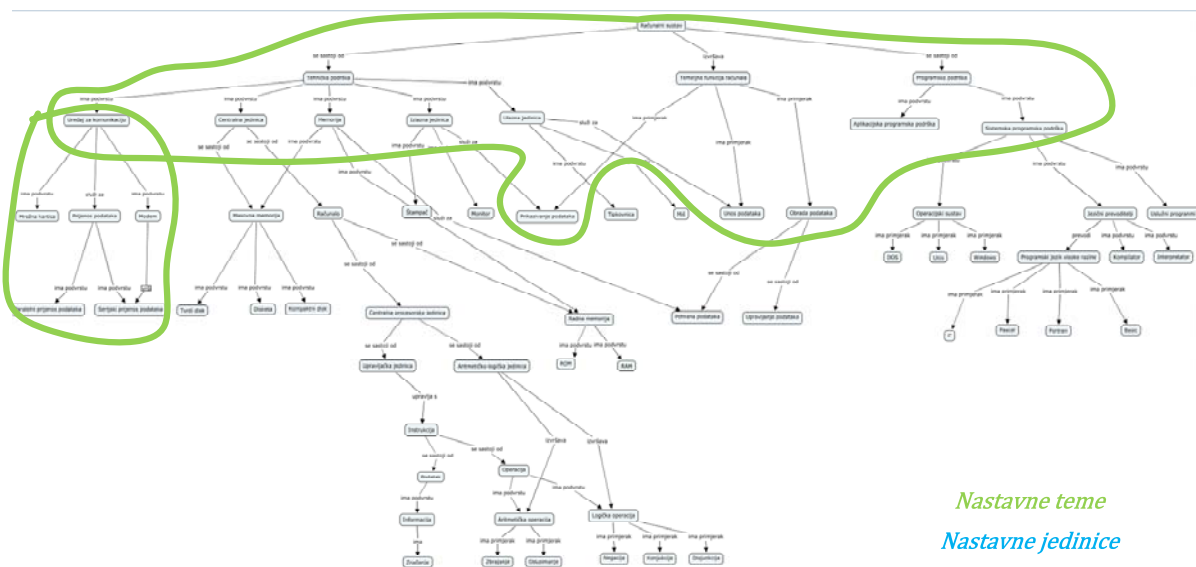
$$R_{NT}^{Uređaj\ za\ komunikaciju} = (1,2,2)$$

Promotrimo sada listove nastavne teme  $NC_{Uređaj\ za\ komunikaciju}$  - Mrežna kartica, Paralelni prijenos podataka i Serijski prijenos podataka. Za nijedan od njih se ne može konstruirati niti nastavna tema niti nastavna jedinica.

Za koncept *Sistemska programska podrška* se može konstruirati nastavna tema jer postoji put u cjelini  $NC_{Računalni\ sustav}$  koji počinje u tom listu, a ima duljinu barem 2

$$V_{NT}^{Sistemska\ programska\ podrška} = \left\{ \begin{array}{l} Sistemska\ programska\ podrška, Operacijski\ sustav, \\ Jezični\ prevoditelji, Uslužni\ programi, DOS, Unix, Windows, \\ Programski\ jezik\ visoke\ razine, Komilator, Intepretator \end{array} \right\}$$

$$R_{NT}^{Sistemska\ programska\ podrška} = (1,2,2)$$



Promotrimo sada listove nastavne teme  $NT_{Sistemska\ programska\ podrška}$  - DOS, Unix, Windows, Programski jezik visoke razine, Komilator, Intepretator.

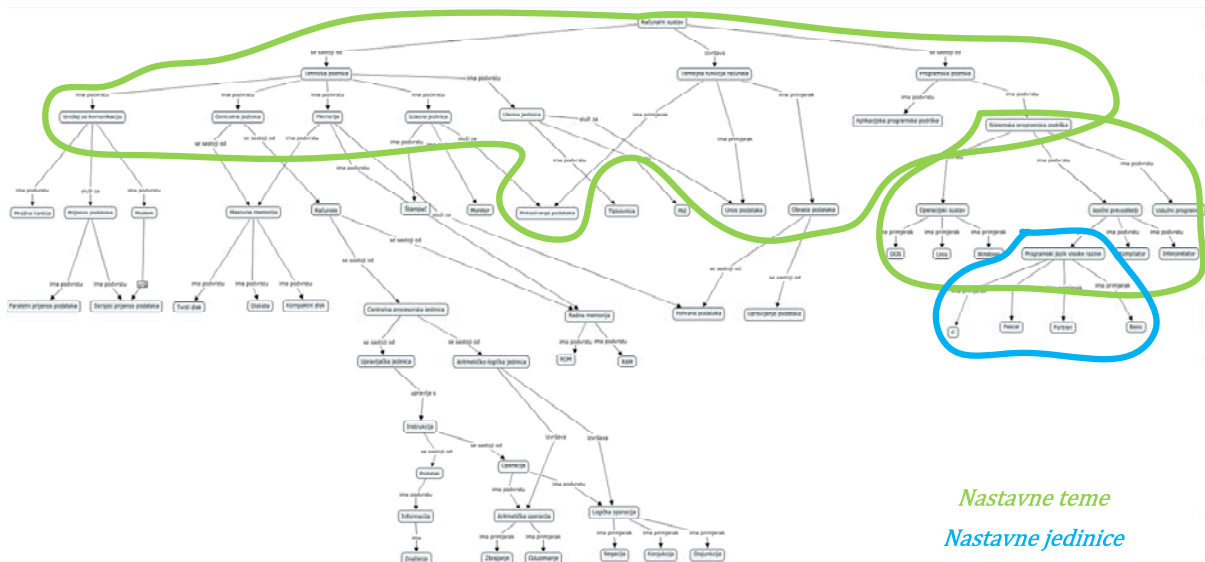
Za koncepte DOS, Unix, Windows, Komilator, Intepretator se ne može konstruirati niti nastavna tema niti nastavna jedinica.

Za koncept Programski jezik visoke razine možemo konstruirati nastavnu jedinicu:

$$V_{NJ}^{Programski\ jezik\ visoke\ razine} = \{Programski\ jezik\ visoke\ razine, C, Pascal, Fortran, BASIC\},$$

$$R_{NJ}^{Programski\ jezik\ visoke\ razine} = (4,1,3)$$





Nastavne teme  
Nastavne jedinice

**Nastavna cjelina  $NC_{Model\ računalnog\ sustava}$**

Analogno se definiraju sljedeće nastavne teme nastavne cjeline  $NC_{Model\ računalnog\ sustava}$  :

$$V_{NT_{Model\ računalnog\ ustava}} = \left\{ \begin{array}{l} Model\ računalnog\ ustava, Ulazna\ jedinica, Izlazna\ jedinica, \\ Centralna\ jedinica, Tipkovnica, Unos\ podataka, Miš, Štampač, \\ Monitor, Prikazivanje\ podataka, Računalo, Masovna\ memorija \end{array} \right\}$$

$$R_{NT_{Model\ računalnog\ ustava}} = (2,0,2)$$

$$V_{NT_{Računalo}} = \left\{ \begin{array}{l} Računalo, Radna\ memorija, Centralna\ procesorska\ jedinica, RAM, ROM, \\ Upravljačka\ jedinica, Aritmetičko – logička\ jedinica \end{array} \right\}$$

$$R_{NT_{Računalo}} = (2,2,2)$$

$$V_{NT_{Upravljačka\ jedinica}} = \{Upravljačka\ jedinica, Instrukcija, Podatak, Operacija\},$$

$$R_{NT_{Upravljačka\ jedinica}} = (2,4,2)$$

$$V_{NT_{Operacija}} = \left\{ \begin{array}{l} Operacija, Logička\ operacija, Aritmetička\ operacija, Konjunktija, Negacija, \\ Disjunktija, Zbrajanje, Oduzimanje \end{array} \right\}$$

$$R_{NT_{Operacija}} = (2,6,2)$$

Nastavna jedinica  $NJ_{Podatak}$  je već konstruirana u okviru nastavne cjeline  $NC_{Računalni\ sustav}$  pa konstruiramo novu koja ima drugi rang (2,6,3).

$$V_{NT_{Aritmetičko-logička\ jedinica}} = \left\{ \begin{array}{l} Aritmetičko – logička\ jedinica, Logička\ operacija, Aritmetička \\ operacija, Konjunktija, Negacija, Disjunktija, Zbrajanje, Oduzimanje \end{array} \right\}$$

$$R_{NT_{Aritmetičko-logička\ jedinica}} = (2,4,2)$$

$$V_{NJ_{Masovna\ memorija}} = \{Masovna\ memorija, Disketa, Tvrdi\ disk, Kompaktni\ disk\},$$

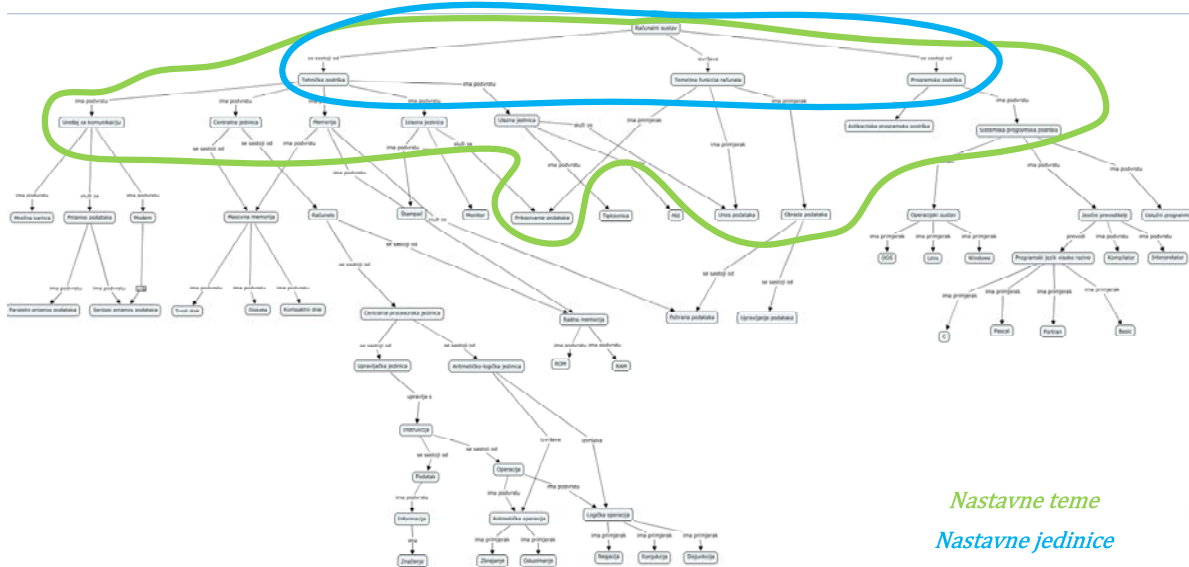
$$R_{NJ_{Masovna\ memorija}} = (2,2,3)$$

## Nastavna tema $NT_{Računalni\ sustav}$

Definirajmo nastavne jedinice unutar nastavne teme  $NT_{Računalni\ sustav}$

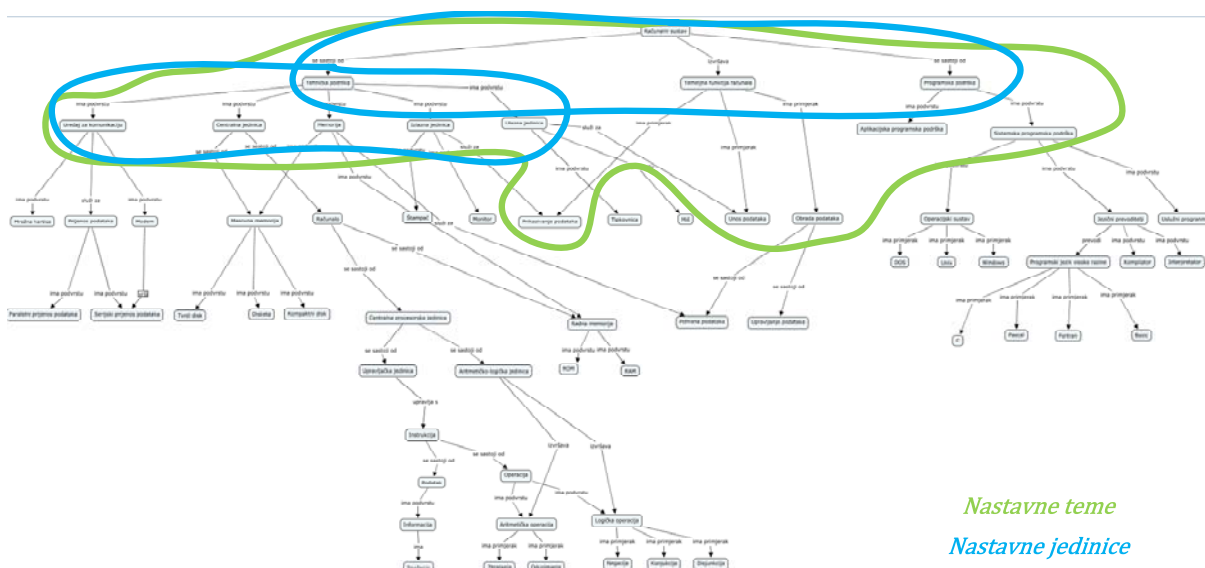
Krećemo od centralnog vrha  $Računalni\ sustav$

$$V_{NJ_{Računalni\ sustav}} = \left\{ \begin{array}{l} Računalni\ sustav, Temeljna\ funkcija\ računala, \\ Tehnička\ podrška, Programska\ podrška \end{array} \right\}, R_{NJ_{Računalni\ sustav}} = (1,0,3)$$



Promotrimo sada listove nastavne jedinice  $NJ_{Računalni\ sustav}$  - Tehnička podrška, Temeljna funkcija računala, Programska podrška. Za svih se može konstruirati nastavna jedinica jer postoji put u temi  $NT_{Računalni\ sustav}$  koji počinje u tom listu, a ima duljinu barem 1.

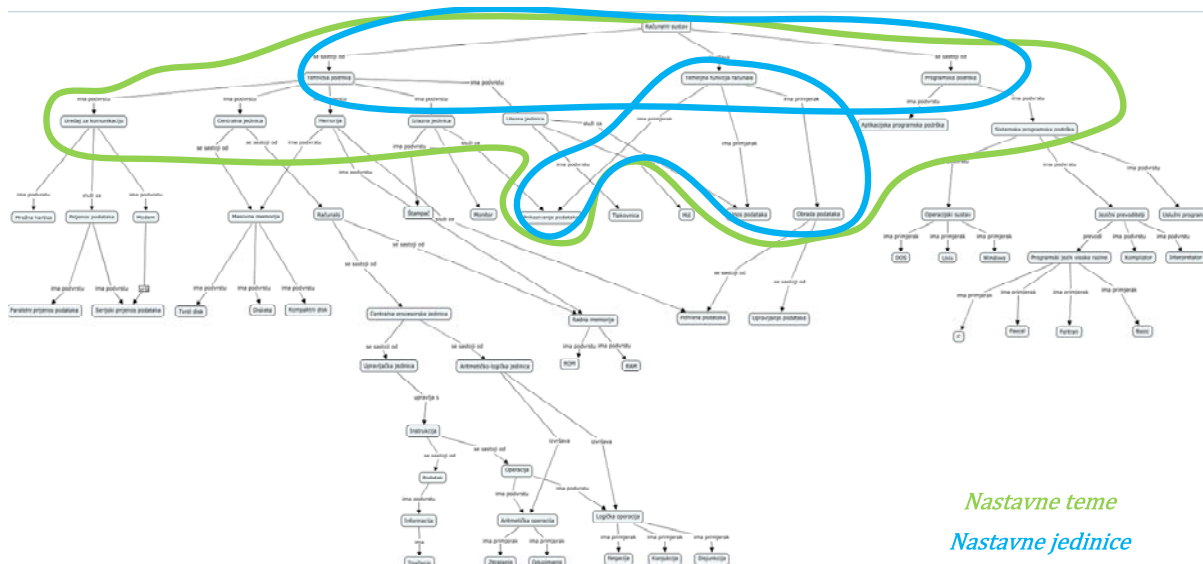
$$V_{NJ_{Tehnička\ podrška}} = \left\{ \begin{array}{l} Tehnička\ podrška, Centralna\ jedinica, \\ Memorija, Izlazna\ jedinica, Ulazna\ jedinica, \\ Uređaj\ za\ komunikaciju \end{array} \right\}, R_{NJ_{Tehnička\ podrška}} = (1,1,3)$$



Promotrimo sada listove nastavne jedinice  $NJ_{Tehnička\ podrška}$ . Za nijednog se ne može konstruirati nastavna jedinica jer ne postoji put u temi  $NT_{Računalni\ sustav}$  koji počinje u tom listu, a ima duljinu barem 1.

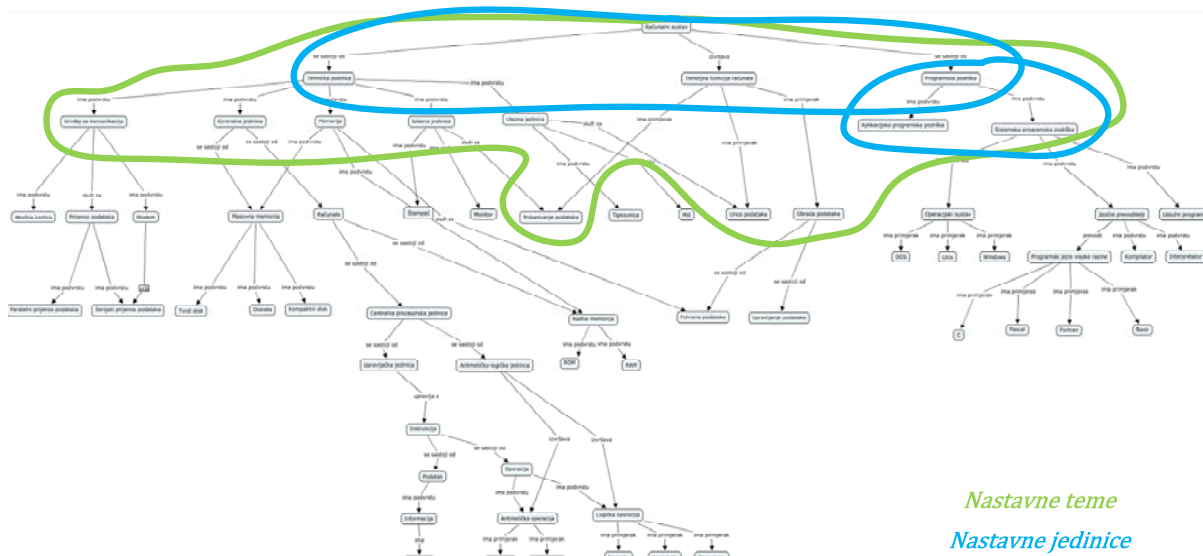
$$V_{NJ_{Tehnička\ podrška}} = \{ \text{Temeljna funkcija računala, Unos podataka, Prikazivanje podataka, Obrada podataka} \}$$

$$R_{NJ_{Tehnička\ podrška}} = (1,1,3)$$



Promotrimo sada listove nastavne jedinice  $NJ_{Temeljna\ funkcija\ računala}$ . Za nijednog se ne može konstruirati nastavna jedinica jer ne postoji put u temi  $NT_{Računalni\ sustav}$  koji počinje u tom listu, a ima duljinu barem 1.

$$V_{NJ_{Programska\ podrška}} = \{ \text{Programska podrška, Aplikacijska programska podrška, Sistemska programska podrška} \}, R_{NJ_{Programska\ podrška}} = (1,1,3)$$



Promotrimo sada listove nastavne jedinice  $NJ_{Programska\ podrška}$ . Za nijednog se ne može konstruirati nastavna jedinica jer ne postoji put u temi  $NT_{Računalni\ sustav}$  koji počinje u tom listu, a ima duljinu barem 1.

**Nastavne teme**  $NT_{\text{Logički sklop}}$ ,  $NT_{\text{Programski jezik}}$ ,  $NT_{\text{Sistemska programska podrška}}$ ,  
 $NT_{\text{Memorija}}$  i  $NT_{\text{Uređaj za komunikaciju}}$

Analogno se definiraju ostale nastavne jedinice za nastavne teme  $NT_{\text{Logički sklop}}$ ,  $NT_{\text{Programski jezik}}$ ,  $NT_{\text{Sistemska programska podrška}}$ ,  $NT_{\text{Memorija}}$  i  $NT_{\text{Uređaj za komunikaciju}}$ :

$$V_{NJ_{\text{Logički sklop}}} = \{\text{Logički sklop, Logička operacija, ILI sklop, I sklop, NE sklop, NILI sklop, NI sklop}\},$$

$$R_{NJ_{\text{Logički sklop}}} = (3,0,3)$$

Nastavna jedinica  $NJ_{\text{Logička operacija}}$  je već konstruirana u okviru nastavne cjeline  $NC_{\text{Računalni sustav}}$  pa konstruiramo novu koja ima drugi rang (2,5,3).

$$V_{NJ_{\text{ILI sklop}}} = \{\text{ILI sklop, Disjunkcija}\}, R_{NJ_{\text{ILI sklop}}} = (3,1,3)$$

$$V_{NJ_{\text{I sklop}}} = \{\text{I sklop, Konjukcija}\}, R_{NJ_{\text{I sklop}}} = (3,1,3)$$

$$V_{NJ_{\text{NILI sklop}}} = \{\text{NILI sklop, Disjunkcija, Negacija}\}, R_{NJ_{\text{NILI sklop}}} = (3,1,3)$$

$$V_{NJ_{\text{NI sklop}}} = \{\text{NI sklop, Konjukcija, Negacija}\}, R_{NJ_{\text{NI sklop}}} = (3,1,3)$$

$$V_{NJ_{\text{NE sklop}}} = \{\text{NE sklop, Negacija}\}, R_{NJ_{\text{NE sklop}}} = (3,1,3)$$

$$V_{NJ_{\text{Programski jezik}}} = \left\{ \begin{array}{l} \text{Programski jezik, Programski jezik niske razine,} \\ \text{Programski jezik visoke razine} \end{array} \right\}, R_{NJ_{\text{Programski jezik}}} = (4,0,3)$$

$$V_{NJ_{\text{Programski jezik niske razine}}} = \{\text{Programski jezik niske razine, Asembler}\}, R_{NJ_{\text{Programski jezik niske razine}}} = (4,1,3)$$

$$V_{NJ_{\text{Programski jezik visoke razine}}} = \{\text{Programski jezik visoke razine, C, Pascal, Fortran, BASIC}\},$$

$$R_{NJ_{\text{Programski jezik visoke razine}}} = (4,1,3)$$

$$V_{NJ_{\text{Sistemska programska podrška}}} = \left\{ \begin{array}{l} \text{Sistemska programska podrška, Operacijski sustav,} \\ \text{Jezični prevoditelji, Uslužni programi} \end{array} \right\},$$

$$R_{NJ_{\text{Sistemska programska podrška}}} = (1,2,3)$$

$$V_{NJ_{\text{Jezični prevoditelji}}} = \left\{ \begin{array}{l} \text{Jezični prevoditelji, Programski jezik visoke razine,} \\ \text{Interpreter, Kompilator} \end{array} \right\}, R_{NJ_{\text{Jezični prevoditelji}}} = (1,3,3)$$

$$V_{NJ_{\text{Operacijski sustav}}} = \left\{ \begin{array}{l} \text{Operacijski sustav, DOS,} \\ \text{Windows, Unix} \end{array} \right\}, R_{NJ_{\text{Operacijski sustav}}} = (1,3,3)$$

$$V_{NJ_{\text{Memorija}}} = \left\{ \begin{array}{l} \text{Memorija, Masovna memorija,} \\ \text{Radna memorija, Pohrana podataka} \end{array} \right\}, R_{NJ_{\text{Memorija}}} = (1,2,3)$$

$$V_{NJ_{\text{Uređaj za komunikaciju}}} = \left\{ \begin{array}{l} \text{Uređaj za komunikaciju, Mrežna kartica,} \\ \text{Modem, Prijenos podataka} \end{array} \right\}, R_{NJ_{\text{Uređaj za komunikaciju}}} = (1,2,3)$$

$$V_{NJ_{\text{Prijenos podataka}}} = \left\{ \begin{array}{l} \text{Prijenos podataka, Paralelni prijenos podataka,} \\ \text{Serijski prijenos podataka} \end{array} \right\}, R_{NJ_{\text{Prijenos podataka}}} = (1,3,3)$$

$$V_{NJModem} = \left\{ \begin{array}{l} \text{Modem,} \\ \text{Serijski prijenos podataka} \end{array} \right\}, R_{NJModem} = (1,3,3)$$

**Nastavne teme**  $NT_{Model\ računalnog\ ustava}$  ,  $NT_{Računalo}$  ,  $NT_{Upravljačka\ jedinica}$  ,  $NT_{Operacija}$  ,  $NT_{Aritmetičko-logička\ jedinica}$

Analogno se definiraju ostale nastavne jedinice za nastavne teme  $NT_{Model\ računalnog\ ustava}$  ,  $NT_{Računalo}$  ,  $NT_{Upravljačka\ jedinica}$  ,  $NT_{Operacija}$  ,  $NT_{Aritmetičko-logička\ jedinica}$  :

$$V_{NJModel\ računalnog\ ustava} = \left\{ \begin{array}{l} \text{Model računalnog ustava, Ulazna jedinica,} \\ \text{Izlazna jedinica, Centralna jedinica} \end{array} \right\}, R_{NJModel\ računalnog\ ustava} = (2,0,3)$$

$$V_{NJCentralna\ jedinica} = \{ \text{Centralna jedinica, Masovna memorija, Računalo} \}, R_{NJCentralna\ jedinica} = (1,2,3)$$

Nastavne jedinice  $NJ_{Ulazna\ jedinica}$  i  $NJ_{Izlazna\ jedinica}$  su već konstruirane u okviru nastavne cjeline  $NC_{Računalni\ sustav}$  pa konstruiramo nove koje imaju redom drugi rang (2,1,3) i (2,1,3).

$$V_{NJRačunalo} = \left\{ \begin{array}{l} \text{Računalo, Radna memorija,} \\ \text{Centralna procesorska jedinica} \end{array} \right\}, R_{NJRačunalo} = (2,2,3)$$

Nastavna jedinica  $NJ_{Radna\ memorija}$  je već konstruirana u okviru nastavne cjeline  $NC_{Računalni\ sustav}$  pa konstruiramo novu koja ima drugi rang (2,3,3).

$$V_{NJCentralna\ procesorska\ jedinica} = \left\{ \begin{array}{l} \text{Centralna procesorska jedinica, Upravljačka} \\ \text{jedinica, Aritmetičko – logička jedinica} \end{array} \right\}, R_{NJCentralna\ procesorska\ jedinica} = (1,4,3)$$

$$V_{NJUpravljačka\ jedinica} = \left\{ \begin{array}{l} \text{Upravljačka jedinica, Instrukcija,} \\ \text{Podatak, Operacija} \end{array} \right\}, R_{NJUpravljačka\ jedinica} = (2,4,3)$$

$$V_{NJ\text{Aritmetičko-logička}\ jedinica} = \left\{ \begin{array}{l} \text{Aritmetičko – logička jedinica, Logička} \\ \text{operacija, Aritmetička operacija} \end{array} \right\}, R_{NJ\text{Aritmetičko-logička}\ jedinica} = (2,4,3)$$

$$V_{NJOperacija} = \{ \text{Operacija, Logička operacija, Aritmetička operacija} \}, R_{NJOperacija} = (2,6,3)$$

Nastavne jedinice  $NJ_{Podatak}$  ,  $NJ_{Logička\ operacija}$  i  $NJ_{Aritmetička\ operacija}$  su već konstruirane u okviru nastavnih cjelina  $NC_{Računalni\ sustav}$  i  $NC_{Model\ računalnog\ sustava}$  pa nemamo više nijednu cjelinu unutar koje bismo ih mogli konstruirati.

## 8.4.2 Nastavne cjeline, teme i jedinice u područnom znanju „Računalo kao sustav“

Na područnom znanju „Računalo kao sustav“ imamo 5 cjelina (označavamo ih sa 1 – Računalni sustav, 2 – Model računalnog sustava, 3 – Logički sklop, 4 – Programski jezik, 5 – Brojevni sustav). Cjeline su podijeljene u:

- 2 nastavne cjeline (Računalni sustav, Model računalnog sustava),
- 13 nastavnih tema:
  - u nastavnoj cjelini Računalni sustav imamo 7 nastavnih tema:
    - Računalni sustav, Centralna jedinica, Memorija, Sistemska programska podrška, Uređaj za komunikaciju, Centralna procesorska jedinica, Operacija,
  - u nastavnoj cjelini Model računalnog sustava imamo 5 nastavnih tema:
    - Model računalnog sustava, Računalo, Aritmetičko-Logička jedinica, Upravljačka jedinica, Operacija,
  - cjelina 3 je sama za sebe nastavna tema Logički sklop,
  - cjelina 4 je sama za sebe nastavna tema Programski jezik,
- 43 nastavne jedinice:
  - u nastavnoj cjelini Računalni sustav imamo 26 nastavnih jedinica:
    - Računalni sustav, Programska podrška, Tehnička podrška, Temeljna funkcija računala, Centralna jedinica, Izlazna jedinica, Memorija, Obrada podataka, Sistemska programska podrška, Ulazna jedinica, Uređaj za komunikaciju, Jezični prevoditelji, Masovna memorija, Modem, Operacijski sustav, Prijenos podataka, Računalo, Radna memorija, Centralna procesorska jedinica, Programski jezik visoke razine, Aritmetičko-Logička jedinica, Upravljačka jedinica, Aritmetička operacija, Logička operacija, Operacija, Podatak,
  - u nastavnoj cjelini Model računalnog sustava imamo 14 nastavnih jedinica:
    - Model računalnog sustava, Centralna jedinica, Izlazna jedinica, Ulazna jedinica, Masovna memorija, Računalo, Centralna procesorska jedinica, Radna memorija, Aritmetičko-Logička jedinica, Upravljačka jedinica, Logička operacija, Operacija, Podatak
  - u nastavnoj temi Logički sklop imamo 6 nastavnih jedinica:
    - Logički sklop, I sklop, ILI sklop, NE sklop, NI sklop, NILI sklop, Logička operacija
  - u nastavnoj temi Programski jezik imamo 3 nastavne jedinice:
    - Programski jezik, Programski jezik niske razine, Programski jezik visoke razine
  - cjelina 5 je sama za sebe nastavna jedinica Brojevni sustav

Sve elemente računalom oblikovanog nastavnog sadržaja možemo vidjeti u Tablica 8.3. (sortirano prema oznaci cjeline, zatim prema vrsti, te na kraju prema udaljenosti od korijena cjeline), te u Tablica 8.4. (sortirano prema korijenu elementa računalom oblikovanog nastavnog sadržaja).

Tablica 8.3. Elementi računalom oblikovanog nastavnog sadržaja "Računalo kao sustav"

Rbr	Korijen elementa	Oznaka cjeline	Vrsta elementa	Udaljenost od korijena određene cjeline	Broj koncepata	Popis koncepata koji se nalaze u elementu
1	Brojevni sustav	Brojevni sustav	Nastavna jedinica	0	5	Binarni brojevni sustav Brojevni sustav Dekadski brojevni sustav Heksadekadski brojevni sustav Oktakni brojevni sustav
2	I sklop	Logički sklop	Nastavna jedinica	1	2	I sklop Konjunkcija
3	ILI sklop	Logički sklop	Nastavna jedinica	1	2	Disjunkcija ILI sklop
4	Logička operacija	Logički sklop	Nastavna jedinica	1	4	Disjunkcija Konjunkcija Logička operacija Negacija
5	Logički sklop	Logički sklop	Nastavna jedinica	0	7	I sklop ILI sklop Logička operacija Logički sklop NE sklop NI sklop NILI sklop
6	NE sklop	Logički sklop	Nastavna jedinica	1	2	NE sklop Negacija
7	NI sklop	Logički sklop	Nastavna jedinica	1	3	Konjunkcija Negacija NI sklop
8	NILI sklop	Logički sklop	Nastavna jedinica	1	3	Disjunkcija Negacija NILI sklop
9	Logički sklop	Logički sklop	Nastavna tema	0	10	Disjunkcija I sklop ILI sklop Konjunkcija Logička operacija Logički sklop NE sklop Negacija NI sklop NILI sklop
10	Model računalnog sustava	Model računalnog sustava	Nastavna cjelina	0	33	Aritmetička operacija Aritmetičko-Logička jedinica Centralna jedinica Centralna procesorska jedinica Disjunkcija Disketa Informacija Instrukcija Izlazna jedinica Kompaktni disk Konjunkcija Logička operacija Masovna memorija Miš Model računalnog sustava Monitor Negacija Oduzimanje Operacija Podatak Prikazivanje podataka Računalo Radna memorija RAM ROM Štampač

						Tipkovnica Tvrđi disk Ulazna jedinica Unos podataka Upravljačka jedinica Zbrajanje Značenje
11	Aritmetička operacija	Model računalnog sustava	Nastavna jedinica	5	3	Aritmetička operacija oc319069238 brajanje
12	Aritmetičko-Logička jedinica	Model računalnog sustava	Nastavna jedinica	4	3	Aritmetička operacija Aritmetičko-Logička jedinica Logička operacija
13	Centralna jedinica	Model računalnog sustava	Nastavna jedinica	1	3	Centralna jedinica Masovna memorija Računalo
14	Centralna procesorska jedinica	Model računalnog sustava	Nastavna jedinica	3	3	Aritmetičko-Logička jedinica Centralna procesorska jedinica Upravljačka jedinica
15	Izlazna jedinica	Model računalnog sustava	Nastavna jedinica	1	4	Izlazna jedinica Monitor Prikazivanje podataka Štampač
16	Logička operacija	Model računalnog sustava	Nastavna jedinica	5	4	Disjunkcija Konjunkcija Logička operacija Negacija
17	Masovna memorija	Model računalnog sustava	Nastavna jedinica	2	4	Disketa Kompaktni disk Masovna memorija Tvrđi disk
18	Model računalnog sustava	Model računalnog sustava	Nastavna jedinica	0	4	Centralna jedinica Izlazna jedinica Model računalnog sustava Ulazna jedinica
19	Operacija	Model računalnog sustava	Nastavna jedinica	6	3	Aritmetička operacija Logička operacija Operacija
20	Podatak	Model računalnog sustava	Nastavna jedinica	6	3	Informacija Podatak Značenje
21	Računalo	Model računalnog sustava	Nastavna jedinica	2	3	Centralna procesorska jedinica Računalo Radna memorija
22	Radna memorija	Model računalnog sustava	Nastavna jedinica	3	3	Radna memorija RAM ROM
23	Ulazna jedinica	Model računalnog sustava	Nastavna jedinica	1	4	Miš Tipkovnica Ulazna jedinica Unos podataka
24	Upravljačka jedinica	Model računalnog sustava	Nastavna jedinica	4	4	Instrukcija Operacija Podatak Upravljačka jedinica
25	Aritmetičko-Logička jedinica	Model računalnog sustava	Nastavna tema	4	8	Aritmetička operacija Aritmetičko-Logička jedinica Disjunkcija Konjunkcija Logička operacija Negacija Oduzimanje Zbrajanje
26	Model računalnog sustava	Model računalnog sustava	Nastavna tema	0	12	Centralna jedinica Izlazna jedinica Masovna memorija Miš Model računalnog sustava Monitor



						Prikazivanje podataka Računalo Štampač Tipkovnica Ulazna jedinica Unos podataka
27	Operacija	Model računalnog sustava	Nastavna tema	6	8	Aritmetička operacija Disjunkcija Konjunkcija Logička operacija Negacija Oduzimanje Operacija Zbrajanje
28	Računalo	Model računalnog sustava	Nastavna tema	2	7	Aritmetičko-Logička jedinica Centralna procesorska jedinica Računalo Radna memorija RAM ROM Upravljačka jedinica
29	Upravljačka jedinica	Model računalnog sustava	Nastavna tema	4	4	Instrukcija Operacija Podatak Upravljačka jedinica
30	Programski jezik	Programski jezik	Nastavna jedinica	0	3	Programski jezik Programski jezik niske razine Programski jezik visoke razine
31	Programski jezik niske razine	Programski jezik	Nastavna jedinica	1	2	Asembler Programski jezik niske razine
32	Programski jezik visoke razine	Programski jezik	Nastavna jedinica	1	5	Basic C Fortran Pascal Programski jezik visoke razine
33	Programski jezik	Programski jezik	Nastavna tema	0	8	Asembler Basic C Fortran Pascal Programski jezik Programski jezik niske razine Programski jezik visoke razine
34	Računalni sustav	Računalni sustav	Nastavna cjelina	0	61	Aplikacijska programska podrška Aritmetička operacija Aritmetičko-Logička jedinica Basic C Centralna jedinica Centralna procesorska jedinica Disjunkcija Disketa DOS Fortran Informacija Instrukcija Interpretator Izlazna jedinica Jezični prevoditelji Kompaktni disk Kompilator Konjunkcija Logička operacija Masovna memorija Memorija Miš Modem Monitor Mrežna kartica

						Negacija Obrada podataka Oduzimanje Operacija Operacijski sustav Paralelni prijenos podataka Pascal Podatak Pohrana podataka Prijenos podataka Prikazivanje podataka Programska podrška Programski jezik visoke razine Računalni sustav Računalo Radna memorija RAM ROM Serijski prijenos podataka Sistemska programska podrška Štampač Tehnička podrška Temeljna funkcija računala Tipkovnica Tvrdi disk Ulazna jedinica Unix Unos podataka Upravljačka jedinica Upravljanje podataka Uređaj za komunikaciju Uslužni programi Windows Zbrajanje Značenje
35	Aritmetička operacija	Računalni sustav	Nastavna jedinica	6	3	Aritmetička operacija Oduzimanje Zbrajanje
36	Aritmetičko-Logička jedinica	Računalni sustav	Nastavna jedinica	5	3	Aritmetička operacija Aritmetičko-Logička jedinica Logička operacija
37	Centralna jedinica	Računalni sustav	Nastavna jedinica	2	3	Centralna jedinica Masovna memorija Računalo
38	Centralna procesorska jedinica	Računalni sustav	Nastavna jedinica	4	3	Aritmetičko-Logička jedinica Centralna procesorska jedinica Upravljačka jedinica
39	Izlazna jedinica	Računalni sustav	Nastavna jedinica	2	4	Izlazna jedinica Monitor Prikazivanje podataka Štampač
40	Jezični prevoditelji	Računalni sustav	Nastavna jedinica	3	4	Interpretator Jezični prevoditelj Kompilator Programski jezik visoke razine
41	Logička operacija	Računalni sustav	Nastavna jedinica	6	4	Disjunkcija Konjunkcija Logička operacija Negacija
42	Masovna memorija	Računalni sustav	Nastavna jedinica	3	4	Disketa Kompaktni disk Masovna memorija Tvrdi disk
43	Memorija	Računalni sustav	Nastavna jedinica	2	4	Masovna memorija Memorija Pohrana podataka Radna memorija
44	Modem	Računalni sustav	Nastavna jedinica	3	2	Modem Serijski prijenos podataka
45	Obrada podataka	Računalni sustav	Nastavna	2	3	Obrada podataka

			jedinica			Pohrana podataka Upravljanje podataka
46	Operacija	Računalni sustav	Nastavna jedinica	7	3	Aritmetička operacija Logička operacija Operacija
47	Operacijski sustav	Računalni sustav	Nastavna jedinica	3	4	DOS Operacijski sustav Unix Windows
48	Podatak	Računalni sustav	Nastavna jedinica	7	3	Informacija Podatak Značenje
49	Prijenos podataka	Računalni sustav	Nastavna jedinica	3	3	Paralelni prijenos podataka Prijenos podataka Serijski prijenos podataka
50	Programska podrška	Računalni sustav	Nastavna jedinica	1	3	Aplikacijska programska podrška Programska podrška Sistemska programska podrška
51	Programski jezik visoke razine	Računalni sustav	Nastavna jedinica	4	5	Basic C Fortran Pascal Programski jezik visoke razine
52	Računalni sustav	Računalni sustav	Nastavna jedinica	0	4	Programska podrška Računalni sustav Tehnička podrška Temeljna funkcija računala
53	Računalo	Računalni sustav	Nastavna jedinica	3	3	Centralna procesorska jedinica Računalo Radna memorija
54	Radna memorija	Računalni sustav	Nastavna jedinica	3	3	Radna memorija RAM ROM
55	Sistemska programska podrška	Računalni sustav	Nastavna jedinica	2	4	Jezični prevoditelji Operacijski sustav Sistemska programska podrška Uslužni programi
56	Tehnička podrška	Računalni sustav	Nastavna jedinica	1	6	Centralna jedinica Izlazna jedinica Memorija Tehnička podrška Ulazna jedinica Uređaj za komunikaciju
57	Temeljna funkcija računala	Računalni sustav	Nastavna jedinica	1	4	Obrada podataka Prikazivanje podataka Temeljna funkcija računala Unos podataka
58	Ulazna jedinica	Računalni sustav	Nastavna jedinica	2	4	Miš Tipkovnica Ulazna jedinica Unos podataka
59	Upravljačka jedinica	Računalni sustav	Nastavna jedinica	5	4	Instrukcija Operacija Podatak Upravljačka jedinica
60	Uređaj za komunikaciju	Računalni sustav	Nastavna jedinica	2	4	Modem Mrežna kartica Prijenos podataka Uređaj za komunikaciju
61	Centralna jedinica	Računalni sustav	Nastavna tema	2	8	Centralna jedinica Centralna procesorska jedinica Disketa Kompaktni disk Masovna memorija Računalo Radna memorija Tvrđi disk
62	Centralna procesorska	Računalni sustav	Nastavna	4	8	Aritmetička operacija

	jedinica		tema			Aritmetičko-Logička jedinica Centralna procesorska jedinica Instrukcija Logička operacija Operacija Podatak Upravljačka jedinica
63	Memorija	Računalni sustav	Nastavna tema	2	9	Disketa Kompaktni disk Masovna memorija Memorija Pohrana podataka Radna memorija RAM ROM Tvrđi disk
64	Operacija	Računalni sustav	Nastavna tema	7	8	Aritmetička operacija Disjunkcija Konjunkcija Logička operacija Negacija Oduzimanje Operacija Zbrajanje
65	Računalni sustav	Računalni sustav	Nastavna tema	0	14	Aplikacijska programska podrška Centralna jedinica Izlazna jedinica Memorija Obrada podataka Prikazivanje podataka Programska podrška Računalni sustav Sistemska programska podrška Tehnička podrška Temeljna funkcija računala Ulazna jedinica Unos podataka Uređaj za komunikaciju
66	Sistemska programska podrška	Računalni sustav	Nastavna tema	2	10	DOS Interpretator Jezični prevoditelji Kompilator Operacijski sustav Programski jezik visoke razine Sistemska programska podrška Unix Uslužni programi Windows
67	Uređaj za komunikaciju	Računalni sustav	Nastavna tema	2	6	Modem Mrežna kartica Paralelni prijenos podataka Prijenos podataka Serijski prijenos podataka Uređaj za komunikaciju

Tablica 8.4. Elementi računalom oblikovanog nastavnog sadržaja "Računalo kao sustav" - abecedno

Rbr	Korijen elementa	Oznaka cjeline	Vrsta elementa	Udaljenost od korijena određene cjeline	Broj koncepata	Popis koncepata koji se nalaze u elementu
1	Aritmetička operacija	Model računalnog sustava	Nastavna jedinica	5	3	Aritmetička operacija Oduzimanje Zbrajanje
2	Aritmetička operacija	Računalni sustav	Nastavna jedinica	6	3	Aritmetička operacija Oduzimanje Zbrajanje
3	Aritmetičko-Logička jedinica	Model računalnog sustava	Nastavna jedinica	4	3	Aritmetička operacija Aritmetičko-Logička jedinica Logička operacija
4	Aritmetičko-Logička jedinica	Računalni sustav	Nastavna jedinica	5	3	Aritmetička operacija Aritmetičko-Logička jedinica Logička operacija
5	Aritmetičko-Logička jedinica	Model računalnog sustava	Nastavna tema	4	8	Aritmetička operacija Aritmetičko-Logička jedinica Disjunkcija Konjunkcija Logička operacija Negacija Oduzimanje Zbrajanje
6	Brojevni sustav	Brojevni sustav	Nastavna jedinica	0	5	Binarni brojevni sustav Brojevni sustav Dekadski brojevni sustav Heksadekadski brojevni sustav Oktalni brojevni sustav
7	Centralna jedinica	Model računalnog sustava	Nastavna jedinica	1	3	Centralna jedinica Masovna memorija Računalo
8	Centralna jedinica	Računalni sustav	Nastavna jedinica	2	3	Centralna jedinica Masovna memorija Računalo
9	Centralna jedinica	Računalni sustav	Nastavna tema	2	8	Centralna jedinica Centralna procesorska jedinica Disketa Kompaktni disk Masovna memorija Računalo Radna memorija Tvrđi disk
10	Centralna procesorska jedinica	Model računalnog sustava	Nastavna jedinica	3	3	Aritmetičko-Logička jedinica Centralna procesorska jedinica Upravljačka jedinica
11	Centralna procesorska jedinica	Računalni sustav	Nastavna jedinica	4	3	Aritmetičko-Logička jedinica Centralna procesorska jedinica Upravljačka jedinica
12	Centralna procesorska jedinica	Računalni sustav	Nastavna tema	4	8	Aritmetička operacija Aritmetičko-Logička jedinica Centralna procesorska jedinica Instrukcija Logička operacija Operacija Podatak Upravljačka jedinica
13	I sklop	Logički sklop	Nastavna jedinica	1	2	I sklop Konjunkcija
14	ILI sklop	Logički sklop	Nastavna jedinica	1	2	Disjunkcija ILI sklop
15	Izlazna jedinica	Model računalnog sustava	Nastavna jedinica	1	4	Izlazna jedinica Monitor Prikazivanje podataka Štampač
16	Izlazna jedinica	Računalni sustav	Nastavna jedinica	2	4	Izlazna jedinica Monitor Prikazivanje podataka

						Štampač
17	Jezični prevoditelji	Računalni sustav	Nastavna jedinica	3	4	Interpretator Jezični prevoditelj Kompilator Programski jezik visoke razine
18	Logička operacija	Logički sklop	Nastavna jedinica	1	4	Disjunkcija Konjunkcija Logička operacija Negacija
19	Logička operacija	Model računalnog sustava	Nastavna jedinica	5	4	Disjunkcija Konjunkcija Logička operacija Negacija
20	Logička operacija	Računalni sustav	Nastavna jedinica	6	4	Disjunkcija Konjunkcija Logička operacija Negacija
21	Logički sklop	Logički sklop	Nastavna jedinica	0	7	I sklop ILI sklop Logička operacija Logički sklop NE sklop NI sklop NILI sklop
22	Logički sklop	Logički sklop	Nastavna tema	0	10	Disjunkcija I sklop ILI sklop Konjunkcija Logička operacija Logički sklop NE sklop Negacija NI sklop NILI sklop
23	Masovna memorija	Model računalnog sustava	Nastavna jedinica	2	4	Disketa Kompaktni disk Masovna memorija Tvrđi disk
24	Masovna memorija	Računalni sustav	Nastavna jedinica	3	4	Disketa Kompaktni disk Masovna memorija Tvrđi disk
25	Memorija	Računalni sustav	Nastavna jedinica	2	4	Masovna memorija Memorija Pohrana podataka Radna memorija
26	Memorija	Računalni sustav	Nastavna tema	2	9	Disketa Kompaktni disk Masovna memorija Memorija Pohrana podataka Radna memorija RAM ROM Tvrđi disk
27	Model računalnog sustava	Model računalnog sustava	Nastavna cjelina	0	33	...
28	Model računalnog sustava	Model računalnog sustava	Nastavna jedinica	0	4	Centralna jedinica Izlazna jedinica Model računalnog sustava Ulazna jedinica
29	Model računalnog sustava	Model računalnog sustava	Nastavna tema	0	12	Centralna jedinica Izlazna jedinica Masovna memorija Miš Model računalnog sustava Monitor Prikazivanje podataka Računalo Štampač

						Tipkovnica Ulazna jedinica Unos podataka
30	Modem	Računalni sustav	Nastavna jedinica	3	2	Modem Serijski prijenos podataka
31	NE sklop	Logički sklop	Nastavna jedinica	1	2	NE sklop Negacija
32	NI sklop	Logički sklop	Nastavna jedinica	1	3	Konjunktija Negacija NI sklop
33	NILI sklop	Logički sklop	Nastavna jedinica	1	3	Disjunktija Negacija NILI sklop
34	Obrada podataka	Računalni sustav	Nastavna jedinica	2	3	Obrada podataka Pohrana podataka Upravljanje podataka
35	Operacija	Model računalnog sustava	Nastavna jedinica	6	3	Aritmetička operacija Logička operacija Operacija
36	Operacija	Računalni sustav	Nastavna jedinica	7	3	Aritmetička operacija Logička operacija Operacija
37	Operacija	Model računalnog sustava	Nastavna tema	6	8	Aritmetička operacija Disjunktija Konjunktija Logička operacija Negacija Oduzimanje Operacija Zbrajanje
38	Operacija	Računalni sustav	Nastavna tema	7	8	Aritmetička operacija Disjunktija Konjunktija Logička operacija Negacija Oduzimanje Operacija Zbrajanje
39	Operacijski sustav	Računalni sustav	Nastavna jedinica	3	4	DOS Operacijski sustav Unix Windows
40	Podatak	Model računalnog sustava	Nastavna jedinica	6	3	Informacija Podatak Značenje
41	Podatak	Računalni sustav	Nastavna jedinica	7	3	Informacija Podatak Značenje
42	Prijenos podataka	Računalni sustav	Nastavna jedinica	3	3	Paralelni prijenos podataka Prijenos podataka Serijski prijenos podataka
43	Programska podrška	Računalni sustav	Nastavna jedinica	1	3	Aplikacijska programska podrška Programska podrška Sistemska programska podrška
44	Programski jezik	Programski jezik	Nastavna jedinica	0	3	Programski jezik Programski jezik niske razine Programski jezik visoke razine
45	Programski jezik	Programski jezik	Nastavna tema	0	8	Asembler Basic C Fortran Pascal Programski jezik Programski jezik niske razine Programski jezik visoke razine
46	Programski jezik niske razine	Programski jezik	Nastavna jedinica	1	2	Asembler Programski jezik niske razine
47	Programski jezik visoke	Programski jezik	Nastavna	1	5	Basic

	razine		jedinica			C Fortran Pascal Programski jezik visoke razine
48	Programski jezik visoke razine	Računalni sustav	Nastavna jedinica	4	5	Basic C Fortran Pascal Programski jezik visoke razine
49	Računalni sustav	Računalni sustav	Nastavna cjelina	0	61	...
50	Računalni sustav	Računalni sustav	Nastavna jedinica	0	4	Programska podrška Računalni sustav Tehnička podrška Temeljna funkcija računala
51	Računalni sustav	Računalni sustav	Nastavna tema	0	14	Aplikacijska programska podrška Centralna jedinica Izlazna jedinica Memorija Obrada podataka Prikazivanje podataka Programska podrška Računalni sustav Sistemska programska podrška Tehnička podrška Temeljna funkcija računala Ulazna jedinica Unos podataka Uređaj za komunikaciju
52	Računalo	Model računalnog sustava	Nastavna jedinica	2	3	Centralna procesorska jedinica Računalo Radna memorija
53	Računalo	Računalni sustav	Nastavna jedinica	3	3	Centralna procesorska jedinica Računalo Radna memorija
54	Računalo	Model računalnog sustava	Nastavna tema	2	7	Aritmetičko-Logička jedinica Centralna procesorska jedinica Računalo Radna memorija RAM ROM Upravljačka jedinica
55	Radna memorija	Model računalnog sustava	Nastavna jedinica	3	3	Radna memorija RAM ROM
56	Radna memorija	Računalni sustav	Nastavna jedinica	3	3	Radna memorija RAM ROM
57	Sistemska programska podrška	Računalni sustav	Nastavna jedinica	2	4	Jezični prevoditelji Operacijski sustav Sistemska programska podrška Uslužni programi
58	Sistemska programska podrška	Računalni sustav	Nastavna tema	2	10	DOS Interpreter Jezični prevoditelji Kompilator Operacijski sustav Programski jezik visoke razine Sistemska programska podrška Unix Uslužni programi Windows
59	Tehnička podrška	Računalni sustav	Nastavna jedinica	1	6	Centralna jedinica Izlazna jedinica Memorija Tehnička podrška Ulazna jedinica Uređaj za komunikaciju
60	Temeljna funkcija	Računalni sustav	Nastavna	1	4	Obrada podataka



	računala		jedinica			Prikazivanje podataka Temeljna funkcija računala Unos podataka
61	Ulazna jedinica	Model računalnog sustava	Nastavna jedinica	1	4	Miš Tipkovnica Ulazna jedinica Unos podataka
62	Ulazna jedinica	Računalni sustav	Nastavna jedinica	2	4	Miš Tipkovnica Ulazna jedinica Unos podataka
63	Upravljačka jedinica	Model računalnog sustava	Nastavna jedinica	4	4	Instrukcija Operacija Podatak Upravljačka jedinica
64	Upravljačka jedinica	Računalni sustav	Nastavna jedinica	5	4	Instrukcija Operacija Podatak Upravljačka jedinica
65	Upravljačka jedinica	Model računalnog sustava	Nastavna tema	4	4	Instrukcija Operacija Podatak Upravljačka jedinica
66	Uređaj za komunikaciju	Računalni sustav	Nastavna jedinica	2	4	Modem Mrežna kartica Prijenos podataka Uređaj za komunikaciju
67	Uređaj za komunikaciju	Računalni sustav	Nastavna tema	2	6	Modem Mrežna kartica Paralelni prijenos podataka Prijenos podataka Serijski prijenos podataka Uređaj za komunikaciju

#### **8.4.2.1 Strukture statičkog računalom oblikovanog nastavnog sadržaja za svaki od stereotipova prema znanju**

##### **Stereotip novak**

Za generiranje *računalom oblikovanog nastavnog sadržaja* za stereotip *novak* na područnom znanju "Računalo kao sustav" na raspolaganju imamo 51 nastavnu jedinicu (15 se izbacuje jer se ponavljaju)

- 26 nastavnih jedinica iz cjeline Računalni sustav (oznaka cjeline=1)
- 14 nastavnih jedinica iz cjeline Model računalnog sustava (oznaka cjeline=2)
- 7 nastavnih jedinica iz cjeline Logički sklop (oznaka cjeline=3)
- 3 nastavne jedinice iz cjeline Programski jezik (oznaka cjeline=4)
- 1 nastavna jedinica iz cjeline Brojevni sustav (oznaka cjeline=5)

Zbog navedenih elemenata *računalom oblikovanog nastavnog sadržaja* se dodaje 14 testova (prema Algoritam 11.):

- 6 testova zbog promjene udaljenosti od korijena
- 3 testova zbog niza od tri nastavne jedinice
- 5 testova zbog završetka cjeline

Tablica 8.5. Struktura statičkog računalom oblikovanog nastavnog sadržaja za stereotip *novak*

Korijen elementa	Oznaka cjeline	Udaljenost od korijena određene cjeline	Vrsta elementa	Broj koncepata koji se nalaze u elementu
Računalni sustav	1	0	Nastavna jedinica	4
Programska podrška	1	1	Nastavna jedinica	3
Tehnička podrška	1	1	Nastavna jedinica	6
Temeljna funkcija računala	1	1	Nastavna jedinica	4
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Centralna jedinica	1	2	Nastavna jedinica	3
Izlazna jedinica	1	2	Nastavna jedinica	4
Memorija	1	2	Nastavna jedinica	4
<b>TEST – zbog niza od tri nastavne jedinice</b>				
Obrada podataka	1	2	Nastavna jedinica	3
Sistemska programska podrška	1	2	Nastavna jedinica	4
Ulazna jedinica	1	2	Nastavna jedinica	4
Uređaj za komunikaciju	1	2	Nastavna jedinica	4
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Jezični prevoditelji	1	3	Nastavna jedinica	4
Masovna memorija	1	3	Nastavna jedinica	4
Modem	1	3	Nastavna jedinica	2
<b>TEST – zbog niza od tri nastavne jedinice</b>				
Operacijski sustav	1	3	Nastavna jedinica	4
Prijenos podataka	1	3	Nastavna jedinica	3
Računalo	1	3	Nastavna jedinica	3
Radna memorija	1	3	Nastavna jedinica	3
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Centralna procesorska jedinica	1	4	Nastavna jedinica	3
Programski jezik visoke razine	1	4	Nastavna jedinica	5
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Aritmetičko-Logička jedinica	1	5	Nastavna jedinica	3
Upravljačka jedinica	1	5	Nastavna jedinica	4
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Aritmetička operacija	1	6	Nastavna jedinica	3
Logička operacija	1	6	Nastavna jedinica	4
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Operacija	1	7	Nastavna jedinica	3
Podatak	1	7	Nastavna jedinica	3
<b>TEST – zbog završetka cjeline</b>				
Model računalnog sustava	2	0	Nastavna jedinica	4
<b>TEST – zbog završetka cjeline</b>				
Logički sklop	3	0	Nastavna jedinica	7
I sklop	3	1	Nastavna jedinica	2
ILI sklop	3	1	Nastavna jedinica	2
<b>TEST – zbog niza od tri nastavne jedinice</b>				
NE sklop	3	1	Nastavna jedinica	2
NI sklop	3	1	Nastavna jedinica	3
NILI sklop	3	1	Nastavna jedinica	3
<b>TEST – zbog završetka cjeline</b>				
Programski jezik	4	0	Nastavna jedinica	3
Programski jezik niske razine	4	1	Nastavna jedinica	2
<b>TEST – zbog završetka cjeline</b>				
Brojevi sustav	5	0	Nastavna jedinica	5
<b>TEST – zbog završetka računalom oblikovanog nastavnog sadržaja</b>				

**Učenik stereotipa novak će dobiti sljedeći računalom oblikovani nastavni sadržaj bez promjene stereotipa nakon svakog testa:**

Računalni sustav izvršava Temeljna funkcija računala. Računalni sustav je nadkoncept od Temeljna funkcija računala.

Računalni sustav se sastoji od Programska podrška. Programska podrška je podkoncept od Računalni sustav.

Računalni sustav se sastoji od Tehnička podrška. Tehnička podrška je podkoncept od Računalni sustav.

Programska podrška ima podvrstu Aplikacijska programska podrška. Programska podrška je nadkoncept od Aplikacijska programska podrška.

Programska podrška ima podvrstu Sistemska programska podrška. Sistemska programska podrška je podkoncept od Programska podrška.

Tehnička podrška ima podvrstu Centralna jedinica. Centralna jedinica je podkoncept od Tehnička podrška.

Tehnička podrška ima podvrstu Izlazna jedinica. Tehnička podrška je nadkoncept od Izlazna jedinica.

Tehnička podrška ima podvrstu Memorija. Memorija je podkoncept od Tehnička podrška.

Tehnička podrška ima podvrstu Ulazna jedinica. Tehnička podrška je nadkoncept od Ulazna jedinica.

Tehnička podrška ima podvrstu Uređaj za komunikaciju. Uređaj za komunikaciju je podkoncept od Tehnička podrška.

Temeljna funkcija računala ima primjerak Obrada podataka. Temeljna funkcija računala je nadkoncept od Obrada podataka.

Temeljna funkcija računala ima primjerak Prikazivanje podataka. Prikazivanje podataka je podkoncept od Temeljna funkcija računala.

Temeljna funkcija računala ima primjerak Unos podataka. Temeljna funkcija računala je nadkoncept od Unos podataka.

**TEST – zbog promjene udaljenosti od korijena**

Izlazna jedinica ima podvrstu Monitor. Monitor je podkoncept od Izlazna jedinica.

Izlazna jedinica ima podvrstu Štampač. Izlazna jedinica je nadkoncept od Štampač.

Izlazna jedinica služi za Prikazivanje podataka. Prikazivanje podataka je podkoncept od Izlazna jedinica.

Memorija ima podvrstu Masovna memorija. Memorija je nadkoncept od Masovna memorija.

Memorija ima podvrstu Radna memorija. Radna memorija je podkoncept od Memorija.

Memorija služi za Pohrana podataka. Memorija je nadkoncept od Pohrana podataka.

Obrada podataka se sastoji od Pohrana podataka. Pohrana podataka je podkoncept od Obrada podataka.

Obrada podataka se sastoji od Upravljanje podataka. Obrada podataka je nadkoncept od Upravljanje podataka.

**TEST – zbog niza od tri nastavne jedinice**

Ulazna jedinica ima podvrstu Miš. Miš je podkoncept od Ulazna jedinica.

Ulazna jedinica ima podvrstu Tipkovnica. Ulazna jedinica je nadkoncept od Tipkovnica.

Ulazna jedinica služi za Unos podataka. Unos podataka je podkoncept od Ulazna jedinica.

Sistemska programska podrška ima podvrstu Jezični prevoditelji. Sistemska programska podrška je nadkoncept od Jezični prevoditelji.

Sistemska programska podrška ima podvrstu Operacijski sustav. Operacijski sustav je podkoncept od Sistemska programska podrška.

---

Sistemska programska podrška ima podvrstu Uslužni programi. Sistemska programska podrška je nadkoncept od Uslužni programi.

Uređaj za komunikaciju ima podvrstu Modem. Uređaj za komunikaciju je nadkoncept od Modem.

Uređaj za komunikaciju ima podvrstu Mrežna kartica. Mrežna kartica je podkoncept od Uređaj za komunikaciju.

Uređaj za komunikaciju služi za Prijenos podataka. Uređaj za komunikaciju je nadkoncept od Prijenos podataka.

#### **TEST – zbog promjene udaljenosti od korijena**

Jezični prevoditelji ima podvrstu Interpretator. Interpretator je podkoncept od Jezični prevoditelji.

Jezični prevoditelji ima podvrstu Kompilator. Jezični prevoditelji je nadkoncept od Kompilator.

Jezični prevoditelji prevodi Programski jezik visoke razine. Programski jezik visoke razine je podkoncept od Jezični prevoditelji.

Masovna memorija ima podvrstu Disketa. Masovna memorija je nadkoncept od Disketa.

Masovna memorija ima podvrstu Kompaktni disk. Kompaktni disk je podkoncept od Masovna memorija.

Masovna memorija ima podvrstu Tvrdi disk. Masovna memorija je nadkoncept od Tvrdi disk.

Modem vrši Serijski prijenos podataka. Serijski prijenos podataka je podkoncept od Modem.

#### **TEST – zbog niza od tri nastavne jedinice**

Operacijski sustav ima primjerak DOS. Operacijski sustav je nadkoncept od DOS.

Operacijski sustav ima primjerak Unix. Unix je podkoncept od Operacijski sustav.

Operacijski sustav ima primjerak Windows. Operacijski sustav je nadkoncept od Windows.

Prijenos podataka ima podvrstu Serijski prijenos podataka. Serijski prijenos podataka je podkoncept od Prijenos podataka.

Prijenos podataka ima podvrstu Paralelni prijenos podataka. Prijenos podataka je nadkoncept od Paralelni prijenos podataka.

Radna memorija ima podvrstu RAM. RAM je podkoncept od Radna memorija.

Radna memorija ima podvrstu ROM. Radna memorija je nadkoncept od ROM.

#### **TEST – zbog promjene udaljenosti od korijena**

Centralna procesorska jedinica se sastoji od Aritmetičko-Logička jedinica. Aritmetičko-Logička jedinica je podkoncept od Centralna procesorska jedinica.

Centralna procesorska jedinica se sastoji od Upravljačka jedinica. Centralna procesorska jedinica je nadkoncept od Upravljačka jedinica.

Programski jezik visoke razine ima primjerak Basic. Basic je podkoncept od Programski jezik visoke razine.

Programski jezik visoke razine ima primjerak C. Programski jezik visoke razine je nadkoncept od C.

Programski jezik visoke razine ima primjerak Fortran. Fortran je podkoncept od Programski jezik visoke razine.

Programski jezik visoke razine ima primjerak Pascal. Programski jezik visoke razine je nadkoncept od Pascal.

#### **TEST – zbog promjene udaljenosti od korijena**

Aritmetička operacija ima primjerak Oduzimanje. Oduzimanje je podkoncept od Aritmetička operacija.

---

Aritmetička operacija ima primjerak Zbrajanje. Aritmetička operacija je nadkoncept od Zbrajanje.

Logička operacija ima primjerak Disjunkcija. Disjunkcija je podkoncept od Logička operacija.

Logička operacija ima primjerak Konjunkcija. Logička operacija je nadkoncept od Konjunkcija.

Logička operacija ima primjerak Negacija. Negacija je podkoncept od Logička operacija.

Podatak ima podvrstu Informacija. Informacija je podkoncept od Podatak.

Informacija ima Značenje. Informacija je nadkoncept od Značenje.

### **TEST – zbog završetka cjeline**

Model računalnog sustava se sastoji od Centralna jedinica. Model računalnog sustava je nadkoncept od Centralna jedinica.

Model računalnog sustava se sastoji od Izlazna jedinica. Izlazna jedinica je podkoncept od Model računalnog sustava.

Model računalnog sustava se sastoji od Ulazna jedinica. Model računalnog sustava je nadkoncept od Ulazna jedinica.

Centralna jedinica se sastoji od Masovna memorija. Masovna memorija je podkoncept od Centralna jedinica.

Centralna jedinica se sastoji od Računalo. Centralna jedinica je nadkoncept od Računalo.

### **TEST – zbog promjene udaljenosti od korijena**

Računalo se sastoji od Centralna procesorska jedinica. Centralna procesorska jedinica je podkoncept od Računalo.

Računalo se sastoji od Radna memorija. Računalo je nadkoncept od Radna memorija.

Aritmetičko-Logička jedinica izvršava Aritmetička operacija. Aritmetička operacija je podkoncept od Aritmetičko-Logička jedinica.

Aritmetičko-Logička jedinica izvršava Logička operacija. Aritmetičko-Logička jedinica je nadkoncept od Logička operacija.

Upravljačka jedinica upravlja s Instrukcija. Instrukcija je podkoncept od Upravljačka jedinica.

Instrukcija se sastoji od Operacija. Instrukcija je nadkoncept od Operacija.

Instrukcija se sastoji od Podatak. Podatak je podkoncept od Instrukcija.

Operacija ima podvrstu Aritmetička operacija. Aritmetička operacija je podkoncept od Operacija.

Operacija ima podvrstu Logička operacija. Operacija je nadkoncept od Logička operacija.

### **TEST – zbog završetka cjeline**

Logički sklop ima primjerak I sklop. I sklop je podkoncept od Logički sklop.

Logički sklop ima primjerak ILI sklop. Logički sklop je nadkoncept od ILI sklop.

Logički sklop ima primjerak NE sklop. NE sklop je podkoncept od Logički sklop.

Logički sklop ima primjerak NI sklop. Logički sklop je nadkoncept od NI sklop.

Logički sklop ima primjerak NILI sklop. NILI sklop je podkoncept od Logički sklop.

Logički sklop izvršava Logička operacija. Logički sklop je nadkoncept od Logička operacija.

I sklop izvršava Konjunkcija. Konjunkcija je podkoncept od I sklop.

---

ILI sklop izvršava Disjunkcija. ILI sklop je nadkoncept od Disjunkcija.

**TEST – zbog niza od tri nastavne jedinice**

NE sklop izvršava Negacija. NE sklop je podkoncept od Negacija.

NI sklop izvršava Konjunkcija. NI sklop je nadkoncept od Konjunkcija.

NI sklop izvršava Negacija. Negacija je podkoncept od NI sklop.

NILI sklop izvršava Disjunkcija. NILI sklop je nadkoncept od Disjunkcija.

NILI sklop izvršava Negacija. Negacija je podkoncept od NILI sklop.

**TEST – zbog završetka cjeline**

Programski jezik ima podvrstu Programski jezik niske razine. Programski jezik je nadkoncept od Programski jezik niske razine.

Programski jezik ima podvrstu Programski jezik visoke razine. Programski jezik visoke razine je podkoncept od Programski jezik.

Programski jezik niske razine ima primjerak Asembler. Programski jezik niske razine je nadkoncept od Asembler.

**TEST – zbog završetka cjeline**

Brojevni sustav ima primjerak Binarni brojevni sustav. Binarni brojevni sustav je podkoncept od Brojevni sustav. Brojevni sustav ima primjerak Dekadski brojevni sustav. Brojevni sustav je nadkoncept od Dekadski brojevni sustav.

Brojevni sustav ima primjerak Heksadekadski brojevni sustav. Heksadekadski brojevni sustav je podkoncept od Brojevni sustav.

Brojevni sustav ima primjerak Oktakni brojevni sustav. Brojevni sustav je nadkoncept od Oktakni brojevni sustav.

**TEST – zbog završetka cjeline*****Stereotip početnik***

Za generiranje *računalom oblikovanog nastavnog sadržaja* za stereotip *početnik* na područnom znanju "Računalo kao sustav" na raspolaganju imamo: 5 cjelina, 51 nastavnih jedinica (15 se izbacuje jer se ponavljaju)

- 26 nastavnih jedinica iz cjeline Računalni sustav (oznaka cjeline=1)
- 14 nastavnih jedinica iz cjeline Model računalnog sustava (oznaka cjeline=2)
- 7 nastavnih jedinica iz cjeline Logički sklop (oznaka cjeline=3)
- 3 nastavne jedinice iz cjeline Programski jezik (oznaka cjeline=4)
- 1 nastavna jedinica iz cjeline Brojevni sustav (oznaka cjeline=5)

Zbog navedenih elemenata *računalom oblikovanog nastavnog sadržaja* se dodaje 7 testova (prema Algoritam 13.):

- 4 testova zbog promjene udaljenosti od korijena
- 3 testa zbog završetka cjeline

Tablica 8.6. Struktura statičkog računalom oblikovanog nastavnog sadržaja za stereotip početnik

Korijen elementa	Oznaka cjeline	Udaljenost od korijena određene cjeline	Vrsta elementa	Broj koncepata koji se nalaze u elementu
Računalni sustav	1	0	3	4
Programska podrška	1	1	3	3
Tehnička podrška	1	1	3	6
Temeljna funkcija računala	1	1	3	4
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Centralna jedinica	1	2	3	3
Izlazna jedinica	1	2	3	4
Memorija	1	2	3	4
Obrada podataka	1	2	3	3
Sistemska programska podrška	1	2	3	4
Ulazna jedinica	1	2	3	4
Uređaj za komunikaciju	1	2	3	4
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Jezični prevoditelji	1	3	3	4
Masovna memorija	1	3	3	6
Modem	1	3	3	2
Operacijski sustav	1	3	3	4
Prijenos podataka	1	3	3	3
Računalo	1	3	3	3
Radna memorija	1	3	3	3
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Centralna procesorska jedinica	1	4	3	3
Programski jezik visoke razine	1	4	3	5
Aritmetičko-Logička jedinica	1	5	3	3
Upravljačka jedinica	1	5	3	4
<b>TEST – zbog promjene udaljenosti od korijena</b>				
Aritmetička operacija	1	6	3	3
Logička operacija	1	6	3	4
Operacija	1	7	3	3
Podatak	1	7	3	3
<b>TEST – zbog završetka cjeline</b>				
Model računalnog sustava	2	0	3	4
Logički sklop	3	0	3	7
I sklop	3	1	3	2
ILI sklop	3	1	3	2
NE sklop	3	1	3	2
NI sklop	3	1	3	3
NILI sklop	3	1	3	3
<b>TEST – zbog završetka cjeline</b>				
Programski jezik	4	0	3	3
Programski jezik niske razine	4	1	3	2
Brojevni sustav	5	0	3	15
<b>TEST – zbog završetka računalom oblikovanog nastavnog sadržaja</b>				

---

**Učenik stereotipa početnik će dobiti sljedeći računalom oblikovani nastavni sadržaj bez promjene stereotipa nakon svakog testa:**

Računalni sustav se sastoji od Programska podrška, Tehnička podrška. Ovo su podkoncepti od Računalni sustav.

Računalni sustav izvršava Temeljna funkcija računala. Temeljna funkcija računala ima nadkoncept Računalni sustav.

Programska podrška ima podvrstu Aplikacijska programska podrška, Sistemska programska. Ovo su podkoncepti od Programska podrška.

Tehnička podrška ima podvrstu Centralna jedinica, Izlazna jedinica, Memorija, Ulazna jedinica, Uređaj za komunikaciju. Ovo su podkoncepti od Tehnička podrška.

Temeljna funkcija računala ima primjerak Obrada podataka, Unos podataka, Prikazivanje podataka. Ovo su podkoncepti od Temeljna funkcija računala.

Računalni sustav i Ulazna jedinica su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→Ulazna jedinica.

Računalni sustav i Centralna jedinica su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→ Centralna jedinica.

Računalni sustav i Uređaj za komunikaciju su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→ Uređaj za komunikaciju.

Računalni sustav i Memorija su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→ Memorija.

Računalni sustav i Izlazna jedinica su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→ Izlazna jedinica.

Računalni sustav i Sistemska programska podrška su posredno povezani jer postoji put Računalni sustav→ Programska podrška → Sistemska programska podrška.

Računalni sustav i Aplikacijska programska podrška su posredno povezani jer postoji put Računalni sustav→ Programska podrška → Aplikacijska programska podrška.

Računalni sustav i Obrada podataka su posredno povezani jer postoji put Računalni sustav→ Temeljna funkcija računala→ Obrada podataka.

Računalni sustav i Unos podataka su posredno povezani jer postoji put Računalni sustav→ Temeljna funkcija računala→ Unos podataka.

Računalni sustav i Prikazivanje podataka su posredno povezani jer postoji put Računalni sustav→ Temeljna funkcija računala→ Prikazivanje podataka.

**TEST – zbog promjene udaljenosti od korijena**

Izlazna jedinica ima podvrstu Monitor, Štampač. Ovo su podkoncepti od Izlazna jedinica.

Izlazna jedinica služi za Prikazivanje podataka. Prikazivanje podataka ima nadkoncept Izlazna jedinica.

Memorija ima podvrstu Masovna memorija, Radna memorija. Ovo su podkoncepti od Memorija.

Memorija služi za Pohrana podataka. Pohrana podataka ima nadkoncept Memorija.



---

Obrada podataka se sastoji od Pohrana podataka, Upravljanje podataka. Ovo su podkoncepti od Obrada podataka.

Ulazna jedinica ima podvrstu Miš, Tipkovnica. Ovo su podkoncepti od Ulazna jedinica.  
Ulazna jedinica služi za Unos podataka. Unos podataka ima nadkoncept Ulazna jedinica.

Sistemska programska podrška ima podvrstu Jezični prevoditelji, Operacijski sustav, Uslužni programi. Ovo su podkoncepti od Sistemska programska podrška.

Uređaj za komunikaciju ima podvrstu Modem, Mrežna kartica. Ovo su podkoncepti od Uređaj za komunikaciju.  
Uređaj za komunikaciju služi za Prijenos podataka. Prijenos podataka ima nadkoncept Uređaj za komunikaciju.

#### **TEST – zbog promjene udaljenosti od korijena**

Jezični prevoditelji ima podvrstu Interpretator, Kompilator. Ovo su podkoncepti od Jezični prevoditelji.  
Jezični prevoditelji prevodi Programski jezik visoke razine. Programski jezik visoke razine ima nadkoncept Jezični prevoditelji.

Masovna memorija ima podvrstu Disketa, Kompaktni disk, Tvrdi disk. Ovo su podkoncepti od Masovna memorija.

Modem vrši Serijski prijenos podataka. Serijski prijenos podataka ima nadkoncept Modem.

Operacijski sustav ima primjerak DOS, Unix, Windows. Ovo su podkoncepti od Operacijski sustav.

Prijenos podataka ima podvrstu Paralelni prijenos podataka, Serijski prijenos podataka. Ovo su podkoncepti od Prijenos podataka.

Centralna procesorska jedinica se sastoji od Aritmetičko-Logička jedinica, Upravljačka jedinica. Ovo su podkoncepti od Centralna procesorska jedinica.

Programski jezik visoke razine ima primjerak Basic, C, Fortran, Pascal. Ovo su podkoncepti od Programski jezik visoke razine.

Radna memorija ima podvrstu RAM, ROM. Ovo su podkoncepti od Radna memorija.

#### **TEST – zbog promjene udaljenosti od korijena**

Aritmetička operacija ima primjerak Oduzimanje, Zbrajanje. Ovo su podkoncepti od Aritmetička operacija.

Logička operacija ima primjerak Disjunkcija, Konjukcija, Negacija. Ovo su podkoncepti od Logička operacija.

Podatak ima podvrstu Informacija. Informacija ima nadkoncept Podatak.  
Informacija ima Značenje. Značenje ima nadkoncept Informacija.

#### **TEST – zbog završetka cjeline**

Model računalnog sustava se sastoji od Centralna jedinica, Izlazna jedinica, Ulazna jedinica. Ovo su podkoncepti od Model računalnog sustava.

Centralna jedinica se sastoji od Masovna memorija, Računalo. Ovo su podkoncepti od Centralna jedinica.

Računalo se sastoji od Centralna procesorska jedinica, Radna memorija. Ovo su podkoncepti od Računalo.

Model računalnog sustava i Masovna memorija su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Masovna memorija.

Model računalnog sustava i Računalo su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo.

Model računalnog sustava i Centralna procesorska jedinica su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Centralna procesorska jedinica.

Model računalnog sustava i Radna memorija su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Radna memorija.

Centralna jedinica i Centralna procesorska jedinica su posredno povezani jer postoji put Centralna jedinica → Računalo → Centralna procesorska jedinica.

Centralna jedinica i Radna memorija su posredno povezani jer postoji put Centralna jedinica → Računalo → Radna memorija.

### **TEST – zbog promjene udaljenosti od korijena**

Aritmetičko-Logička jedinica izvršava Aritmetička operacija, Logička operacija. Ovo su podkoncepti od Aritmetičko-Logička jedinica.

Upravljačka jedinica upravlja s Instrukcija. Instrukcija ima nadkoncept Upravljačka jedinica.

Instrukcija se sastoji od Operacija, Podatak. Ovo su podkoncepti od Operacija.

Operacija ima podvrstu Aritmetička operacija, Logička operacija. Ovo su podkoncepti od Operacija.

Upravljačka jedinica i Podatak su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Podatak.

Upravljačka jedinica i Operacija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija.

Upravljačka jedinica i Aritmetička operacija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija → Aritmetička operacija.

Upravljačka jedinica i Logička operacija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija → Logička operacija.

Instrukcija i Aritmetička operacija su posredno povezani jer postoji put Instrukcija → Operacija → Aritmetička operacija.

Instrukcija i Logička operacija su posredno povezani jer postoji put Instrukcija → Operacija → Logička operacija.

### **TEST – zbog završetka cjeline**

Logički sklop ima primjerak I sklop, ILI sklop, NE sklop, NI sklop, NILI sklop. Ovo su podkoncepti od Logički sklop. Logički sklop izvršava Logička operacija. Logička operacija ima nadkoncept Logički sklop.

I sklop izvršava Konjunkcija. Konjunkcija ima nadkoncept I sklop.

ILI sklop izvršava Disjunkcija. Disjunkcija ima nadkoncept ILI sklop.

NE sklop izvršava Negacija. Negacija ima nadkoncept NE sklop.

NI sklop izvršava Konjunkcija, Negacija. Ovo su podkoncepti od NI sklop.

NILI sklop izvršava Disjunkcija, Negacija. Ovo su podkoncepti od NILI sklop.

ILI sklop, NILI sklop izvršava Disjunkcija. Ovo su nadkoncepti od Disjunkcija.

I sklop, NI sklop izvršava Konjukcija. Ovo su nadkoncepti od Konjukcija.

NE, NI sklop, NILI sklop izvršava Negacija. Ovo su nadkoncepti od Negacija.

Logički sklop i Disjunkcija su posredno povezani jer postoji put Logički sklop → ILI sklop → Disjunkcija.

Logički sklop i Negacija su posredno povezani jer postoji put Logički sklop → NE sklop → Negacija.

Logički sklop i Konjukcija su posredno povezani jer postoji put Logički sklop → NI sklop → Konjukcija.

### TEST – zbog završetka cjeline

Programski jezik ima podvrstu Programski jezik niske razine, Programski jezik visoke razine. Ovo su nadkoncepti od Programski jezik.

Programski jezik niske razine ima primjerak Asembler. Asembler ima nadkoncept Programski jezik niske razine.

Brojevni sustav ima primjerak Binarni brojevni sustav, Dekadski brojevni sustav, Heksadekadski brojevni sustav, Oktakni brojevni sustav. Ovo su nadkoncepti od Brojevni sustav.

Programski jezik i Asembler su posredno povezani jer postoji put Programski jezik → Programski jezik niske razine → Asembler.

### TEST – zbog završetka računalom oblikovanog nastavnog sadržaja

## **Stereotip osrednji**

Za generiranje *računalom oblikovanog nastavnog sadržaja* za stereotip *osrednji* na područnom znanju "Računalo kao sustav" na raspolaganju imamo: 5 cjelina, 13 nastavnih tema, 1 nastavna jedinica

- 6 nastavnih tema i 2 nastavne jedinice iz cjeline Računalni sustav (oznaka cjeline=1):
  - Udaljenost od korijena=0: Računalni sustav
  - Udaljenost od korijena=2: Centralna jedinica, Memorija, Sistemska programska podrška, Uređaj za komunikaciju, Obrada podataka
  - Udaljenost od korijena=3: Radna memorija
  - Udaljenost od korijena=4: Centralna procesorska jedinica, Programski jezik visoke razine
  - Udaljenost od korijena=7: Podatak
- 5 nastavnih tema iz cjeline Model računalnog sustava (oznaka cjeline=2):
  - Udaljenost od korijena=0: Model računalnog sustava
  - Udaljenost od korijena=2: Računalo
  - Udaljenost od korijena=4: Aritmetičko-Logička jedinica, Upravljačka jedinica
  - Udaljenost od korijena=6: Operacija
- 1 nastavna tema iz cjeline Logički sklop:
  - Udaljenost od korijena=0: Logički sklop
- 1 nastavna tema iz cjeline Programski jezik:
  - Udaljenost od korijena=0: Programski jezik
- 1 nastavna jedinica iz cjeline Brojevni sustav:
  - Udaljenost od korijena=0: Brojevni sustav

Zbog navedenih elemenata *računalom oblikovanog nastavnog sadržaja* se dodaje 5 testova (prema Algoritam 15.):

- 2 testova zbog niza od tri nastavne jedinice
- 3 testova zbog završetka cjeline

**Tablica 8.7. Struktura statičkog računalom oblikovanog nastavnog sadržaja a za stereotip osrednji**

Korijen elementa	Oznaka cjeline	Udaljenost od korijena određene cjeline	Vrsta elementa	Broj koncepata koji se nalaze u elementu	
Računalni sustav		1	0	2	14
Centralna jedinica		1	2	2	10
Memorija		1	2	2	11
<b>TEST – zbog niza od tri nastavna elementa</b>					
Sistemska programska podrška		1	2	2	10
Uređaj za komunikaciju		1	2	2	6
Obrada podataka		1	2	3	3
<b>TEST – zbog promjene udaljenosti od korijena</b>					
Centralna procesorska jedinica		1	4	2	8
Operacija		1	7	2	8
Podatak		1	7	3	3
<b>TEST – zbog završetka cjeline</b>					
Model računalnog ustava		2	0	2	12
Računalo		2	2	2	7
Aritmetičko-Logička jedinica		2	4	2	8
Upravljačka jedinica		2	4	2	4
<b>TEST – zbog promjene udaljenosti od korijena</b>					
Logički sklop		3	0	2	10
Programski jezik		4	0	2	8
Brojevni sustav		5	0	3	15
<b>TEST – zbog završetka računalom oblikovanog nastavnog sadržaja</b>					

**Učenik stereotipa osrednji će dobiti sljedeći računalom oblikovani nastavni sadržaj bez promjene stereotipa nakon svakog testa:**

Računalni sustav se sastoji od Programska podrška, Tehnička podrška. Ovo su podkoncepti od Računalni sustav.

Računalni sustav izvršava Temeljna funkcija računala. Temeljna funkcija računala ima nadkoncept Računalni sustav.

Programska podrška ima podvrstu Aplikacijska programska podrška, Sistemska programska. Ovo su podkoncepti od Programska podrška.

Tehnička podrška ima podvrstu Centralna jedinica, Izlazna jedinica, Memorija, Ulazna jedinica, Uređaj za komunikaciju. Ovo su podkoncepti od Tehnička podrška.

Temeljna funkcija računala ima primjerak Obrada podataka, Unos podataka, Prikazivanje podataka. Ovo su podkoncepti od Temeljna funkcija računala.

Centralna jedinica se sastoji od Masovna memorija, Računalo. Ovo su podkoncepti od Centralna jedinica.

Masovna memorija ima podvrstu Disketa, Kompaktni disk, Tvrdi disk. Ovo su podkoncepti od Masovna memorija.

Računalo se sastoji od Centralna procesorska jedinica, Radna memorija. Ovo su podkoncepti od Računalo.

Memorija ima podvrstu Masovna memorija, Radna memorija. Ovo su podkoncepti od Memorija.

---

Memorija služi za Pohrana podataka. Pohrana podataka ima nadkoncept Memorija.

Masovna memorija ima podvrstu Disketa, Kompaktni disk, Tvrdi disk. Ovo su podkoncepti od Masovna memorija.

Radna memorija ima podvrstu RAM, ROM. Ovo su podkoncepti od Radna memorija.

Računalni sustav i Ulazna jedinica su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→Ulazna jedinica.

Računalni sustav i Centralna jedinica su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→ Centralna jedinica.

Računalni sustav i Uređaj za komunikaciju su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→ Uređaj za komunikaciju.

Računalni sustav i Memorija su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→ Memorija.

Računalni sustav i Izlazna jedinica su posredno povezani jer postoji put Računalni sustav→Tehnička podrška→ Izlazna jedinica.

Računalni sustav i Sistemska programska podrška su posredno povezani jer postoji put Računalni sustav→ Programska podrška → Sistemska programska podrška.

Računalni sustav i Aplikacijska programska podrška su posredno povezani jer postoji put Računalni sustav→ Programska podrška → Aplikacijska programska podrška.

Računalni sustav i Obrada podataka su posredno povezani jer postoji put Računalni sustav→ Temeljna funkcija računala→ Obrada podataka.

Računalni sustav i Unos podataka su posredno povezani jer postoji put Računalni sustav→ Temeljna funkcija računala→ Unos podataka.

Računalni sustav i Prikazivanje podataka su posredno povezani jer postoji put Računalni sustav→ Temeljna funkcija računala→ Prikazivanje podataka.

Računalni sustav i Pohrana podataka su posredno povezani jer postoji put Računalni sustav→ Tehnička podrška→ Memorija→ Pohrana podataka.

Računalni sustav i Računalo su posredno povezani jer postoji put Računalni sustav → Tehnička podrška → Centralna jedinica → Računalo.

Računalni sustav i Masovna memorija su posredno povezani jer postoji put Računalni sustav → Tehnička podrška → Centralna jedinica → Masovna memorija.

Računalni sustav i Disketa su posredno povezani jer postoji put Računalni sustav → Tehnička podrška → Centralna jedinica → Masovna memorija → Disketa.

Računalni sustav i Kompaktni disk su posredno povezani jer postoji put Računalni sustav → Tehnička podrška → Centralna jedinica → Masovna memorija → Kompaktni disk.

Računalni sustav i Tvrdi disk su posredno povezani jer postoji put Računalni sustav → Tehnička podrška → Centralna jedinica → Masovna memorija → Tvrdi disk.

Računalni sustav i Centralna procesorska jedinica su posredno povezani jer postoji put Računalni sustav → Tehnička podrška → Centralna jedinica → Računalo → Centralna procesorska jedinica.

Računalni sustav i Radna memorija su posredno povezani jer postoji put Računalni sustav → Tehnička podrška → Centralna jedinica → Računalo → Radna memorija.

Računalni sustav i RAM su posredno povezani jer postoji put Računalni sustav → Tehnička podrška → Centralna jedinica → Računalo → Radna memorija→ RAM.

Računalni sustav i ROM su posredno povezani jer postoji put Računalni sustav → Tehnička podrška → Centralna jedinica → Računalo → Radna memorija→ ROM.

Tehnička podrška i Pohrana podataka su posredno povezani jer postoji put Tehnička podrška→ Memorija→ Pohrana podataka.

Tehnička podrška i Računalo su posredno povezani jer postoji put Tehnička podrška → Centralna jedinica → Računalo.

Tehnička podrška i Masovna memorija su posredno povezani jer postoji put Tehnička podrška → Centralna jedinica → Masovna memorija.

Tehnička podrška i Disketa su posredno povezani jer postoji put Tehnička podrška → Centralna jedinica → Masovna memorija → Disketa.

Tehnička podrška i Kompaktni disk su posredno povezani jer postoji put Tehnička podrška → Centralna jedinica → Masovna memorija → Kompaktni disk.

Tehnička podrška i Tvrdi disk su posredno povezani jer postoji put Tehnička podrška → Centralna jedinica → Masovna memorija → Tvrdi disk.

Tehnička podrška i Centralna procesorska jedinica su posredno povezani jer postoji put Tehnička podrška → Centralna jedinica → Računalo → Centralna procesorska jedinica.

Tehnička podrška i Radna memorija su posredno povezani jer postoji put Tehnička podrška → Centralna jedinica → Računalo → Radna memorija.

Tehnička podrška i RAM su posredno povezani jer postoji put Tehnička podrška → Centralna jedinica → Računalo → Radna memorija → RAM.

Tehnička podrška i ROM su posredno povezani jer postoji put Tehnička podrška → Centralna jedinica → Računalo → Radna memorija → ROM.

Centralna jedinica i Centralna procesorska jedinica su posredno povezani jer postoji put Centralna jedinica → Računalo → Centralna procesorska jedinica.

Centralna jedinica i Radna memorija su posredno povezani jer postoji put Centralna jedinica → Računalo → Radna memorija.

Centralna jedinica i Disketa su posredno povezani jer postoji put Centralna jedinica → Masovna memorija → Disketa.

Centralna jedinica i Kompaktni disk su posredno povezani jer postoji put Centralna jedinica → Masovna memorija → Kompaktni disk.

Centralna jedinica i Tvrdi disk su posredno povezani jer postoji put Centralna jedinica → Masovna memorija → Tvrdi disk.

Centralna jedinica i Radna memorija su posredno povezani jer postoji put Centralna jedinica → Računalo → Radna memorija.

Centralna jedinica i RAM su posredno povezani jer postoji put Centralna jedinica → Računalo → Radna memorija → RAM.

Centralna jedinica i ROM su posredno povezani jer postoji put Centralna jedinica → Računalo → Radna memorija → ROM.

Memorija i Disketa su posredno povezani jer postoji put Memorija → Masovna memorija → Disketa.

Memorija i Kompaktni disk su posredno povezani jer postoji put Memorija → Masovna memorija → Kompaktni disk.

Memorija i Tvrdi disk su posredno povezani jer postoji put Memorija → Masovna memorija → Tvrdi disk.

Memorija i RAM su posredno povezani jer postoji put Memorija → Radna memorija → RAM.

Memorija i ROM su posredno povezani jer postoji put Memorija → Radna memorija → ROM.

### TEST – zbg niza od tri nastavne jedinice

Sistemska programska podrška ima podvrstu Jezični prevoditelji, Operacijski sustav, Uslužni programi. Ovo su podkoncepti od Sistemska programska podrška.

Jezični prevoditelji ima podvrstu Interpretator, Kompilator. Ovo su podkoncepti od Jezični prevoditelji.

Jezični prevoditelji prevodi Programski jezik visoke razine. Programski jezik visoke razine ima nadkoncept Jezični prevoditelji.

---

Operacijski sustav ima primjerak DOS, Unix, Windows. Ovo su podkoncepti od Operacijski sustav.

Uređaj za komunikaciju ima podvrstu Modem, Mrežna kartica. Ovo su podkoncepti od Uređaj za komunikaciju. Uređaj za komunikaciju služi za Prijenos podataka. Prijenos podataka ima nadkoncept Uređaj za komunikaciju. Modem vrši Serijski prijenos podataka. Serijski prijenos podataka ima nadkoncept Modem. Prijenos podataka ima podvrstu Paralelni prijenos podataka, Serijski prijenos podataka. Ovo su podkoncepti od Prijenos podataka.

Obrada podataka se sastoji od Pohrana podataka, Upravljanje podataka. Ovo su podkoncepti od Obrada podataka.

Sistemska programska podrška i Interpretator su posredno povezani jer postoji put Sistemska programska podrška → Jezični prevoditelji → Interpretator.

Sistemska programska podrška i Kompilator su posredno povezani jer postoji put Sistemska programska podrška → Jezični prevoditelji → Kompilator.

Sistemska programska podrška i Programski jezik visoke razine su posredno povezani jer postoji put Sistemska programska podrška → Jezični prevoditelji → Programski jezik visoke razine.

Sistemska programska podrška i DOS su posredno povezani jer postoji put Sistemska programska podrška → Operacijski sustav → DOS.

Sistemska programska podrška i Unix su posredno povezani jer postoji put Sistemska programska podrška → Operacijski sustav → Unix.

Sistemska programska podrška i Windows su posredno povezani jer postoji put Sistemska programska podrška → Operacijski sustav → Windows.

Uređaj za komunikaciju i Serijski prijenos podataka su posredno povezani jer postoji put Uređaj za komunikaciju → Modem → Serijski prijenos podataka.

Uređaj za komunikaciju i Paralelni prijenos podataka su posredno povezani jer postoji put Uređaj za komunikaciju → Prijenos podataka → Paralelni prijenos podataka.

### **TEST – zbog niza od tri nastavne jedinice**

Centralna procesorska jedinica se sastoji od Aritmetičko-Logička jedinica, Upravljačka jedinica. Ovo su podkoncepti od Centralna procesorska jedinica.

Aritmetičko-Logička jedinica izvršava Aritmetička operacija, Logička operacija. Ovo su podkoncepti od Aritmetičko-Logička jedinica.

Upravljačka jedinica upravlja s Instrukcija. Instrukcija ima nadkoncept Upravljačka jedinica.

Instrukcija se sastoji od Operacija, Podatak. Ovo su podkoncepti od Operacija.

Operacija ima podvrstu Aritmetička operacija, Logička operacija. Ovo su podkoncepti od Operacija.

Podatak ima podvrstu Informacija. Informacija ima nadkoncept Podatak.

Podatak ima podvrstu Informacija. Informacija ima nadkoncept Podatak.

Informacija ima Značenje. Značenje ima nadkoncept Informacija.

Centralna procesorska jedinica i Aritmetička operacija su posredno povezani jer postoji put Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija.

Centralna procesorska jedinica i Logička operacija su posredno povezani jer postoji put Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija.

Centralna procesorska jedinica i Instrukcija su posredno povezani jer postoji put Centralna procesorska jedinica → Upravljačka jedinica → Instrukcija.

Centralna procesorska jedinica i Operacija su posredno povezani jer postoji put Centralna procesorska jedinica → Upravljačka jedinica → Instrukcija → Operacija.

Centralna procesorska jedinica i Podatak su posredno povezani jer postoji put Centralna procesorska jedinica → Upravljačka jedinica → Instrukcija → Podatak.

Centralna procesorska jedinica i Aritmetička operacija su posredno povezani jer postoji put Centralna procesorska jedinica → Upravljačka jedinica → Instrukcija → Operacija → Aritmetička operacija.

Centralna procesorska jedinica i Logička operacija su posredno povezani jer postoji put Centralna procesorska jedinica → Upravljačka jedinica → Instrukcija → Operacija → Logička operacija.

Centralna procesorska jedinica i Informacija su posredno povezani jer postoji put Centralna procesorska jedinica → Upravljačka jedinica → Instrukcija → Podatak → Informacija.

Centralna procesorska jedinica i Značenje su posredno povezani jer postoji put Centralna procesorska jedinica → Upravljačka jedinica → Instrukcija → Podatak → Informacija → Značenje.

Upravljačka jedinica i Operacija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija.

Upravljačka jedinica i Podatak su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Podatak.

Upravljačka jedinica i Aritmetička operacija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija → Aritmetička operacija.

Upravljačka jedinica i Logička operacija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija → Logička operacija.

Upravljačka jedinica i Informacija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Podatak → Informacija.

Upravljačka jedinica i Značenje su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Podatak → Informacija → Značenje.

Instrukcija i Aritmetička operacija su posredno povezani jer postoji put Instrukcija → Operacija → Aritmetička operacija.

Instrukcija i Logička operacija su posredno povezani jer postoji put Instrukcija → Operacija → Logička operacija.

Instrukcija i Informacija su posredno povezani jer postoji put Instrukcija → Podatak → Informacija.

Instrukcija i Značenje su posredno povezani jer postoji put Instrukcija → Podatak → Informacija → Značenje.

Podatak i Značenje su posredno povezani jer postoji put Podatak → Informacija → Značenje.

### **TEST – zbog završetka cjeline**

Model računalnog sustava se sastoji od Centralna jedinica, Izlazna jedinica, Ulazna jedinica. Ovo su podkoncepti od Model računalnog sustava.

Centralna jedinica se sastoji od Masovna memorija, Računalo. Ovo su podkoncepti od Centralna jedinica.

Izlazna jedinica ima podvrstu Monitor, Štampač. Ovo su podkoncepti od Izlazna jedinica.

Izlazna jedinica služi za Prikazivanje podataka. Prikazivanje podataka ima nadkoncept Izlazna jedinica.

Ulazna jedinica ima podvrstu Miš, Tipkovnica. Ovo su podkoncepti od Ulazna jedinica.

Ulazna jedinica služi za Unos podataka. Unos podataka ima nadkoncept Ulazna jedinica.

Računalo se sastoji od Centralna procesorska jedinica, Radna memorija. Ovo su podkoncepti od Računalo.

Centralna procesorska jedinica se sastoji od Aritmetičko-Logička jedinica, Upravljačka jedinica. Ovo su podkoncepti od Centralna procesorska jedinica.

Radna memorija ima podvrstu RAM, ROM. Ovo su podkoncepti od Radna memorija.



---

Aritmetičko-Logička jedinica izvršava Aritmetička operacija, Logička operacija. Ovo su podkoncepti od Aritmetičko-Logička jedinica.

Aritmetička operacija ima primjerak Oduzimanje, Zbrajanje. Ovo su podkoncepti od Aritmetička operacija.

Logička operacija ima primjerak Disjunkcija, Konjukcija, Negacija. Ovo su podkoncepti od Logička operacija.

Model računalnog sustava i Masovna memorija su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Masovna memorija.

Model računalnog sustava i Računalo su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo.

Model računalnog sustava i Centralna procesorska jedinica su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Centralna procesorska jedinica.

Model računalnog sustava i Radna memorija su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Radna memorija.

Model računalnog sustava i RAM su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Radna memorija → RAM.

Model računalnog sustava i ROM su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Radna memorija → ROM.

Model računalnog sustava i Aritmetičko-Logička jedinica su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica.

Model računalnog sustava i Upravljačka jedinica su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Centralna procesorska jedinica → Upravljačka jedinica.

Model računalnog sustava i Aritmetička operacija su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija.

Model računalnog sustava i Logička operacija su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija.

Model računalnog sustava i Oduzimanje su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija → Oduzimanje.

Model računalnog sustava i Zbrajanje su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija → Zbrajanje.

Model računalnog sustava i Disjunkcija su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija → Disjunkcija.

Model računalnog sustava i Konjukcija su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija → Konjukcija.

Model računalnog sustava i Negacija su posredno povezani jer postoji put Model računalnog sustava → Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija → Negacija.

Model računalnog sustava i Monitor su posredno povezani jer postoji put Model računalnog sustava → Izlazna jedinica → Monitor.

Model računalnog sustava i Štampač su posredno povezani jer postoji put Model računalnog sustava → Izlazna jedinica → Štampač.

Model računalnog sustava i Prikazivanje podataka su posredno povezani jer postoji put Model računalnog sustava → Izlazna jedinica → Prikazivanje podataka.

---

Model računalnog sustava i Miš su posredno povezani jer postoji put Model računalnog sustava → Ulazna jedinica → Miš.

Model računalnog sustava i Tipkovnica su posredno povezani jer postoji put Model računalnog sustava → Ulazna jedinica → Tipkovnica.

Model računalnog sustava i Unos podataka su posredno povezani jer postoji put Model računalnog sustava → Ulazna jedinica → Unos podataka.

Centralna jedinica i Centralna procesorska jedinica su posredno povezani jer postoji put Centralna jedinica → Računalo → Centralna procesorska jedinica.

Centralna jedinica i Radna memorija su posredno povezani jer postoji put Centralna jedinica → Računalo → Radna memorija.

Centralna jedinica i RAM su posredno povezani jer postoji put Centralna jedinica → Računalo → Radna memorija → RAM.

Centralna jedinica i ROM su posredno povezani jer postoji Centralna jedinica → Računalo → Radna memorija → ROM.

Centralna jedinica i Aritmetičko-Logička jedinica su posredno povezani jer postoji put Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica.

Centralna jedinica i Upravljačka jedinica su posredno povezani jer postoji put Centralna jedinica → Računalo → Centralna procesorska jedinica → Upravljačka jedinica.

Centralna jedinica i Aritmetička operacija su posredno povezani jer postoji put Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija.

Centralna jedinica i Logička operacija su posredno povezani jer postoji put Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija.

Centralna jedinica i Oduzimanje su posredno povezani jer postoji put Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija → Oduzimanje.

Centralna jedinica i Zbrajanje su posredno povezani jer postoji put Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija → Zbrajanje.

Centralna jedinica i Disjunkcija su posredno povezani jer postoji put Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija → Disjunkcija.

Centralna jedinica i Konjukcija su posredno povezani jer postoji put Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija → Konjukcija.

Centralna jedinica i Negacija su posredno povezani jer postoji put Centralna jedinica → Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija → Negacija.

Računalo i RAM su posredno povezani jer postoji put Računalo → Radna memorija → RAM.

Računalo i ROM su posredno povezani jer postoji Računalo → Radna memorija → ROM.

Računalo i Aritmetičko-Logička jedinica su posredno povezani jer postoji put Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica.

Računalo i Upravljačka jedinica su posredno povezani jer postoji put Računalo → Centralna procesorska jedinica → Upravljačka jedinica.

Računalo i Aritmetička operacija su posredno povezani jer postoji put Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija.

Računalo i Logička operacija su posredno povezani jer postoji put Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija.

Računalo i Oduzimanje su posredno povezani jer postoji put Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija → Oduzimanje.

Računalo i Zbrajanje su posredno povezani jer postoji put Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija → Zbrajanje.

Računalo i Disjunkcija su posredno povezani jer postoji put Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija → Disjunkcija.

Računalo i Konjunkcija su posredno povezani jer postoji put Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija → Konjunkcija.

Računalo i Negacija su posredno povezani jer postoji put Računalo → Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija → Negacija.

Centralna procesorska jedinica i Aritmetička operacija su posredno povezani jer postoji put Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija.

Centralna procesorska jedinica i Logička operacija su posredno povezani jer postoji put Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija.

Centralna procesorska jedinica i Oduzimanje su posredno povezani jer postoji put Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija → Oduzimanje.

Centralna procesorska jedinica i Zbrajanje su posredno povezani jer postoji put Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Aritmetička operacija → Zbrajanje.

Centralna procesorska jedinica i Disjunkcija su posredno povezani jer postoji put Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija → Disjunkcija.

Centralna procesorska jedinica i Konjunkcija su posredno povezani jer postoji put Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija → Konjunkcija.

Centralna procesorska jedinica i Negacija su posredno povezani jer postoji put Centralna procesorska jedinica → Aritmetičko-Logička jedinica → Logička operacija → Negacija.

Aritmetičko-Logička jedinica i Oduzimanje su posredno povezani jer postoji put Aritmetičko-Logička jedinica → Aritmetička operacija → Oduzimanje.

Aritmetičko-Logička jedinica i Zbrajanje su posredno povezani jer postoji put Aritmetičko-Logička jedinica → Aritmetička operacija → Zbrajanje.

Aritmetičko-Logička jedinica i Disjunkcija su posredno povezani jer postoji put Aritmetičko-Logička jedinica → Logička operacija → Disjunkcija.

Aritmetičko-Logička jedinica i Konjunkcija su posredno povezani jer postoji put Aritmetičko-Logička jedinica → Logička operacija → Konjunkcija.

Aritmetičko-Logička jedinica i Negacija su posredno povezani jer postoji put Aritmetičko-Logička jedinica → Logička operacija → Negacija.

### **TEST – zbog niza od tri nastavne jedinice**

Upravljačka jedinica upravlja s Instrukcija. Instrukcija ima nadkoncept Upravljačka jedinica.

Instrukcija se sastoji od Operacija, Podatak. Ovo su podkoncepti od Operacija.

Podatak ima podvrstu Informacija. Informacija ima nadkoncept Podatak.

Informacija ima Značenje. Značenje ima nadkoncept Informacija.

Operacija ima podvrstu Aritmetička operacija, Logička operacija. Ovo su podkoncepti od Operacija.

Aritmetička operacija ima primjerak Oduzimanje, Zbrajanje. Ovo su podkoncepti od Aritmetička operacija.

Logička operacija ima primjerak Disjunkcija, Konjunkcija, Negacija. Ovo su podkoncepti od Logička operacija.

Upravljačka jedinica i Operacija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija.

Upravljačka jedinica i Podatak su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Podatak.

Upravljačka jedinica i Aritmetička operacija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija → Aritmetička operacija.

Upravljačka jedinica i Logička operacija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija → Logička operacija.

Upravljačka jedinica i Informacija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Podatak → Informacija.

Upravljačka jedinica i Značenje su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Podatak → Informacija → Značenje.

Upravljačka jedinica i Oduzimanje su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija → Aritmetička operacija → Oduzimanje.

Upravljačka jedinica i Zbrajanje su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija → Aritmetička operacija → Zbrajanje.

Upravljačka jedinica i Disjunkcija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija → Logička operacija → Disjunkcija.

Upravljačka jedinica i Konjunkcija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija → Logička operacija → Konjunkcija.

Upravljačka jedinica i Negacija su posredno povezani jer postoji put Upravljačka jedinica → Instrukcija → Operacija → Logička operacija → Negacija.

Instrukcija i Aritmetička operacija su posredno povezani jer postoji put Instrukcija → Operacija → Aritmetička operacija.

Instrukcija i Logička operacija su posredno povezani jer postoji put Instrukcija → Operacija → Logička operacija.

Instrukcija i Informacija su posredno povezani jer postoji put Instrukcija → Podatak → Informacija.

Instrukcija i Značenje su posredno povezani jer postoji put Instrukcija → Podatak → Informacija → Značenje.

Instrukcija i Oduzimanje su posredno povezani jer postoji put Instrukcija → Operacija → Aritmetička operacija → Oduzimanje.

Instrukcija i Zbrajanje su posredno povezani jer postoji put Instrukcija → Operacija → Aritmetička operacija → Zbrajanje.

Instrukcija i Disjunkcija su posredno povezani jer postoji put Instrukcija → Operacija → Logička operacija → Disjunkcija.

Instrukcija i Konjunkcija su posredno povezani jer postoji put Instrukcija → Operacija → Logička operacija → Konjunkcija.

Instrukcija i Negacija su posredno povezani jer postoji put Instrukcija → Operacija → Logička operacija → Negacija.

Operacija i Oduzimanje su posredno povezani jer postoji put Operacija → Aritmetička operacija → Oduzimanje.

Operacija i Zbrajanje su posredno povezani jer postoji put Operacija → Aritmetička operacija → Zbrajanje.

Operacija i Disjunkcija su posredno povezani jer postoji put Operacija → Logička operacija → Disjunkcija.

Operacija i Konjunkcija su posredno povezani jer postoji put Operacija → Logička operacija → Konjunkcija.

Operacija i Negacija su posredno povezani jer postoji put Operacija → Logička operacija → Negacija.

### TEST – zbog završetka cjeline

Logički sklop ima primjerak I sklop, ILI sklop, NE sklop, NI sklop, NILI sklop. Ovo su podkoncepti od Logički sklop.

Logički sklop izvršava Logička operacija. Logička operacija ima nadkoncept Logički sklop.

I sklop, NI sklop izvršava Konjunkcija. Ovo su nadkoncepti od Konjunkcija.

Logička operacija ima primjerak Konjunkcija. Logička operacija ima podkoncept Konjunkcija.

ILI sklop, NILI sklop izvršava Disjunkcija. Ovo su nadkoncepti od Disjunkcija.

Logička operacija ima primjerak Disjunkcija. Logička operacija ima podkoncept Disjunkcija.

NE sklop, NI sklop, NILI sklop izvršava Negacija. Ovo su nadkoncepti od Negacija.

Logička operacija ima primjerak Negacija. Logička operacija ima podkoncept Negacija.

I sklop izvršava Konjunkcija. Konjunkcija ima nadkoncept I sklop.

ILI sklop izvršava Disjunkcija. Disjunkcija ima nadkoncept ILI sklop.

NE sklop izvršava Negacija. Negacija ima nadkoncept NE sklop.

NI sklop izvršava Konjunkcija, Negacija. Ovo su podkoncepti od NI sklop.  
 NILI sklop izvršava Disjunkcija, Negacija. Ovo su podkoncepti od NILI sklop.

Programski jezik ima podvrstu Programski jezik niske razine, Programski jezik visoke razine. Ovo su podkoncepti od Programski jezik.

Programski jezik niske razine ima primjerak Asembler. Asembler je podkoncept od Programski jezik niske razine.

Programski jezik visoke razine ima primjerak Basic, C, Fortran, Pascal. Ovo su podkoncepti od Programski jezik visoke razine.

Logički sklop i Konjunkcija su posredno povezani jer postoji put Logički sklop → I sklop → Konjunkcija.

Logički sklop i Disjunkcija su posredno povezani jer postoji put Logički sklop → ILI sklop → Disjunkcija.

Logički sklop i Negacija su posredno povezani jer postoji put Logički sklop → NE sklop → Negacija.

Programski jezik i Asembler su posredno povezani jer postoji put Programski jezik → Programski jezik niske razine → Asembler.

Programski jezik i Basic su posredno povezani jer postoji put Programski jezik → Programski jezik visoke razine → Basic.

Programski jezik i C su posredno povezani jer postoji put Programski jezik → Programski jezik visoke razine → C.

Programski jezik i Fortran su posredno povezani jer postoji put Programski jezik → Programski jezik visoke razine → Fortran.

Programski jezik i Pascal su posredno povezani jer postoji put Programski jezik → Programski jezik visoke razine → Pascal.

#### **TEST – zbog završetka cjeline**

Brojevni sustav ima primjerak Binarni brojevni sustav, Dekadski brojevni sustav, Heksadekadski brojevni sustav, Oktalni brojevni sustav. Ovo su podkoncepti od Brojevni sustav.

#### **TEST – zbog završetka računalom oblikovanog nastavnog sadržaja**

### ***Stereotip napredni***

Za generiranje računalom oblikovanog nastavnog sadržaja za stereotip napredni na područnom znanju "Računalo kao sustav" na raspolaganju imamo: 5 cjelina, 13 nastavnih tema, 1 nastavna jedinica

- 6 nastavnih tema i 2 nastavne jedinice iz cjeline Računalni sustav (oznaka cjeline=1):
  - Udaljenost od korijena=0: Računalni sustav
  - Udaljenost od korijena=2: Centralna jedinica, Memorija, Sistemska programska podrška, Uređaj za komunikaciju, Obrada podataka
  - Udaljenost od korijena=4: Centralna procesorska jedinica, Programski jezik visoke razine, Radna memorija
  - Udaljenost od korijena=7: Podatak
- 5 nastavnih tema iz cjeline Model računalnog sustava (oznaka cjeline=2):
  - Udaljenost od korijena=0: Model računalnog sustava
  - Udaljenost od korijena=2: Računalo
  - Udaljenost od korijena=4: Aritmetičko-Logička jedinica, Upravljačka jedinica
  - Udaljenost od korijena=6: Operacija
- 1 nastavna tema iz cjeline Logički sklop:

- Udaljenost od korijena=0: Logički sklop
- 1 nastavna tema iz cjeline Programski jezik:
  - Udaljenost od korijena=0: Programski jezik
- 1 nastavna jedinica iz cjeline Brojevni sustav:
  - Udaljenost od korijena=0: Brojevni sustav

Zbog navedenih elemenata *računalom oblikovanog nastavnog sadržaja* se dodaje 5 testova:

- 3 testa zbog završetka cjeline

**Tablica 8.8. Struktura statičkog računalom oblikovanog nastavnog sadržaja za stereotip napredni**

Korijen elementa	Oznaka cjeline	Udaljenost od korijena određene cjeline	Vrsta elementa	Broj koncepata koji se nalaze u elementu
Računalni sustav	1	0	2	14
Centralna jedinica	1	2	2	10
Memorija	1	2	2	11
Sistemska programska podrška	1	2	2	10
Uređaj za komunikaciju	1	2	2	6
Obrada podataka	1	2	3	3
Centralna procesorska jedinica	1	4	2	8
Operacija	1	7	2	8
Podatak	1	7	3	3
<b>TEST – zbog završetka cjeline</b>				
Model računalnog ustava	2	0	2	12
Računalo	2	2	2	7
Aritmetičko-Logička jedinica	2	4	2	8
Upravljačka jedinica	2	4	2	4
<b>TEST – zbog završetka cjeline</b>				
Logički sklop	3	0	2	10
Programski jezik	4	0	2	8
Brojevni sustav	5	0	3	15
<b>TEST – zbog završetka računalom oblikovanog nastavnog sadržaja</b>				

**Učenik stereotipa napredni će dobiti sljedeći računalom oblikovani nastavni sadržaj bez promjene stereotipa nakon svakog testa:**

Računalni sustav je neposredno povezan s 3 koncepata. Ti koncepti su: Temeljna funkcija računala, Programska podrška, Tehnička podrška.

Računalni sustav ima 3 podkoncepata. Računalni sustav izvršava Temeljna funkcija računala, Računalni sustav se sastoji od Programska podrška, Tehnička podrška.

Računalni sustav ima 0 nadkoncepata.

Temeljna funkcija računala je neposredno povezan s 4 koncepata. Ti koncepti su: Obrada podataka, Prikazivanje podataka, Računalni sustav, Unos podataka.

Temeljna funkcija računala ima 3 podkoncepata. Temeljna funkcija računala ima primjerak Obrada podataka, Prikazivanje podataka, Unos podataka.

Temeljna funkcija računala ima 1 nadkoncepata. Računalni sustav izvršava Temeljna funkcija računala

Programska podrška je neposredno povezan s 3 koncepata. Ti koncepti su: Aplikacijska programska podrška, Sistemska programska podrška, Računalni sustav.

Programska podrška ima 2 podkoncepata. Programska podrška ima podvrstu Aplikacijska programska podrška, Sistemska programska podrška.

---

Programska podrška ima 1 nadkonceptata. Računalni sustav izvršava Programska podrška

Tehnička podrška je neposredno povezan s 6 konceptata. Ti koncepti su: Centralna jedinica, Izlazna jedinica, Memorija, Ulazna jedinica, Računalni sustav, Uređaj za komunikaciju.

Tehnička podrška ima 5 podkonceptata. Tehnička podrška ima podvrstu Centralna jedinica, Izlazna jedinica, Memorija, Ulazna jedinica, Uređaj za komunikaciju.

Tehnička podrška ima 1 nadkonceptata. Računalni sustav izvršava Tehnička podrška

Centralna jedinica je neposredno povezan s 4 konceptata. Ti koncepti su: Masovna memorija, Računalo, Tehnička podrška.

Centralna jedinica ima 2 podkonceptata. Centralna jedinica se sastoji od Masovna memorija, Računalo.

Centralna jedinica ima 2 nadkonceptata. Tehnička podrška ima podvrstu Centralna jedinica. Model računalnog sustava se sastoji od Centralna jedinica.

Masovna memorija je neposredno povezan s 5 konceptata. Ti koncepti su: Centralna jedinica, Disketa, Kompaktni disk, Memorija, Tvrdi disk.

Masovna memorija ima 3 podkonceptata. Masovna memorija ima podvrstu Disketa, Kompaktni disk, Tvrdi disk.

Masovna memorija ima 2 nadkonceptata. Centralna jedinica se sastoji od Masovna memorija. Memorija ima podvrstu Masovna memorija.

Računalo je neposredno povezan s 3 konceptata. Ti koncepti su: Centralna jedinica, Centralna procesorska jedinica, Radna memorija.

Računalo ima 2 podkonceptata. Računalo se sastoji od Centralna procesorska jedinica, Radna memorija.

Računalo ima 1 nadkonceptata. Centralna jedinica se sastoji od Računalo.

Memorija je neposredno povezan s 4 konceptata. Ti koncepti su: Masovna memorija, Radna memorija, Pohrana podataka, Tehnička podrška.

Memorija ima 3 podkonceptata. Memorija ima podvrstu Masovna memorija, Radna memorija. Memorija služi za Pohrana podataka.

Memorija ima 1 nadkonceptata. Tehnička podrška ima podvrstu Memorija.

Masovna memorija je neposredno povezan s 5 konceptata. Ti koncepti su: Centralna jedinica, Disketa, Kompaktni disk, Memorija, Tvrdi disk.

Masovna memorija ima 3 podkonceptata. Masovna memorija ima podvrstu Disketa, Kompaktni disk, Tvrdi disk.

Masovna memorija ima 2 nadkonceptata. Centralna jedinica se sastoji od Masovna memorija. Memorija ima podvrstu Masovna memorija.

Radna memorija je neposredno povezan s 4 konceptata. Ti koncepti su: Centralna jedinica, Disketa, Kompaktni disk, Memorija, Tvrdi disk.

Radna memorija ima 2 podkonceptata. Radna memorija ima podvrstu RAM, ROM.

Radna memorija ima 2 nadkonceptata. Memorija ima podvrstu Radna memorija. Računalo se sastoji od Radna memorija.

Sistemska programska podrška je neposredno povezan s 4 konceptata. Ti koncepti su: Jezični prevoditelji, Operacijski sustav, Programska podrška, Uslužni programi.

Sistemska programska podrška ima 3 podkonceptata. Sistemska programska podrška ima podvrstu Jezični prevoditelji, Operacijski sustav, Uslužni programi.

Sistemska programska podrška ima 1 nadkonceptata. Programska podrška ima podvrstu Sistemska programska podrška.

Jezični prevoditelji je neposredno povezan s 4 konceptata. Ti koncepti su: Interpretator, Kompilator, Programski jezik visoke razine, Sistemska programska podrška.

Jezični prevoditelji ima 3 podkonceptata. Jezični prevoditelji ima podvrstu Interpretator, Kompilator, Programski jezik visoke razine.

Jezični prevoditelji ima 1 nadkonceptata. Sistemska programska podrška ima podvrstu Jezični prevoditelji.

---

Operacijski sustav je neposredno povezan s 4 konceptata. Ti koncepti su: DOS, Unix, Windows, Sistemska programska podrška.

Operacijski sustav ima 3 podkonceptata. Operacijski sustav ima primjerak DOS, Unix, Windows.

Operacijski sustav ima 1 nadkonceptata. Sistemska programska podrška ima podvrstu Operacijski sustav.

Uređaj za komunikaciju je neposredno povezan s 4 konceptata. Ti koncepti su: Modem, Mrežna kartica, Prijenos podataka, Tehnička podrška.

Uređaj za komunikaciju ima 3 podkonceptata. Uređaj za komunikaciju ima podvrstu Modem, Mrežna kartica.

Uređaj za komunikaciju služi za Prijenos podataka.

Uređaj za komunikaciju ima 1 nadkonceptata. Tehnička podrška ima podvrstu Uređaj za komunikaciju.

Modem je neposredno povezan s 2 konceptata. Ti koncepti su: Serijski prijenos podataka, Uređaj za komunikaciju.

Modem ima 1 podkonceptata. Modem vrši Serijski prijenos podataka.

Modem ima 1 nadkonceptata. Uređaj za komunikaciju ima podvrstu Modem.

Prijenos podataka je neposredno povezan s 3 konceptata. Ti koncepti su: Paralelni prijenos podataka, Serijski prijenos podataka, Uređaj za komunikaciju.

Prijenos podataka ima 2 podkonceptata. Prijenos podataka ima podvrstu Paralelni prijenos podataka, Serijski prijenos podataka.

Prijenos podataka ima 1 nadkonceptata. Uređaj za komunikaciju služi za Prijenos podataka.

Obrada podataka je neposredno povezan s 3 konceptata. Ti koncepti su: Pohrana podataka, Upravljanje podataka, Temeljna funkcija računala.

Obrada podataka ima 2 podkonceptata. Obrada podataka se sastoji od Pohrana podataka, Upravljanje podataka.

Obrada podataka ima 1 nadkonceptata. Temeljna funkcija računala ima primjerak Obrada podataka.

Centralna procesorska jedinica je neposredno povezan s 3 konceptata. Ti koncepti su: Aritmetičko-Logička jedinica, Računalo, Upravljačka jedinica.

Centralna procesorska jedinica ima 2 podkonceptata. Centralna procesorska jedinica se sastoji od Aritmetičko-Logička jedinica, Upravljačka jedinica.

Centralna procesorska jedinica ima 1 nadkonceptata. Računalo se sastoji od Centralna procesorska jedinica.

Aritmetičko-Logička jedinica je neposredno povezan s 3 konceptata. Ti koncepti su: Aritmetička operacija, Centralna procesorska jedinica, Logička operacija.

Aritmetičko-Logička jedinica ima 2 podkonceptata. Aritmetičko-Logička jedinica izvršava Aritmetička operacija, Logička operacija.

Aritmetičko-Logička jedinica ima 1 nadkonceptata. Centralna procesorska jedinica se sastoji od Aritmetičko-Logička jedinica.

Upravljačka jedinica je neposredno povezan s 2 konceptata. Ti koncepti su: Centralna procesorska jedinica, Instrukcija.

Upravljačka jedinica ima 1 podkonceptata. Upravljačka jedinica upravlja s Instrukcija.

Upravljačka jedinica ima 1 nadkonceptata. Centralna procesorska jedinica se sastoji od Upravljačka jedinica.

Instrukcija je neposredno povezan s 3 konceptata. Ti koncepti su: Operacija, Podatak, Upravljačka jedinica.

Instrukcija ima 2 podkonceptata. Instrukcija se sastoji od Operacija, Podatak.

Instrukcija ima 1 nadkonceptata. Upravljačka jedinica upravlja s Instrukcija.

Operacija je neposredno povezan s 3 konceptata. Ti koncepti su: Aritmetička operacija, Logička operacija, Operacija.

Operacija ima 2 podkonceptata. Operacija ima podvrstu Aritmetička operacija, Logička operacija.

Operacija ima 1 nadkonceptata. Instrukcija se sastoji od Operacija.



Podatak je neposredno povezan s 2 konceptata. Ti koncepti su: Informacija, Instrukcija.  
 Podatak ima 1 podkonceptata. Podatak ima podvrstu Informacija.  
 Podatak ima 1 nadkonceptata. Instrukcija se sastoji od Podatak.

Podatak je neposredno povezan s 2 konceptata. Ti koncepti su: Instrukcija, Informacija.  
 Podatak ima 1 podkonceptata. Podatak ima podvrstu Informacija.  
 Podatak ima 1 nadkonceptata. Instrukcija se sastoji od Podatak.  
 Informacija je neposredno povezan s 2 konceptata. Ti koncepti su: Podatak, Značenje.  
 Informacija ima 1 podkonceptata. Informacija ima Značenje.  
 Informacija ima 1 nadkonceptata. Podatak ima podvrstu Informacija.

### **TEST – zbog završetka cjeline**

Model računalnog sustava je neposredno povezan s 3 konceptata. Ti koncepti su: Centralna jedinica, Izlazna jedinica, Ulazna jedinica.

Model računalnog sustava ima 3 podkonceptata. Model računalnog sustava se sastoji od Centralna jedinica, Izlazna jedinica, Ulazna jedinica.

Model računalnog sustava ima 0 nadkonceptata.

Centralna jedinica je neposredno povezan s 4 konceptata. Ti koncepti su: Masovna memorija, Model računalnog sustava, Računalo, Tehnička podrška.

Centralna jedinica ima 2 podkonceptata. Centralna jedinica se sastoji od Masovna memorija, Računalo.

Centralna jedinica ima 2 nadkonceptata. Tehnička podrška ima podvrstu Centralna jedinica. Model računalnog sustava se sastoji od Centralna jedinica.

Izlazna jedinica je neposredno povezan s 5 konceptata. Ti koncepti su: Monitor, Model računalnog sustava, Štampač, Prikazivanje podataka, Tehnička podrška.

Izlazna jedinica ima 3 podkonceptata. Izlazna jedinica ima podvrstu Monitor, Štampač, Prikazivanje podataka.

Izlazna jedinica ima 2 nadkonceptata. Tehnička podrška ima podvrstu Izlazna jedinica. Model računalnog sustava se sastoji od Centralna jedinica.

Ulazna jedinica je neposredno povezan s 5 konceptata. Ti koncepti su: Miš, Tipkovnica, Unos podataka, Model računalnog sustava, Tehnička podrška.

Ulazna jedinica ima 3 podkonceptata. Izlazna jedinica ima podvrstu Miš, Tipkovnica, Unos podataka.

Ulazna jedinica ima 2 nadkonceptata. Tehnička podrška ima podvrstu Izlazna jedinica. Model računalnog sustava se sastoji od Centralna jedinica.

Računalo je neposredno povezan s 3 konceptata. Ti koncepti su: Centralna jedinica, Centralna procesorska jedinica, Radna memorija.

Računalo ima 2 podkonceptata. Računalo se sastoji od Centralna procesorska jedinica, Radna memorija.

Računalo ima 1 nadkonceptata. Centralna jedinica se sastoji od Računalo.

Centralna procesorska jedinica je neposredno povezan s 3 konceptata. Ti koncepti su: Aritmetičko-Logička jedinica, Računalo, Upravljačka jedinica.

Centralna procesorska jedinica ima 2 podkonceptata. Centralna procesorska jedinica se sastoji od Aritmetičko-Logička jedinica, Upravljačka jedinica.

Centralna procesorska jedinica ima 1 nadkonceptata. Računalo se sastoji od Centralna procesorska jedinica.

Radna memorija je neposredno povezan s 4 konceptata. Ti koncepti su: Centralna jedinica, Disketa, Kompaktni disk, Memorija, Tvrdi disk.

Radna memorija ima 2 podkonceptata. Radna memorija ima podvrstu RAM, ROM.

Radna memorija ima 2 nadkonceptata. Memorija ima podvrstu Radna memorija. Računalo se sastoji od Radna memorija.

Aritmetičko-Logička jedinica je neposredno povezan s 3 konceptata. Ti koncepti su: Aritmetička operacija, Centralna procesorska jedinica, Logička operacija.

Aritmetičko-Logička jedinica ima 2 podkonceptata. Aritmetičko-Logička jedinica izvršava Aritmetička operacija, Logička operacija.

Aritmetičko-Logička jedinica ima 1 nadkonceptata. Centralna procesorska jedinica se sastoji od Aritmetičko-Logička jedinica.

Aritmetička operacija je neposredno povezan s 3 konceptata. Ti koncepti su: Aritmetičko-Logička jedinica, Oduzimanje, Operacija, Zbrajanje.

Aritmetička operacija ima 2 podkonceptata. Aritmetička operacija ima primjerak Oduzimanje, Zbrajanje.

Aritmetička operacija ima 2 nadkonceptata. Operacija ima podvrstu Aritmetička operacija. Aritmetičko-Logička jedinica izvršava Aritmetička operacija.

Logička operacija je neposredno povezan s 3 konceptata. Ti koncepti su: Aritmetičko-Logička jedinica, Oduzimanje, Operacija, Zbrajanje.

Logička operacija ima 3 podkonceptata. Logička operacija ima primjerak Disjunkcija, Konjunkcija, Negacija.

Logička operacija ima 2 nadkonceptata. Operacija ima podvrstu Logička operacija. Aritmetičko-Logička jedinica izvršava Logička operacija.

Upravljačka jedinica je neposredno povezan s 2 konceptata. Ti koncepti su: Centralna procesorska jedinica, Instrukcija.

Upravljačka jedinica ima 1 podkonceptata. Upravljačka jedinica upravlja s Instrukcija.

Upravljačka jedinica ima 1 nadkonceptata. Centralna procesorska jedinica se sastoji od Upravljačka jedinica.

Instrukcija je neposredno povezan s 3 konceptata. Ti koncepti su: Operacija, Podatak, Upravljačka jedinica.

Instrukcija ima 2 podkonceptata. Instrukcija se sastoji od Operacija, Podatak.

Instrukcija ima 1 nadkonceptata. Upravljačka jedinica upravlja s Instrukcija.

Podatak je neposredno povezan s 2 konceptata. Ti koncepti su: Informacija, Instrukcija.

Podatak ima 1 podkonceptata. Podatak ima podvrstu Informacija.

Podatak ima 1 nadkonceptata. Instrukcija se sastoji od Podatak.

Operacija je neposredno povezan s 3 konceptata. Ti koncepti su: Aritmetička operacija, Logička operacija, Operacija.

Operacija ima 2 podkonceptata. Operacija ima podvrstu Aritmetička operacija, Logička operacija.

Operacija ima 1 nadkonceptata. Instrukcija se sastoji od Operacija.

Aritmetička operacija je neposredno povezan s 3 konceptata. Ti koncepti su: Aritmetičko-Logička jedinica, Oduzimanje, Operacija, Zbrajanje.

Aritmetička operacija ima 2 podkonceptata. Aritmetička operacija ima primjerak Oduzimanje, Zbrajanje.

Aritmetička operacija ima 2 nadkonceptata. Operacija ima podvrstu Aritmetička operacija. Aritmetičko-Logička jedinica izvršava Aritmetička operacija.

Logička operacija je neposredno povezan s 3 konceptata. Ti koncepti su: Aritmetičko-Logička jedinica, Oduzimanje, Operacija, Zbrajanje.

Logička operacija ima 3 podkonceptata. Logička operacija ima primjerak Disjunkcija, Konjunkcija, Negacija.

Logička operacija ima 2 nadkonceptata. Operacija ima podvrstu Logička operacija. Aritmetičko-Logička jedinica izvršava Logička operacija.

### **TEST – zbog završetka cjeline**

Logički sklop je neposredno povezan s 6 konceptata. Ti koncepti su: I sklop, ILI sklop, Logička operacija, NE sklop, NI sklop, NILI sklop.

Logički sklop ima 6 podkonceptata. Logički sklop ima primjerak I sklop, ILI sklop, NE sklop, NI sklop, NILI sklop.

Logički sklop izvršava Logička operacija.

Logički sklop ima 0 nadkonceptata.

---

I sklop je neposredno povezan s 2 konceptata. Ti koncepti su: Konjunkcija, Logički sklop.

I sklop ima 1 podkonceptata. I sklop izvršava Konjunkcija

I sklop ima 1 nadkonceptata. Logički sklop ima primjerak I sklop.

II sklop je neposredno povezan s 2 konceptata. Ti koncepti su: Disjunkcija, Logički sklop.

II sklop ima 1 podkonceptata. II sklop izvršava Disjunkcija

II sklop ima 1 nadkonceptata. Logički sklop ima primjerak II sklop.

NE sklop je neposredno povezan s 2 konceptata. Ti koncepti su: Negacija, Logički sklop.

NE sklop ima 1 podkonceptata. NE sklop izvršava Negacija

NE sklop ima 1 nadkonceptata. Logički sklop ima primjerak NE sklop.

NI sklop je neposredno povezan s 3 konceptata. Ti koncepti su: Konjunkcija, Logički sklop, Negacija.

NI sklop ima 2 podkonceptata. NI sklop izvršava Konjunkcija, Negacija

NI sklop ima 1 nadkonceptata. Logički sklop ima primjerak NI sklop.

NII sklop je neposredno povezan s 3 konceptata. Ti koncepti su: Disjunkcija, Logički sklop, Negacija.

NII sklop ima 2 podkonceptata. NII sklop izvršava Disjunkcija, Negacija

NII sklop ima 1 nadkonceptata. Logički sklop ima primjerak NII sklop.

Logička operacija je neposredno povezan s 3 konceptata. Ti koncepti su: Aritmetičko-Logička jedinica, Oduzimanje, Operacija, Zbrajanje.

Logička operacija ima 3 podkonceptata. Logička operacija ima primjerak Disjunkcija, Konjunkcija, Negacija.

Logička operacija ima 2 nadkonceptata. Operacija ima podvrstu Logička operacija. Aritmetičko-Logička jedinica izvršava Logička operacija.

Programski jezik je neposredno povezan s 2 konceptata. Ti koncepti su: Programski jezik niske razine, Programski jezik visoke razine.

Programski jezik ima 2 podkonceptata. Programski jezik ima podvrstu Programski jezik niske razine, Programski jezik visoke razine.

Programski jezik ima 0 nadkonceptata.

Programski jezik niske razine je neposredno povezan s 2 konceptata. Ti koncepti su: Programski jezik, Asembler.

Programski jezik niske razine ima 1 podkonceptata. Programski jezik niske razine ima primjerak Asembler.

Programski jezik niske razine ima 1 nadkonceptata. Programski jezik ima podvrstu Programski jezik niske razine

Programski jezik visoke razine je neposredno povezan s 5 konceptata. Ti koncepti su: Programski jezik, Basic, C, Fortran, Pascal.

Programski jezik visoke razine ima 4 podkonceptata. Programski jezik visoke razine ima primjerak Basic, C, Fortran, Pascal.

Programski jezik visoke razine ima 1 nadkonceptata. Programski jezik ima podvrstu Programski jezik visoke razine

Brojevni sustav je neposredno povezan s 4 konceptata. Ti koncepti su: Binarni brojevni sustav, Dekadski brojevni sustav, Heksadekadski brojevni sustav, Oktalni brojevni sustav

Brojevni sustav ima 4 podkonceptata. Brojevni sustav ima primjerak Binarni brojevni sustav, Dekadski brojevni sustav, Heksadekadski brojevni sustav, Oktalni brojevni sustav.

Brojevni sustav ima 0 nadkonceptata.

**TEST – zbog završetka računalom oblikovanog nastavnog sadržaja**

## Stereotip stručnjak

Za generiranje *računalom oblikovanog nastavnog sadržaja* za stereotip *stručnjak* na područnom znanju "Računalo kao sustav" na raspolaganju imamo: 5 cjelina, 2 nastavne cjeline, 2 nastavne teme, 1 nastavna jedinica

- 1 nastavna cjelina iz cjeline Računalni sustav (oznaka cjeline=1):
  - Udaljenost od korijena=0: Računalni sustav
- 1 nastavna cjelina iz cjeline Model računalnog sustava (oznaka cjeline=2):
  - Udaljenost od korijena=0: Model računalnog sustava
- 1 nastavna tema iz cjeline Logički sklop:
  - Udaljenost od korijena=0: Logički sklop
- 1 nastavna tema iz cjeline Programski jezik:
  - Udaljenost od korijena=0: Programski jezik
- 1 nastavna jedinica iz cjeline Brojevi sustav:
  - Udaljenost od korijena=0: Brojevi sustav

Zbog navedenih *računalom oblikovanog nastavnog sadržaja* se dodaje 13 testova:

- 5 testova zbog promjene udaljenosti od korijena
- 3 testova zbog niza od tri nastavne jedinice
- 5 testova zbog završetka cjeline

**Tablica 8.9. Struktura statičkog računalom oblikovanog nastavnog sadržaja za stereotip stručnjak**

Korijen elementa	Oznaka cjeline	Udaljenost od korijena određene cjeline	Vrsta elementa	Broj koncepata koji se nalaze u elementu
Računalni sustav	1	0	1	60
Model računalnog sustava	2	0	1	33
Logički sklop	3	0	2	10
Programski jezik	4	0	2	8
Brojevi sustav	5	0	3	15
<b>TEST – zbog završetka računalom oblikovanog nastavnog sadržaja</b>				

**Učenik stereotipa stručnjak će dobiti sljedeći računalom oblikovani nastavni sadržaj bez promjene stereotipa nakon svakog testa:**

Koncepti su neposredno povezani sljedećim redoslijedom:

Računalni sustav izvršava Temeljna funkcija računala .Temeljna funkcija računala ima primjerak Unos podataka

Temeljna funkcija računala ima primjerak Prikazivanje podataka

Temeljna funkcija računala ima primjerak Obrada podataka. Obrada podataka se sastoji od Pohrana podataka

Obrada podataka se sastoji od Upravljanje podataka

Računalni sustav se sastoji od Programska podrška . Programska podrška ima podvrstu Aplikacijska programska podrška

Programska podrška ima podvrstu Sistemska programska podrška. Sistemska programska podrška ima podvrstu Uslužni programi.

Sistemska programska podrška ima podvrstu Operacijski sustav. Operacijski sustav ima primjerak Unix.

Operacijski sustav ima primjerak DOS

Operacijski sustav ima primjerak Windows

Sistemska programska podrška ima podvrstu Jezični prevoditelji. Jezični prevoditelji ima podvrstu Kompilator.

Jezični prevoditelji ima podvrstu Interpretator

---

Jezični prevoditelji prevodi Programski jezik visoke razine. Programski jezik visoke razine ima primjerak Basic.  
Programski jezik visoke razine ima primjerak Fortran  
Programski jezik visoke razine ima primjerak Pascal  
Programski jezik visoke razine ima primjerak C  
Računalni sustav se sastoji od Tehnička podrška. Tehnička podrška ima podvrstu Uređaj za komunikaciju.  
Uređaj za komunikaciju ima podvrstu Mrežna kartica.  
Uređaj za komunikaciju ima podvrstu Modem. Modem vrši Serijski prijenos podataka.  
Uređaj za komunikaciju služi za Prijenos podataka. Prijenos podataka ima podvrstu Serijski prijenos podataka.  
Prijenos podataka ima podvrstu Paralelni prijenos podataka  
Tehnička podrška ima podvrstu Izlazna jedinica. Izlazna jedinica ima podvrstu Monitor  
Izlazna jedinica ima podvrstu Štampač  
Izlazna jedinica služi za Prikazivanje podataka  
Tehnička podrška ima podvrstu Centralna jedinica. Centralna jedinica se sastoji od Masovna memorija.  
Masovna memorija ima podvrstu Disketa.  
Masovna memorija ima podvrstu Tvrdi disk  
Masovna memorija ima podvrstu Kompaktni disk  
Centralna jedinica se sastoji od Računalo. Računalo se sastoji od Radna memorija. Radna memorija ima podvrstu RAM  
Radna memorija ima podvrstu ROM  
Računalo se sastoji od Centralna procesorska jedinica. Centralna procesorska jedinica se sastoji od Aritmetičko-Logička jedinica. Aritmetičko-Logička jedinica izvršava Aritmetička operacija. Aritmetička operacija ima primjerak Zbrajanje.  
Aritmetička operacija ima primjerak Oduzimanje  
Aritmetičko-Logička jedinica izvršava Logička operacija. Logička operacija ima primjerak Konjunkcija  
Logička operacija ima primjerak Negacija  
Logička operacija ima primjerak Disjunkcija  
Centralna procesorska jedinica se sastoji od Upravljačka jedinica. Upravljačka jedinica upravlja s Instrukcija.  
Instrukcija se sastoji od Operacija. Operacija ima podvrstu Aritmetička operacija  
Operacija ima podvrstu Logička operacija  
Instrukcija se sastoji od Podatak. Podatak ima podvrstu Informacija. Informacija ima Značenje  
Tehnička podrška ima podvrstu Ulazna jedinica. Ulazna jedinica ima podvrstu Tipkovnica  
Ulazna jedinica ima podvrstu Miš  
Ulazna jedinica služi za Unos podataka  
Tehnička podrška ima podvrstu Memorija. Memorija ima podvrstu Radna memorija  
Memorija ima podvrstu Masovna memorija  
Memorija služi za Pohrana podataka  
Kapacitet od Disketa je 1.44MB

Model računalnog sustava se sastoji od Ulazna jedinica  
Model računalnog sustava se sastoji od Izlazna jedinica  
Model računalnog sustava se sastoji od Centralna jedinica

Logički sklop ima primjerak I sklop. I sklop izvršava Konjunkcija  
Logički sklop ima primjerak ILI sklop. ILI sklop izvršava Disjunkcija  
Logički sklop ima primjerak Logička operacija  
Logički sklop ima primjerak NE sklop. NE sklop izvršava Negacija  
Logički sklop ima primjerak NI sklop. NI sklop izvršava Konjunkcija  
NI sklop izvršava Negacija.  
Logički sklop ima primjerak NILI sklop. NILI sklop izvršava Disjunkcija  
NILI sklop izvršava Negacija.

---

Logički sklop izvršava Logička operacija. Logička operacija ima primjerak Disjunkcija.  
Logička operacija ima primjerak Konjunkcija  
Logička operacija ima primjerak Negacija

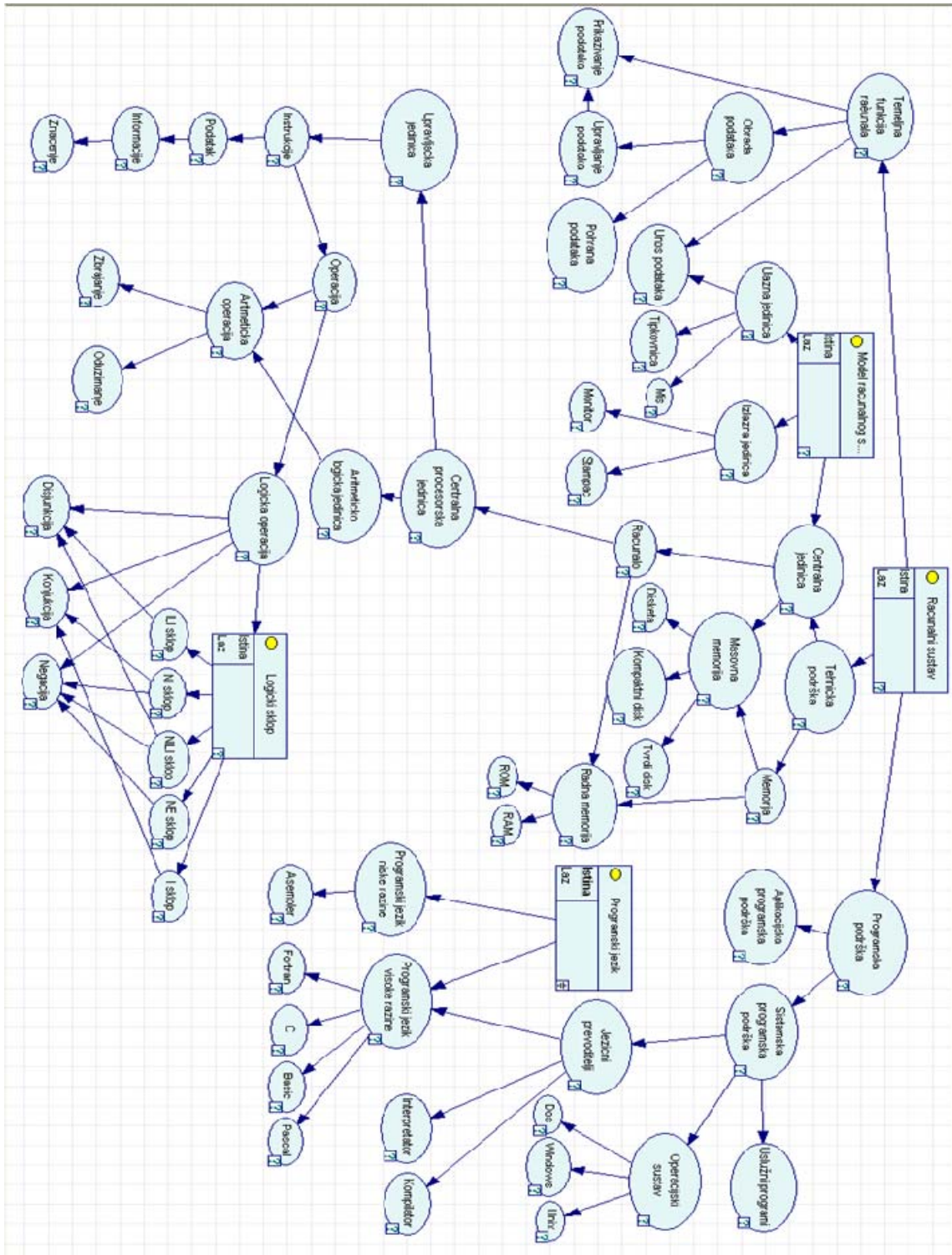
Programski jezik ima podvrstu Programski jezik niske razine. Programski jezik niske razine ima primjerak Asembler .  
Programski jezik ima podvrstu Programski jezik visoke razine.

Brojevni sustav ima primjerak Binarni brojevni sustav .  
Brojevni sustav ima primjerak Dekadski brojevni sustav .  
Brojevni sustav ima primjerak Heksadekadski brojevni sustav .  
Brojevni sustav ima primjerak Oktalni brojevni sustav .

Baza od Binarni brojevni sustav je 2.  
Znamenke od Binarni brojevni sustav je 0 1.  
Baza od Dekadski brojevni sustav je 10.  
Znamenke od Dekadski brojevni sustav je 0 1 2 3 4 5 6 7 8 9.  
Baza od Heksadekadski brojevni sustav je 16.  
Znamenke od Heksadekadski brojevni sustav je 0 1 2 3 4 5 6 7 8 9 A B C D E F.  
Baza od Oktalni brojevni sustav je 8.  
Znamenke od Oktalni brojevni sustav je 0 1 2 3 4 5 6 7.

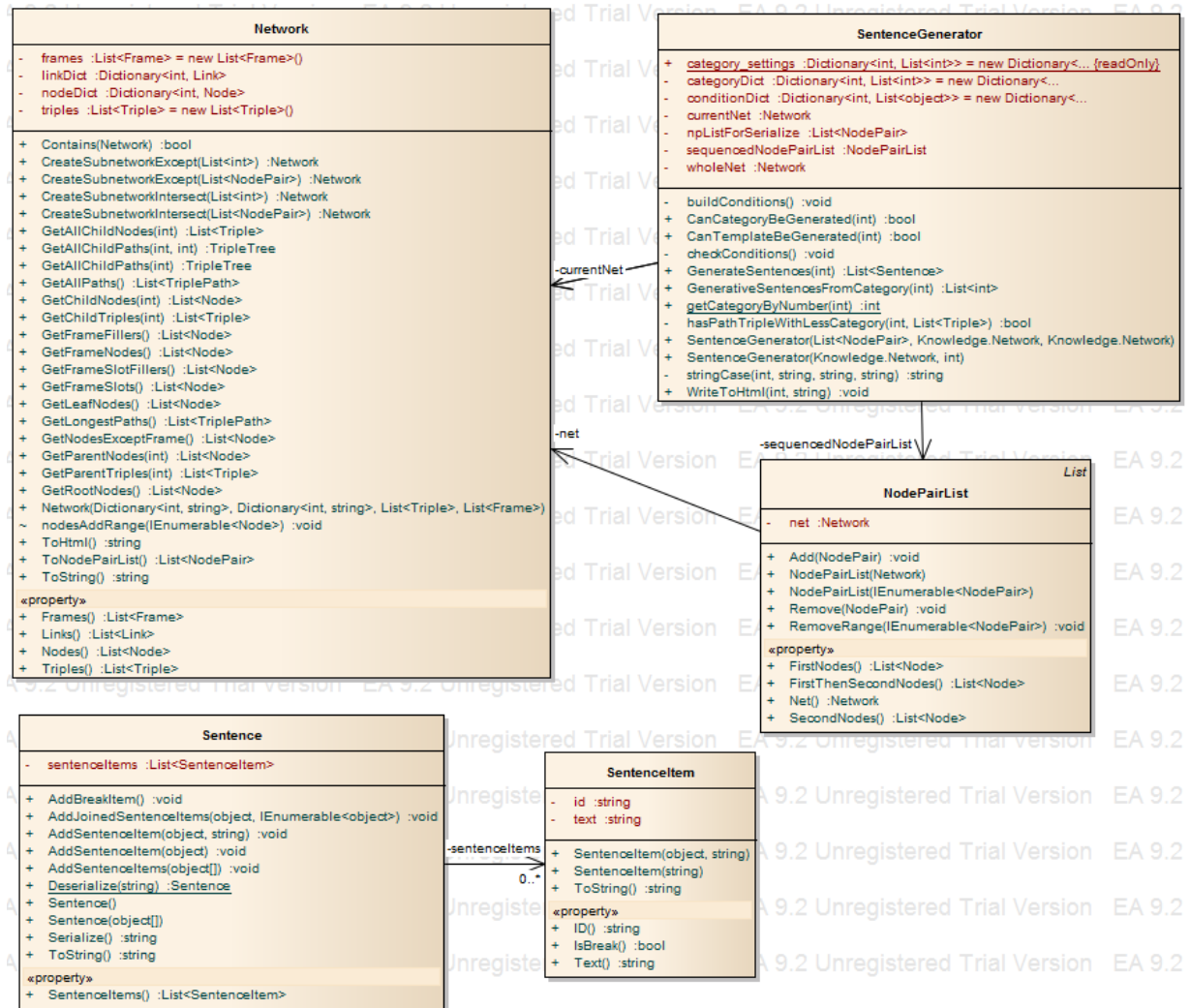
**TEST – zbog završetka računalom oblikovanog nastavnog sadržaja**

# 8.5 Inicijalizirana Bayesova mreža za područno znanje „Računalo kao sustav“



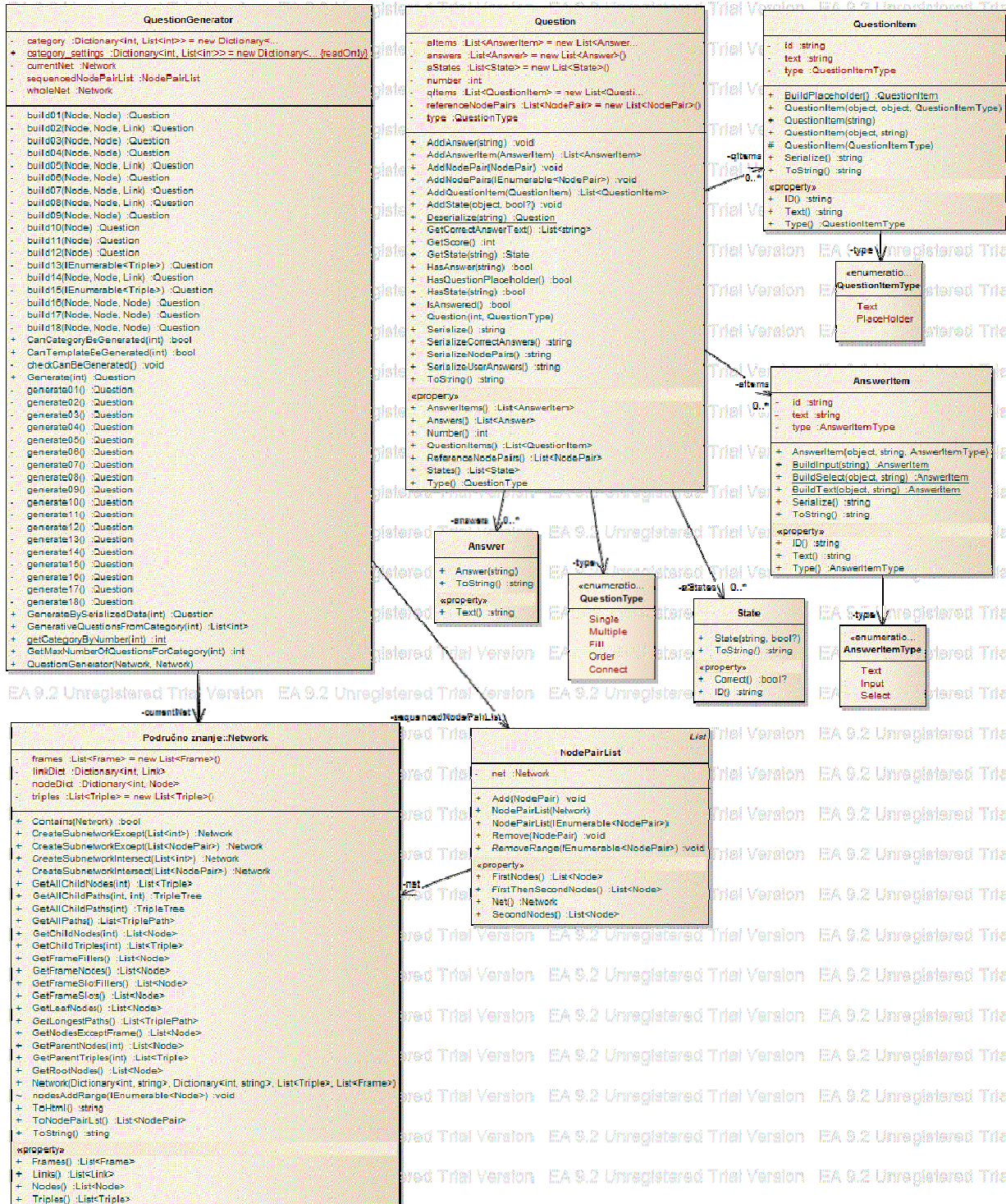
## 8.6 Detaljni dijagrami klasa implementiranog prototipa AC-ware Tutor

Dijagram klasa za učenje:

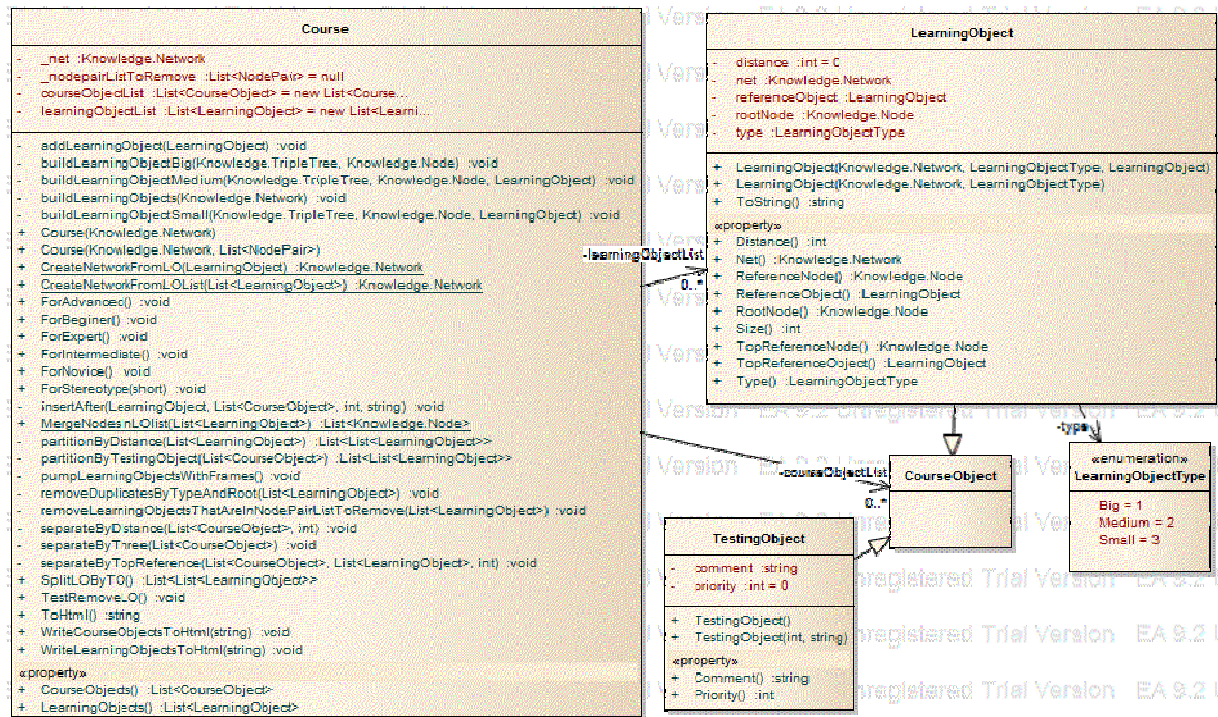




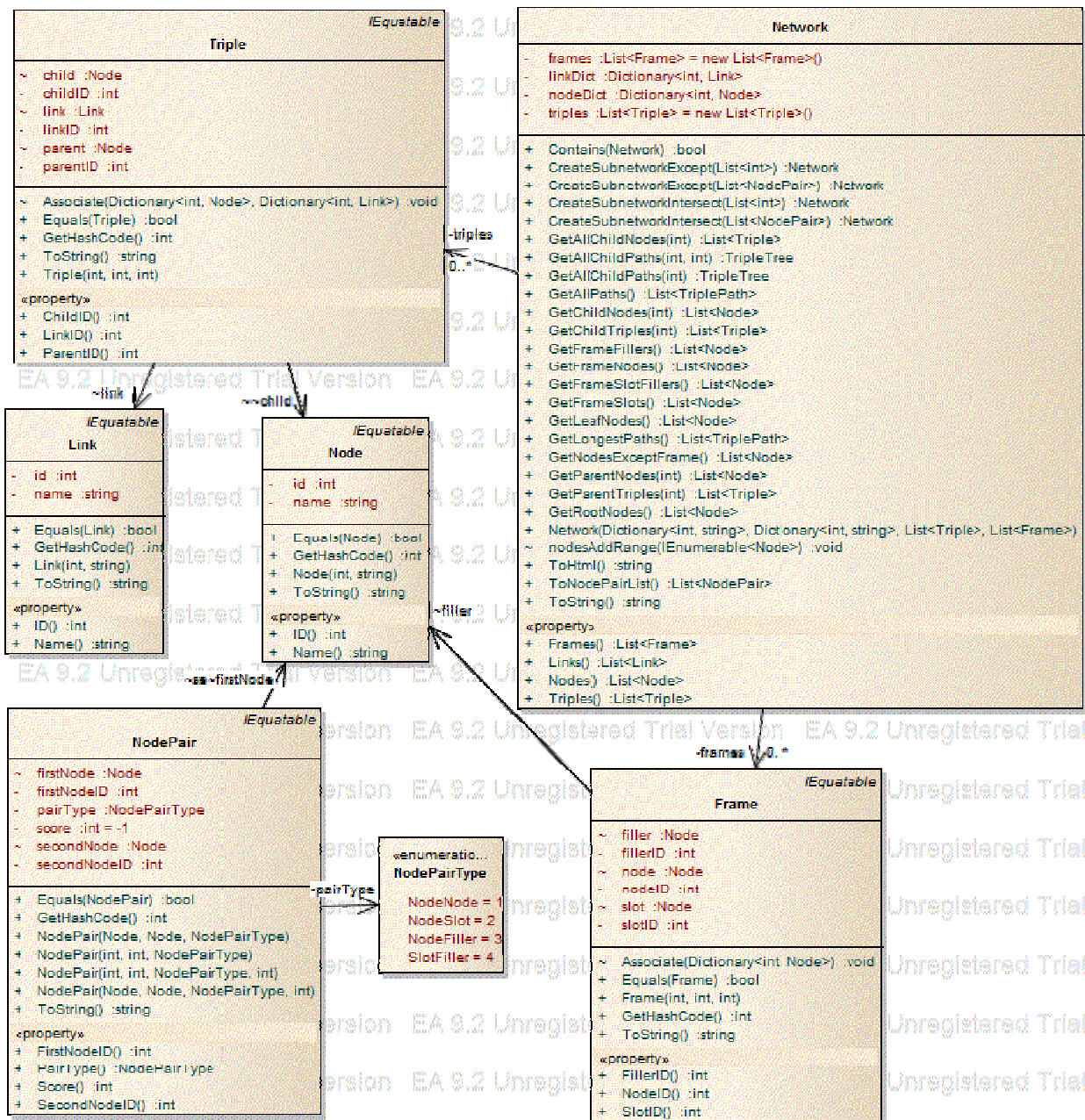
Dijagram klasa za testiranje:



Dijagram klasa paketa Courseware:



Dijagram klasa paketa područno znanje:



Dijagram klasa modela učenika:



## 8.7 Anketni upitnik korišten u okviru prototipnog testiranja sustava AC-ware Tutor

Pitanja o "zadovoljstvu" u korištenju AC-ware Tutor-a za vrijeme učenja i testiranja (9 pitanja):

1. Kako biste izrazili zadovoljstvo/nezadovoljstvo sustavom?
  - a. Vrlo zadovoljan/na
  - b. Zadovoljan/na
  - c. Nezadovoljan/na
  - d. Vrlo nezadovoljan/na
  - e. Nemam mišljenje
  
2. Kako je vrijeme odmicalo, učenje računarstva uz pomoć AC-ware Tutor-a mi je bilo:
  - a. teže (složenije)
  - b. jednako teško
  - c. jednako lako
  - d. lakše (jednostavnije)
  
3. Želite li nastaviti učiti računarstvo uz pomoć AC-ware Tutor-a?
  - a. želim
  - b. želim povremeno
  - c. ne želim
  
4. Nastavni sadržaj računarstva obrađen uz pomoć AC-ware Tutor-a razumio/la sam:
  - a. u potpunosti
  - b. djelomično
  - c. nisam ih uopće razumio/la

Zaokružite broj koji je bliže onomu što smatrate točnim.

5. Sustav AC-ware Tutor nam olakšava učenje  
 Ne slažem se    1    2    3    4    5    Slažem se
  
6. Sustav AC-ware Tutor je jednostavan za korištenje  
 Ne slažem se    1    2    3    4    5    Slažem se
  
7. Način tumačenja gradiva u sustavu AC-ware Tutor je zanimljiv.    5    4    3    2    1    Način tumačenja gradiva u sustavu AC-ware Tutor je dosadan.
  
8. Nastavni sadržaji u sustavu AC-ware Tutor su razumljivi.    5    4    3    2    1    Nastavni sadržaji u sustavu AC-ware Tutor su nerazumljivi.

---

9.	Pitanja u testovima u sustavu AC-ware Tutor su razumljiva	5	4	3	2	1	Pitanja u testovima u sustavu AC-ware Tutor su nerazumljiva.
----	---	---	---	---	---	---	--

**Pitanja o odnosu učenja i testiranja znanja uz pomoć AC-ware Tutor-a i tradicionalne nastave (3 pitanja):**

10. Nastava računarstva uz pomoć AC-ware Tutor-a je u odnosu na klasičnu nastavu:
- zanimljivija
  - jednako zanimljiva
  - manje zanimljiva
11. Rezultati u učenju računarstva uz pomoć AC-ware Tutor-a su u odnosu na rezultate koje sam postizao/la ranije:
- bolji
  - jednaki
  - lošiji
12. Korištenjem AC-ware Tutor-a u nastavi mogla bi se ostvariti kvalitetnija nastava.
- točno
  - djelomično točno
  - netočno

**Pitanja za ocjenjivanje AC-ware Tutora (9 pitanja):**

13. Ocijenite koliko sustav AC-ware Tutor zadovoljava vaše potrebe:
- Potpuno zadovoljava
  - zadovoljava 80%
  - zadovoljava 50%
  - zadovoljava manje od 50%
  - ne zadovoljava
14. Ocijenite preglednost sustava AC-ware Tutor:
- vrlo pregledan
  - pregledan
  - pregledan i nepregledan, ovisno o području
  - nepregledan
  - vrlo nepregledan
15. Biste li preporučili AC-ware Tutor svojim prijateljima i suradnicima?
- svakako ne
  - vjerojatno ne
  - ne znam
  - vjerojatno da
  - svakako da

---

Zaokružite broj koji je bliže onomu što smatrate točnim.

16. Kvaliteta AC-ware Tutor sustava

Loša                    1        2        3        4        5        Odlična

17. Funkcionalnost AC-ware Tutor sustava

Loša                    1        2        3        4        5        Odlična

18. Jasnoća AC-ware Tutor sustava

Loša                    1        2        3        4        5        Odlična

19. Kako Vam se sviđa AC-ware Tutor?

20. Što biste naveli kao najveću prednost sustava AC-ware Tutor?

21. Što biste naveli kao najveću manu sustava AC-ware Tutor?

## 9 Životopis

Rođena je 10.11.1978. godine u Splitu. Završila je Prirodoslovno – matematičku gimnaziju (III. Gimnazija) u Splitu 1997. godine s odličnim uspjehom. Tijekom osnovnoškolskog i srednjoškolskog školovanja sudjelovala je na natjecanjima iz matematike i informatike na kojima je ostvarila značajne rezultate. Nakon devetogodišnjeg učenja engleskog jezika u Centru za strane jezike u Splitu, položila je ispit First Certificate in English.

Studij na Fakultetu prirodoslovno – matematičkih znanosti i odgojnih područja Sveučilišta u Splitu upisala je 1997. godine. Tijekom studija bila je državni stipendist. Diplomirala je 27.11.2001. godine s odličnim uspjehom i postigla visoku spremu i stručno zvanje *profesor matematike i informatike*. Naslov njenog diplomskog rada je „Model traganja – dijagnostička tehnika inteligentnih tutorskih sustava“ napravljenog pod mentorstvom prof.dr.sc. Slavomira Stankova s istog fakulteta.

Od 01.01.2002. zaposlena je kao znanstveni novak na Fakultetu prirodoslovno – matematičkih znanosti i odgojnih područja (danas Prirodoslovno-matematički fakultet) Sveučilišta u Splitu, te je od 2002. do 2006. godine radila kao suradnik na znanstvenoistraživačkom projektu 177110 „Računalni i didaktički aspekti inteligentnih autorskih alata u obrazovanju“.

Sveučilišni znanstveni poslijediplomski studij na Fakultetu elektrotehnike i računarstva, Sveučilište u Zagrebu, smjer Jezgra računarstva, upisala je u veljači 2002. godine.

Istraživačko zvanje mlađeg asistenta (znanstveni novak) za znanstveno područje Tehničkih znanosti, polje Računarstva stekla je 19.12.2002. godine.

Sudjelovala je kao suradnik na Tehnologijskom projektu TP-02/0177-01 „Web-orijentirana inteligentna autorska ljuska“ od 2003. do 2005. godine.

Od 2007. godine prijavljena je kao suradnik na znanstvenoistraživačkom projektu 177-0361994-1996 „Oblikovanje i vrednovanje inteligentnih sustava e-učenja“.

Magistrirala je 01.02.2007. godine na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu. Naslov njenog magistarskog rada je „Vrednovanje učinka inteligentnih sustava e-učenja“ napravljenog pod mentorstvom prof.dr.sc. Slavomira Stankova.

Sveučilišni znanstveni poslijediplomski doktorski studij na Fakultetu elektrotehnike i računarstva upisala je u veljači 2007. godine.

Odlukom Fakultetskog vijeća FER-a, 23.11.2007. joj je dodijeljena srebrna plaketa "Josip Lončar" za posebno istaknuti magistarski rad iz područja tehničkih znanosti znanstveno polje računarstvo.



Za svoje znanstveno djelovanje dobila je sljedeća značajnija priznanja: urednici Marquis "Who's Who" (SAD) odabrali su njen životopis za objavljivanje u desetom izdanju „Who's Who in Science and Engineering, 2008 - 2009“, urednici „International Biographical Centre“ (GB) objavili su njen životopis u „Top 100 scientists 2008“.

Članica je Hrvatskog društva za sustave (CROatian Systems Society – CROSS). Bila je članica organizacije Mensa Hrvatska.

Sudjeluje u izvođenju predavanja i praktičnog dijela nastave na sljedećim kolegijima: Uvod u računarstvo, Programiranje I, Vrednovanje sustava e-učenja i Objektno-orijentirana analiza i oblikovanje na Prirodoslovno-matematičkom fakultetu, te na kolegiju Vizualno modeliranje (učiteljski studij) na Filozofskom fakultetu, oba na Sveučilištu u Splitu.

Autorica je ili suautorica dvadesetak znanstvenih radova objavljenih u časopisima ili zbornicima radova s međunarodnih i domaćih znanstvenih skupova.

Udana je i majka dvoje djece. Aktivno govori i piše engleski jezik.

**Biography**

She was born on the 10.11.1978 in Split. She has finished the Gymnasium for natural sciences and mathematics in Split in 1997 with excellent marks. During her primary and high school education, she had participated in many competitions in mathematics and computer science where she had achieved significant results. After nine years of learning English language in the Center for foreign languages in Split, she has passed the First Certificate in English.

She graduated under supervision of prof. PhD Slavomir Stankov on 27.11.2001 with excellent success on the Faculty of Natural Sciences and Kinesiology University of Split, and gained a BA for mathematics and computer science teacher. The theme of her graduate thesis was "Model tracing – a diagnostic technique of intelligent tutoring systems".

Since 01.01.2002 she has been employed as a scientific novice at the Faculty of natural sciences and kinesiology University of Split, and in the period 2002 – 2006 she has been collaborating on a scientific and research project 177110 „Computational and didactical aspects of intelligent authoring tools in education “.

She enrolled postgraduate study at the Faculty of Electrical Engineering and Computing, University of Zagreb, in February 2002.

She was collaborating on a technological project TP-02/0177-01 „Web-oriented intelligent authoring shell“ from 2003 to 2005.

Since 2007 she has been collaborating on scientific and research project 177-0361994-1996 „Design and evaluation of intelligent e-learning systems “.

She got her MA degree on 01.02.2007 on the Faculty of Electrical Engineering and Computing University of Zagreb. The theme of her master's thesis was „Evaluating educational influence of intelligent e-learning systems“, and it was done under supervision of prof. PhD Slavomir Stankov. The board of Faculty of Electrical Engineering and Computing gave her a silver plaque "Josip Lončar" for outstanding master's thesis.

She enrolled postgraduate doctoral study at the Faculty of Electrical Engineering and Computing, University of Zagreb, in February 2007.

The editors of Marquis "Who's Who" (USA) have selected her biographical profile for inclusion in the 10th Edition of "Who's Who in Science and Engineering, 2008-2009". She is a listee of the IBC "Top 100 Scientists" for the year 2008.

She is a member of Croatian System Society (CROSS). She was a member of organization Mensa Croatia.

She is involved in realization of exercises for following courses: Introduction to computer science, Programming I, Evaluation of e-learning systems, Object-oriented analysis and design.

She authored or coauthored about twenty scientific papers published in journals or conference proceedings.

She is married and has two children. She actively speaks and writes in English language.